universität wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation / Title of the Doctoral Thesis

“Approximation capabilities of deep ReLU neural networks”

verfasst von / submitted by

Dennis Elbrächter

angestrebter akademischer Grad / in partial fulfillment of the requirements for the degree of

Doktor der Naturwissenschaften (Dr. rer. nat.)

Wien, 2021 / Vienna, 2021

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the studen record sheet:	UA 796 605 405
Dissertationsgebiet lt. Studienblatt / field of study as it appears on the student record sheet:	Mathematik
Betreut von / Supervisor:	Univ.-Prof. Dr. Philipp Grohs

Abstract

This thesis contains a series of papers which explore the approximation capabilities of deep ReLU networks. Abstractly speaking they constitute parametrized model classes for nonlinear approximation, where the parameters define a sequence of affine transformations from which the corresponding function is obtained as the composition of these affine transformations with a simple parameter-independent nonlinear function interjected between every two of them. This structure ensures that any composition of functions which individually can be efficiently approximated by deep ReLU networks can itself be efficiently approximated by them. As shown in the thesis, this turns out to be a very versatile and powerful tool. Among other things it is used to establish that deep ReLU networks are capable of approximating the solutions to certain high dimensional partial differential equations with a number of parameters which depends only polynomially on the dimension. Furthermore it is shown that, in a rate-distortion sense, they are at least as good at approximating a given function class as any classical affine or Weyl-Heisenberg dictionary (e.g. wavelet or Gabor frame) under rather mild conditions on their generator functions. Lastly, a novel approach is developed which makes use of approximation capabilities of neural networks to determine scenarios in which the optimization landscape in neural network training does not have bad local minima.

Zusammenfassung

Diese Thesis besteht aus einer Reihe von Publikationen die das Approximationsvermögen von tiefen ReLU Netzwerken untersuchen. Abstrakt gesehen, konstituieren sie parametrisierte Modelklassen für nichtlineare Approximation bei der die Parameter eine Folge von affinen Transformationen definieren aus welchen die korrespondierende Funktion erzeugt wird als Komposition dieser affinen Transformationen, wobei eine simple parameter-unabhängige Funktion zwichen jeweils zwei davon zwichengeschaltet wird. Diese Struktur stellt sicher, dass jede Komposition von Funktionen, die individuell effizient durch tiefe ReLU Netzwerke approximiert werden können, im Ganzen effizient durch tiefe ReLU Netzwerke approximiert werden kann. Wie in dieser Thesis gezeigt wird stellt sich dies als ein vielseitiges und mächtiges Werkzeug heraus. Unter anderem wird es verwendet um zu etablieren dass tiefe ReLU Netzwerke fähig sind Lösungen hochdimensionaler partieller Differentialgleichungen zu approximieren mit einer Anzahl von Parametern die nur polynomiell von der Dimension abhängt. Desweiteren wird gezeigt, dass sie, in einem Raten-Verzerrungs Sinn, mindestens so gut darin sind eine gegebene Funktionenklasse zu approximieren wie jedes klassische affine oder Weyl-Heisenberg Wörterbuch (z.B. Wavelet oder Gabor Frame), unter milden Anforderungen an ihre Generatorfunktionen. Letztlich wird ein Ansatz beschrieben welcher das Approximationsvermögen von neuralen Netzwerken verwendet um Szenarien zu bestimmen in welchen die Optimierungslandschaft beim Trainieren neuraler Netzwerke keine schlechten lokalen Minima hat.

Acknowledgement

I am very grateful to Prof. Philipp Grohs for providing me with exciting research questions as well as an amazing environment to pursue them in. I would like to thank Prof. Caroline Lasser and Prof. Hans Georg Feichtinger for helping me find this great position in the first place. I also extend my gratitude towards Prof. Helmut Bölcskei, Prof. Arnulf Jentzen, Dmytro Perekrestenko, and Prof. Christoph Schwab for the fruitful collaborations and the opportunities to learn writing mathematical papers. Many thanks to Monika Dörfler for her mentorship and the wisdom she imparted, both intentionally and unintentionally. I was very lucky to share my office with some amazing people in Julius Berner, Pavol Harar, and Sarah Koppensteiner, and thank them for their support, patience, and general levity. I would like to thank all my colleagues at OMP1 for many enjoyable lunches, coffee brakes, and discussions about work and other nonsense. I am also very grateful to Prof. Anita Schöbel for awakening and furthering my interest in mathematics at a very young age. Lastly, I would like to thank my family for their constant support and encouragement, without which all of this would not have been possible.

Contents

Preamble

Introduction **5**

Preliminaries **6**

Synopses of the Publications **8**

3.1 DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing 8

3.2 Deep neural network approximation theory 8

3.3 Towards a regularity theory for ReLU networks – chain rule and global error estimates 9

3.4 How degenerate is the parametrization of neural networks with the ReLU activation function? 10

Discussion **12**

References **16**

Publications

DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing **18**

Deep Neural Network Approximation Theory **69**

Towards a regularity theory for ReLU networks – chain rule and global error estimates **150**

How degenerate is the parametrization of neural networks with the ReLU activation function? **156**

Appendix

Curriculum Vitæ **180**

Preamble

1. Introduction

In recent years the study of neural networks has attracted a great number of mathematicians from a variety of different backgrounds trying to answer a very simple question:

Why do they actually work?

More specifically why do they work this time given that the concept of neural networks has been around for many decades and already went through periods of great theoretical interest which, however, did not yield matching practical success. Nowadays on the other hand it seems almost impossible to go a week without hearing about another task where neural networks have produced state-of-the-art results. In particular, they have surpassed the mundane triumphs of excelling in standard machine learning tasks like game playing, image classification, or natural language processing, and proceeded to making a lot of traditional practitioners very nervous by posting impressive results in the context of, e.g., PDEs or inverse problems.

One thing which certainly has changed is the amount of available computing power and as demonstrated by the great Egyptian pyramids it may appear outright miraculous what one can achieve simply by throwing an almost unlimited amount of labour at a problem[1]. Unfortunately, this is not a particularly satisfying answer from a mathematical point of view and as such the most prominently postulated reason for the success of modern neural networks is their depth. While the jury is still out on whether this is actually the case, it is most certainly very intriguing from the perspective of nonlinear approximation as it constitutes a paradigm shift to the study of *compositions of simple things* instead of *linear combinations of simple things.*

As such, the primary endeavour of my doctoral studies was to explore this newfound power of composition in order to establish novel quantitative results on the approximation capabilities of deep neural networks. The results of this journey are recorded in this thesis in the form of four papers I authored, accompanied by some (hopefully) illuminating comments.

[1] Although using GPUs to do the work is certainly an improvement from a moral standpoint.

2. Preliminaries

Before we get into the motivating questions behind my publications, I will briefly concretize the central mathematical object I was interested in. When I talk about a neural network I mean a finite sequence of matrix-vector tuples $\Phi := (A_\ell, b_\ell)_{\ell=1}^{L}$ with $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$, $b_\ell \in \mathbb{R}^{N_\ell}$. It induces a function, called the realization of Φ, via

$$\mathcal{R}(\Phi) := W_L \circ \rho \circ W_{L-1} \circ \cdots \circ \rho \circ W_1,$$

where $W_\ell(x) := A_\ell x + b_\ell$ and $\rho : \mathbb{R} \to \mathbb{R}$ is the activation function which, with the common abuse of notation, is applied componentwise. The entries of the matrices and vectors are, as usual, referred to as the weights of the network. I will call L the depth of the network, N_ℓ the dimension of the ℓ-th layer, and $N := (N_0, N_1, \dots, N_L)$ the architecture of the network.

While in many places in the literature the term neural network is used rather ambiguously to refer to both the collection of weights as well as the associated function, I believe it is worth it to put some emphasis on distinguishing between the two, as the realization map $\mathcal{R}\colon \Phi \mapsto \mathcal{R}(\Phi)$ is very much not injective. A simple consequence of this is that any notion of size of a neural network (e.g. depth or number resp. magnitude of its weights) is only well-defined for Φ but not for its realization $\mathcal{R}(\Phi)$. As a primary concern of the first three papers is establishing the required size of neural networks which are capable of approximation some function (class) of interest with a prescribed accuracy, this distinction mostly serves to phrase things in a rigorous manner. It is even more important in the fourth paper, which studies how badly the realization map fails to be injective.

In the following we will mostly be concerned with ReLU networks, i.e. neural networks where $\rho(x) = \max\{0, x\}$ is the so-called ReLU activation function. The first two papers exclusively consider ReLU networks, whereas the third paper contains some remarks on general continuous piecewise linear activation functions and the fourth paper has a number of abstract observations which are independent of the activation function used. The choice to focus on the ReLU activation function is, of course, partially motivated by its ubiquity in practise, but I also very much like it from a perspective of mathematical beauty as it arguably constitutes one of the simplest (continuous) nonlinear functions. It is quite remarkable to observe how adding just a pinch of non-linearity to a composition of affine linear transformations allows the efficient approximation of a plethora of functions which are not at all affine linear.

As mentioned in the introduction I was particularly interested in making use of the compositional nature of neural networks. One benefit of this is that the approximation of any function of the form $f = g \circ h$ may be broken down into the approximation of g and h individually, which enables a very convenient modular approach[1] of constructing neural network approximations. Another prospect offered is the efficient representation of self-similar functions including a very simple case which nonetheless has been at the center of a number of interesting works. Namely the observation by Telgarsky [2] that composing the hat function[2] n many times with itself yields a function h_n which on each interval $[k2^{-n}, (k+1)2^{-n}]$, $k \in \{0, \dots, 2^n - 1\}$ behaves like the hat function dilated by 2^n. Yarotsky [3] subsequently combined this observation with the rapidly converging series representation $x - x^2 = \sum_{n=0}^{\infty} 4^{-n} h_n(x)$ as well as a polarization and a scaling argument to establish that ReLU networks can approximate multiplication and consequently polynomials on bounded domains with depth and number of weights scaling only logarithmically w.r.t. accuracy and domain size. This finding has been used by Yarotsky as well as several others to establish that ReLU networks are capable of efficiently approximating various types of regular functions even though the ReLU itself is only once (weakly) differentiable. It is interesting to remark that, by virtue of the construction mentioned above, this capability to exploit smoothness comes as a consequence of the ability to efficiently create self-similar structures.

Roughly speaking each of the four papers may be seen as an attempt to answer the corresponding one of the following four questions:

1. What kind of interesting high-dimensional functions can be approximated by deep ReLU networks whose size does not depend exponentially on the dimension?

2. With respect to approximation capability how do deep ReLU networks compare to dictionaries like wavelet or Gabor frames?

3. Can deep ReLU networks approximate functions in (first-order) Sobolev norm?

4. Can the approximation capabilities of neural networks be used to explain the well-behavedness of the optimization landscape in training?

[1] A very nice formal framework for this was introduced by Philipp Petersen and Felix Voigtlaender in [1], which served as the main inspiration for the notation I have used in my works.

[2] I.e. the function on $[0, 1]$ that is 1 at the center of the interval, 0 at both ends, and linear in between, which can, of course, very easily be represented by a ReLU network.

3. Synopses of the Publications

3.1 DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing

This paper is concerned with the efficient approximation of solutions of high dimensional partial differential equations (PDEs). Specifically we consider the solution of the Black-Scholes equation for European option pricing. We start by briefly reviewing how to derive a semi-explicit form using the Feynman-Kac formula. This solution consists of a d-dimensional tensor product of univariate functions which is then integrated w.r.t. an auxiliary variable. We proceed by analyzing the regularity of the univariate functions and the approximation of the integral by composite Gaussian quadrature. Subsequently we construct ReLU networks for the approximation of univariate k-smooth functions and tensor products as well as for the implementation of the Gaussian quadrature. These results are then combined to establish that the solution of the d dimensional Black-Scholes equation for European option pricing may be approximated to within ε-error by ReLU networks, whose depth depends logarithmically on d and ε and whose number of weights has polynomial scaling in d and spectral scaling in ε.

3.2 Deep neural network approximation theory

In this paper we study the approximation capabilities of deep ReLU networks from an information theoretic perspective which enables us to draw a comparison to classical dictionary based approximation methods. In the latter case approximation capabilities are usually quantified by considering how many terms are required to approximate every element of a given function class to within some error by a linear combination of dictionary elements. As neural networks exhibit a different structure one needs to take some care how to quantify their rate of approximation in a comparable manner. Our approach to deal with this is to borrow from rate distortion theory and in particular employ the notion of nonlinear approximation rate under polynomial depth search constraint. Roughly speaking the constraint imposes that one only considers M-term approximations for which the size of the coefficients and the indices of participating dictionary element are bounded by some polynomial in M. Heuristically, this provides some practical relevance to these approximations as it ensure that the indices and coefficients may be encoded as a bit string without incurring too much of an quantization error. On a more abstract level it ascertains that these approximation rates are (sharply) bounded by the optimal exponent of the functions class, which quantifies its complexity via the behaviour of its metric entropy. We introduce an analogous M-weight approximation rate for ReLU networks where networks with M many weights are only allowed to have a depth which is bounded by a polynomial of $\log(M)$ and weights whose magnitude is bounded by a polynomial of M. We show that these conditions ensure that M-weight approximations can similarly be encoded without introducing a significant quantization error, i.e. this M-weight approximation rate is also bounded by the optimal exponent of the function class in question. Moreover we establish a transference result which states that the M-weight approximation rate with ReLU networks is at least as good as the M-term approximation rate in any dictionary consisting of translations, dilations, and modulations of a generator function (e.g. wavelets, shearlets, Gabor systems) which can be approximated well by neural networks. We verify in detail that this condition is fulfilled for spline wavelets and note that it holds for most commonly studied affine and Weyl-Heisenberg dictionaries.

3.3 Towards a regularity theory for ReLU networks – chain rule and global error estimates

This paper primarily deals with an issue we encountered when attempting to establish approximation results for deep ReLU networks in Sobolev $W^{1,\infty}$-norm. Approximation results for deep ReLU networks usually take advantage of the compositional nature of neural networks by first constructing networks which approximate some simple functions and subsequently combining them in order to get results for more complicated functions. As such, one often wants to deduce a bound on $\|\mathcal{R}(\Phi) \circ \mathcal{R}(\Psi) - f \circ g\|$ from bounds on $\|\mathcal{R}(\Phi) - f\|$ and $\|\mathcal{R}(\Psi) - g\|$. While this is very straightforward if $\|\cdot\|$ is the L^∞-norm, the case of $\|\cdot\| = \|\cdot\|_{W^{1,\infty}}$ requires application of the chain rule to ReLU network realizations which are only almost everywhere differentiable. In particular it may happen that $\mathcal{R}(\Psi)$ maps a non-nullset into the set of points for which the derivative of $\mathcal{R}(\Phi)$ is not well-defined, i.e. $(D\mathcal{R}(\Phi))(\mathcal{R}(\Psi)(x))$ is in general not even well-defined almost everywhere. We formally introduce a notion of a derivative of a ReLU network which coincides almost everywhere with the standard derivative of its realization and obeys a chain rule. Subsequently we briefly discuss how our notion of neural network derivative helps in establishing approximation results and how to extend this idea to general continuous piecewise linear activation functions. We conclude by illustrating how to get, for ReLU networks, a certain type of global approximation result, which is relevant in the context of approximating solutions of PDEs.

3.4 How degenerate is the parametrization of neural networks with the ReLU activation function?

In this paper we investigate the notion of inverse stability of the realization map $\mathcal{R}$, i.e. the function which maps a neural network parametrization[1] to its realization. We say inverse stability holds for some subset Ω of parametrizations w.r.t. some norm $\|\cdot\|$, if for any $\Gamma \in \Omega$ and $g \in \mathcal{R}(\Omega)$ there exists $\Phi \in \mathcal{R}^{-1}(g)$ such that we can bound the distance[2] between Γ and Φ relative to $\|\mathcal{R}(\Gamma) - \mathcal{R}(\Phi)\|$. The purpose of establishing inverse stability is to connect the abstract optimization problem $\min_{g\in\mathcal{R}(\Omega)} \mathcal{L}(g)$ over the set of realizations to the practically tangible problem $\min_{\Phi\in\Omega} \mathcal{L}(\mathcal{R}(\Phi))$ over the set of parametrizations, where $\mathcal{L}$ is some loss function that only depends on the realization of a network[3]. This is motivated by the following observation. Let us take some convex set S such that every function in S may be approximated by realizations in $\mathcal{R}(\Omega)$ up to ε-error and assume the loss function is convex and Lipschitz on bounded domains[4]. Then the loss at any local minimum of the regularized optimization problem $\min_{g\in\mathcal{R}(\Omega)\cap S} \mathcal{L}(g)$ is upper bounded by the loss at a global minimum plus a term which is proportional to ε/r, where r is the radius[5] of the local minimum. In cases where the approximation error ε can be guaranteed to be arbitrarily small by choosing a correspondingly large architecture this means that by sufficiently increasing the size of the architecture used one can guarantee that any local minimum is either extremely narrow orä already almost optimal. However, in order to transfer this result to the tangible parametrized problem $\min_{\Phi\in\Omega\cap\mathcal{R}^{-1}(S)} \mathcal{L}(\mathcal{R}(\Phi))$ we would need to know that Γ being a local minimum of the parametrized problem implies that $\mathcal{R}(\Gamma)$ is a local minimum (with a comparably large radius) of the problem over the set of realizations, which is exactly what our inverse stability is designed to ensure. Having established this high level idea, we proceed by solving the question of inverse stability w.r.t. to the Sobolev $W^{1,\infty}$-norm[6] for shallow ReLU networks. We first establish a number of pathologies which prevent inverse stability and subsequently proof that by restricting the set of all parametrizations of a given architecture such that these pathologies are avoided inverse stability w.r.t. to the Sobolev $W^{1,\infty}$-norm does hold. In addition we note that these restrictions, up to some technicalities, only get rid of redundancies. Specifically, for any set Π containing all parametriza-

[1] Note that in this paper we refer to the sequence of matrix-vector tuples as a (neural network) parametrization in order to emphasize the difference to the corresponding realization.

[2] As the space of parametrizations is finite dimensional, the norm in which we consider this distance is not particularly relevant.

[3] Note that this is naturally the case for any loss function which only contains a data fidelity term and no regularization terms, which of course usually do depend on the parametrization.

[4] This is fullfilled for, e.g., the means squared loss $\mathcal{L}(g) = \sum_{i=1}^{n}(g(x_i) - y_i)^2$ with some labeled data $((x_i, y_i))_{i=1}^{n}$.

[5] The radius of a local minimum Γ is the largest $r > 0$ such that Γ is minimal over a ball of radius r around Γ.

[6] Failure of inverse stability w.r.t. the L^∞-norm had previously been shown by Petersen, Raslan, and Voigtlaender.

tions of a given architecture there is a restricted set of parametrizations Ω of a slightly larger architecture such that $\mathcal{R}(\Pi) \subseteq \mathcal{R}(\Omega)$ and inverse stability holds on Ω.

4. Discussion

At face value the first paper only establishes approximation rates for the solution of one specific PDE for which a semi-explicit form is known. While we chose to have a presentation focused around a concrete example, I believe that the relevance of the paper very much is not limited to this specific case as the way we constructed the approximating networks is not at all based on something unique to this PDE. Roughly speaking the paper establishes three things that ReLU networks a capable of doing:

(i) Approximating k-smooth univariate functions

(ii) Approximately implementing multiplication

(iii) Implementing composite Gaussian quadrature

While each of these is by itself not particularly exciting, the crux of the matter which makes neural networks such remarkably powerful tools for approximation is that, without any adaptations to their basic framework, they can do all of these things in composition. In our example in the paper we use (i) to construct networks to approximate univariate functions f_c which we combine with (ii) to get networks approximating their tensor product $F_c(x) := \Pi_{i=1}^{d} f_c(x_i)$. Subsequently we incorporate (iii) to obtain networks approximating $x \mapsto \int_0^\infty 1 - F_c(x)\mathrm{d}c$.

Due to the flexibility of this modular method of construction afforded by the compositional nature of deep ReLU networks, one can establish analogous results for any PDE solution which can be represented via tensor products of low-dimensional functions followed by any operations which can be (approximately) implemented by ReLU networks. Note that even if with exact multiplication of d functions f_i with $\|f_i\|_{L^\infty} \leq B$ and approximations Φ_i with $\|f_i - \Phi_i\|_{L^\infty} \leq \varepsilon$ the error of the tensor product approximation $\|\Pi_{i=1}^{d} f_i - \Pi_{i=1}^{d} \Phi_i\|_{L^\infty}$ may still be of order $dB^{d-1}\varepsilon$ as can, e.g., be seen in the proof of Proposition 6.4 in the first paper. Thus in order to approximate a tensor product by networks with size scaling at most polynomially in the dimension one either requires $B \leq 1$ or the existence of networks approximating the f_i to within error ε with size depending only polylogarithmically[1] on ε. As such, a certain regularity of the partaking univariate functions is still required - or more precisely a 'well-approximability' by ReLU networks, for which regularity is only a sufficient condition. Proposition X.3 in the second paper shows ReLU networks are capable of approximating the nowhere differentiable Weierstrass with polylogarithmic rates, which means that

[1] That is like $\pi(\log(\varepsilon))$, where π is some polynomial.

regularity is certainly not a necessary condition. A satisfying general description of the approximation spaces of neural networks is still very much an open problem. Nonetheless, I believe that the results in the first paper are a good indication that deep ReLU networks provide powerful and flexible model classes for the approximation of solutions to high dimensional PDEs.

The focus of my second paper was to put the approximation rates of deep ReLU networks into context. Neural network approximation results usually describe how the number of required parameters (i.e. weights) of a neural network increases relative to the desired accuracy of approximation. This parameter-accuracy trade-off is, of course, an important consideration for any parametrized model class one might want to use in order to approximate some class of functions. However, unless one takes into account how impactful these parameters are, looking at this trade-off may be misleading as can be seen in the following examples. Any separable function class contains a countable dense set which would constitute a dictionary which can achieve an arbitrarily accurate approximation simply by picking the right index, i.e. a single parameter. Moreover, if the function class is compact one can, for any given $\varepsilon > 0$, take the centers of an ε-covering, which would constitute a finite dictionary which can approximate any element of the function class to within ε-error with only a single parameter. Of course, finding this one parameter in practice would generally be an entirely hopeless endeavour. Nonetheless it illustrates that care needs to be taken when assessing the parameter-accuracy trade-off of parametrized model classes, in particular if one wants to compare them to each other.

In our paper we deal with this issue by using approximation rates which enforce that M-term respectively M-weight approximations both canonically induce bit string encoder-decoder pairs with the same order of accuracy and (up to poly-logarithmic factors) same length M. The main purpose of this is to allow for a reasonable comparison between notions of approximation rates for dictionaries and neural networks even though they have a fundamentally different structure. It also imbues these notions of approximation rate with some practical relevance as it ensures that reasonably accurate quantizations of these abstract approximations could, in principle, be produced by a practical algorithm. This remains, of course, an analysis for which function classes parsimonious neural networks approximations exist, whereas it is still an open question of whether and how these approximations can actually be found in practise. Nonetheless, I believe establishing that ReLU neural networks are capable of simultaneous rate-distortion optimal approximation of a wide variety of classically studied function classes serves as a compelling substantiation of the versatility of ReLU networks as approximators. While to me it does not seem to indicate that neural networks are necessarily better[2] suited than more targeted frameworks in cases where one has sufficient structural knowledge, it appears to be in parallel with the practical observation that neural network based methods are particularly advantageous in cases when not much is known about the desired solution.

[2] In fact, they might very well be worse for any problem of practical size, as our results are asymptotic in nature.

The third paper arose from our endeavour of taking the Yarotsky-type constructions and extending the approximation results from L^∞-norm to Sobolev $W^{1,\infty}$-norm. As described in the paper the otherwise very convenient modular approach of constructing these approximations and obtaining the corresponding error estimates turned out to be a bit more tricky in this case, since we could not employ the standard chain rule as ReLU network realizations are only almost everywhere differentiable. While I am quite satisfied with our solution, we did not pursue this line of research much further as we realized that Gühring, Kutinyok, and Petersen had already been working on the same issue and covered the approximation of Sobolev regular functions in $W^{s,p}$-norm with $s \in [0,1]$, $p \in [1,\infty]$ very nicely in [4]. They circumvented the failure of the standard chain rule by relating the Sobolev semi-norm $|f|_{W^{1,\infty}(\Omega)} := \|Df\|_{L^\infty(\Omega)}$ to the Lipschitz constant of f and obtaining the necessary estimates along this path. This does, however, introduce some overhead in the estimates which may be avoided by our solution. Namely for $f\colon \mathbb{R}^d \to \mathbb{R}^m$, $g\colon \mathbb{R}^m \to \mathbb{R}$, $\Omega_1 \subseteq \mathbb{R}^d$, and $f(\Omega_1) \subseteq \Omega_2 \subseteq \mathbb{R}^m$ they obtain the estimate $|g \circ f|_{W^{1,\infty}(\Omega_1)} \leq m\sqrt{d}|g|_{W^{1,\infty}(\Omega_2)}|f|_{W^{1,\infty}(\Omega_1)}$, whereas our approach would avoid the $m\sqrt{d}$ factor. As such, I believe the results of the third paper do have value from a technical point of view.

The fourth paper diverges from the theme of the first three as it is not concerned with establishing what kind of function classes can be approximated by neural networks, but instead proposes a way to connect approximation capabilities to neural network training. A key observation is that, in a quantifiable way, the optimization problem is well-behaved when considered over the set of realizations of a fixed architecture intersected with a convex function class which can be approximated well by realizations of the chosen architecture. Essentially this mitigates the detriment of nonconvexity of the set of realizations by ensuring that any convex combination of elements in the feasible set is still close to some element in the feasible set. It would certainly be a compelling argument for why stochastic gradient descent is so surprisingly successful, if we could establish the same behaviour for the parametrized problem, i.e. that for any choice of $\varepsilon, r > 0$ a sufficiently large architecture will guarantee that any local minimum with radius greater than r has a loss which is no more than ε worse than the loss at a global optimum. However, as shown in the paper transferring these results from the realizations problem to the parametrized problem proved to be quite challenging even in the simple case of shallow ReLU networks.

The central reason for this is the severe redundancy in the parametrization of neural networks. By identifying and eliminating these detrimental redundancies we were able to establish inverse stability w.r.t. the $W^{1,\infty}$-norm for shallow ReLU networks. While I believe that this result serves as a nice proof of concept and contains a number of technical observations of independent interest, the norm w.r.t. which we established inverse stability here is regrettably a bit to weak to translate to a truly satisfying result. Specifically we would require a function class as a regularizer that can be approximated uniformly well in $W^{1,\infty}$ by neural networks of some fixed architecture, which would suggest something like a $W^{m,\infty}$-norm ball with $m > 1$. Alas, the intersection of such a norm ball with the set of ReLU realizations would only contain linear functions and as such not be particularly interesting.

Luckily, there are multiple paths beyond this. Naturally, one might attempt to establish inverse stability for a stronger norm by imposing additional restrictions to the set of realizations or consider networks with a different activation function. Another approach would be based on the fact that the inverse stability considered in the paper is somewhat of an overkill. In particular, (s, α)-inverse stability guarantees that for any given parametrization Γ which is a local minimum of radius r we know that $\mathcal{R}(\Gamma)$ is a local minimum of radius $(r/s)^{1/\alpha}$ which can then be used to bound the loss at Γ according to Theorem A.2. We do, however, not actually require that $\mathcal{R}(\Gamma)$ is a proper local minimum. Roughly speaking it would suffice to establish that $\mathcal{R}(\Gamma)$ is minimal over a sufficiently dense subset of some ball around it. Concretely, the argument in the proof of Theorem A.2 would still work as long as there exist $R, \delta > 0$ (depending suitably on r and possibly the size of the used architecture) such that for any realization g in the R-ball around $\mathcal{R}(\Gamma)$ there is a Φ in the r-ball around Γ such that $\|g - \mathcal{R}(\Phi)\| \leq \delta$. In other words, it would suffice if we could show, in a quantifiable manner, that Γ being a local minimum implies that $\mathcal{R}(\Gamma)$ is minimal over a sufficiently dense subset of some ball around $\mathcal{R}(\Gamma)$.

While establishing a meaningful almost optimality result for local minima of the parametrized problem is certainly very challenging, I believe that obtaining a better understanding of the relationship between neural network parametrizations and their realizations has great potential. Of particular interest is that the theoretical conditions of balanced weights and lack of redundant directions correspond to regularization methods which have already been observed to improve training in practise. This makes me optimistic that pursuing this line of research cannot only shed light on the underlying mathematical structures, but also inspire novel methods of training based on an understanding of these structures.

References

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

[2] M. Telgarsky, “Representation benefits of deep feedforward networks,” *arXiv:1509.08101*, 2015.

[3] D. Yarotsky, “Error bounds for approximations with deep ReLU networks,” *Neural Networks*, vol. 94, pp. 103–114, 2017.

[4] I. Gühring, G. Kutyniok, and P. Petersen, “Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms,” *Analysis and Applications*, vol. 18, no. 05, pp. 803–859, 2020.

Publications

DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing 18

Deep Neural Network Approximation Theory 69

Towards a regularity theory for ReLU networks – chain rule and global error estimates 150

How degenerate is the parametrization of neural networks with the ReLU activation function? 156

I. DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing

Authors: Dennis Elbrächter, Philipp Grohs, Arnulf Jentzen, and Christoph Schwab

Status: Accepted for publication in the special issue ”Deep Networks in Approximation Theory” of Constructive Approximation (23.04.2020).

Arxiv: https://arxiv.org/abs/1809.07669

Contribution: Main writing; Development of the results based on a rough outline.

DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing*

Dennis Elbrächter, Philipp Grohs, Arnulf Jentzen, and Christoph Schwab

February 2, 2021

Abstract

We analyze approximation rates by deep ReLU networks of a class of multi-variate solutions of Kolmogorov equations which arise in option pricing. Key technical devices are deep ReLU architectures capable of efficiently approximating tensor products. Combining this with results concerning the approximation of well behaved (i.e. fulfilling some smoothness properties) univariate functions, this provides insights into rates of deep ReLU approximation of multi-variate functions with tensor structures. We apply this in particular to the model problem given by the price of a European maximum option on a basket of d assets within the Black-Scholes model for European maximum option pricing. We prove that the solution to the d-variate option pricing problem can be approximated up to an ε-error by a deep ReLU network with depth $\mathcal{O}\big(\ln(d)\ln(\varepsilon^{-1})+\ln(d)^2\big)$ and $\mathcal{O}\big(d^{2+\frac{1}{n}}\varepsilon^{-\frac{1}{n}}\big)$ non-zero weights, where $n\in\mathbb{N}$ is arbitrary (with the constant implied in $\mathcal{O}(\cdot)$ depending on n). The techniques developed in the constructive proof are of independent interest in the analysis of the expressive power of deep neural networks for solution manifolds of PDEs in high dimension.

Keywords: neural network approximation, low-rank approximation, option pricing, high dimensional PDEs.
MSC2010 Classification: 41Axx, 35Kxx, 65-XX, 65D30

1 Introduction

1.1 Motivation

The development of new classification and regression algorithms based on deep neural networks – coined "Deep Learning" – revolutionized the area of artificial intelligence, machine learning, and data analysis [17]. More recently, these methods have been applied to the numerical solution of partial differential equations (PDEs for short) [41, 14, 11, 29, 24, 3, 10, 23, 34]. In these works it has been empirically observed that deep learning-based methods work exceptionally well when used for the numerical solution of high-dimensional problems arising in option pricing. The numerical experiments carried out in [3, 10, 23, 2] in particular suggest that deep learning-based methods may not suffer from the curse of dimensionality for these problems, but only few theoretical results exist which support this claim: In [40], a first theoretical result on rates of expression of infinite-variate generalized polynomial chaos expansions for solution manifolds of certain classes of parametric PDEs has been obtained. Furthermore, recent work [20, 4] shows that the algorithms introduced in [2] for the numerical solution of Kolmogorov PDEs are free of the curse of dimensionality in terms of network size and training sample complexity.

Neural networks constitute a parametrized class of functions constructed by successive applications of affine mappings and coordinatewise nonlinearities, see [37] for a mathematical introduction. As in [36], we introduce a neural network via a tuple of matrix vector pairs

$$\Phi = (((A^1_{i,j})_{i,j=1}^{N_1,N_0},(b^1_i)_{i=1}^{N_1}),\dots,((A^L_{i,j})_{i,j=1}^{N_L,N_{L-1}},(b^L_i)_{i=1}^{N_L}))\in \times_{l=1}^{L}\left(\mathbb{R}^{N_l\times N_{l-1}}\times\mathbb{R}^{N_l}\right)$$

*This work was performed during visits of PG at the Seminar for Applied Mathematics and the FIM of ETH Zürich, and completed during the thematic term "Numerical Analysis of Complex PDE Models in the Sciences" at the Erwin Schrödinger Institute, Vienna, from June-August, 2018. AJ acknowledges support by the Swiss National Science Foundation under grant No. 175699 DE and PhG are supported in part by the Austrian Science Fund (FWF) under project number P 30148.

for given hyperparameters $L \in \mathbb{N}$, $N_0, N_1, \ldots, N_L \in \mathbb{N}$. Given an "activation function" $\varrho \in C(\mathbb{R}, \mathbb{R})$, a neural network Φ then describes a function $R_\varrho(\Phi) \in C(\mathbb{R}^{N_0}, \mathbb{R}^{N_L})$ that can be evaluated by the recursion

$$x_l = \varrho(A_l x_{l-1} + b_1), l = 1, \ldots, L-1, \quad [R_\varrho(\Phi)](x_0) = A_L x_{L-1} + b_L. \tag{1.1}$$

The number of nonzero values in the matrix vector tuples defining Φ describe the size of Φ which will be denoted by $\mathcal{M}(\Phi)$ and the depth of the network Φ, i.e. its number of affine transformations, will be denoted by $\mathcal{L}(\Phi)$. We refer to Setting 5.1 for a more detailed description. A popular activation function ϱ is the so-called *"Rectified Linear Unit"* $\mathrm{ReLU}(x) = \max\{x, 0\}$ [17].

An increasing body of research addresses the approximation properties (or "expressive power") of deep neural networks, where by "approximation properties" we mean the study of the optimal tradeoff between the size $\mathcal{M}(\Phi)$ and the approximation error $\|u - R_\varrho(\Phi)\|$ of neural networks approximating functions u from a given function class. Classical references include [25, 8, 1, 7] as well as the summary [37] and the references therein. In these works it is shown that deep neural networks provide optimal approximation rates for classical smoothness spaces such as Sobolev spaces or Besov spaces. More recently these results have been extended to Shearlet and Ridgelet spaces [5], Modulation spaces [35], piecewise smooth functions [36] and polynomial chaos expansions [40]. All these results indicate that all classical approximation methods based on sparse expansions can be emulated by neural networks.

1.2 Contributions and Main Result

As a first main contribution of this work we show in Proposition 6.4 that low-rank functions of the form

$$(x_1, \ldots, x_d) \in \mathbb{R}^d \mapsto \sum_{s=1}^{R} c_s \prod_{j=1}^{d} h_j^s(x_j), \tag{1.2}$$

with $h_j^s \in C(\mathbb{R}, \mathbb{R})$ sufficiently regular and $(c_s)_{s=1}^R \subseteq \mathbb{R}$ can be approximated to a given relative precision by deep ReLU neural networks of size scaling like Rd^2. In particular, we obtain a dependence on the dimension d that is only polynomial and not exponential, i.e. we avoid the curse of dimensionality. In other words, we show that in addition all classical approximation methods based on sparse expansions and on more general low-rank structures, can be emulated by neural networks. Since the solutions of several classes of high-dimensional PDEs are precisely of this form (see, e.g., [40]), our approximation results can be directly applied to these problems to establish approximation rates for neural network approximations that do not suffer from the curse of dimensionality. Note that approximation results for functions of the form (1.2) have previously been considered in [39] in the context of statistical bounds for nonparametric regression.

Moreover, we remark that the networks realizing the product in (1.2) itself, have a connectivity scaling which is logarithmic in the accuracy ε^{-1}. While we will, for our concrete example, only obtain a spectral connectivity scaling, i.e. like $\varepsilon^{-\frac{1}{n}}$ for any $n \in \mathbb{N}$ with the implicit constant depending on n, this tensor construction may be used to obtain logarithmic scaling (w.r.t. the accuracy) for d-variate functions in cases where the univariate h_j^s can be approximated with a logarithmic scaling.

As a particular application of the tools developed in the present paper, we provide a mathematical analysis of the rates of expressive power of neural networks for a particular, high-dimensional PDE which arises in mathematical finance, namely the pricing of a so-called *European maximum Option* (see, e.g., [43]).

We consider the particular (and not quite realistic) situation that the log-returns of these d assets are uncorrelated, i.e. their log-returns evolve according to d uncorrelated drifted scalar diffusion processes.

The price of the European maximum Option on this basket of d assets can then be obtained as solution of the multivariate Black-Scholes equation which reads, for the presently considered case of uncorrelated assets, as

$$(\tfrac{\partial}{\partial t} u)(t, x) + \tfrac{\mu}{2} \sum_{i=1}^{d} x_i \big(\tfrac{\partial}{\partial x_i} u\big)(t, x) + \tfrac{\sigma^2}{2} \sum_{i=1}^{d} |x_i|^2 \big(\tfrac{\partial^2}{\partial x_i^2} u\big)(t, x) = 0 \, . \tag{1.3}$$

For the European maximum option, (1.3) is completed with the *terminal condition*

$$u(T, x) = \varphi(x) = \max\{x_1 - K_1, x_2 - K_2, \ldots, x_d - K_d, 0\} \tag{1.4}$$

for $x = (x_1, \dots, x_d) \in (0,\infty)^d$. It is well known (see, e.g., [13, 22] and the references there) that there exists a unique solution of (1.3)-(1.4). This solution can be expressed as conditional expectation of the function $\varphi(x)$ in (1.4) over suitable sample paths of a d-dimensional diffusion.

One main result of this paper is the following result (stated with completely detailed assumptions below as Theorem 7.3), on expression rates of deep neural networks for the basket option price $u(0,x)$ for $x \in [a,b]^d$ for some $0 < a < b < \infty$. To render their dependence on the number d of assets in the basket explicit, we write u_d in the statement of the theorem.

Theorem 1.1. *Let $n \in \mathbb{N}$, $\mu \in \mathbb{R}$, $T, \sigma, a \in (0,\infty)$, $b \in (a,\infty)$, $(K_i)_{i\in\mathbb{N}} \subseteq [0, K_{\max})$, and let $u_d \colon (0,\infty) \times [a,b]^d \to \mathbb{R}$, $d \in \mathbb{N}$, be the functions which satisfy for every $d \in \mathbb{N}$, and for every $(t,x) \in [0,T] \times (0,\infty)^d$ the equation* (1.3) *with terminal condition* (1.4).
Then there exist neural networks $(\Gamma_{d,\varepsilon})_{\varepsilon\in(0,1],d\in\mathbb{N}}$ which satisfy

(i) $\displaystyle \sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{\mathcal{L}(\Gamma_{d,\varepsilon})}{\max\{1,\ln(d)\}\,(|\ln(\varepsilon)| + \ln(d) + 1)}\right] < \infty,$

(ii) $\displaystyle \sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{\mathcal{M}(\Gamma_{d,\varepsilon})}{d^{2+\frac{1}{n}}\varepsilon^{-\frac{1}{n}}}\right] < \infty,$ *and*

(iii) for every $\varepsilon \in (0,1]$, $d \in \mathbb{N}$,

$$\sup_{x\in[a,b]^d} |u_d(0,x) - [R_{\mathrm{ReLU}}(\Gamma_{d,\varepsilon})](x)| \le \varepsilon. \tag{1.5}$$

Informally speaking, the previous result states that the price of a d dimensional European maximum option can, for every $n \in \mathbb{N}$, be expressed on cubes $[a,b]^d$ by deep neural networks to pointwise accuracy $\varepsilon > 0$ with network size bounded as $\mathcal{O}(d^{2+1/n}\varepsilon^{-1/n})$ for arbitrary, fixed $n \in \mathbb{N}$ and with the constant implied in $\mathcal{O}(\cdot)$ independent of d and of ε (but depending on n). In other words, the price of a European maximum option on a basket of d assets can be approximated (or "expressed") by deep ReLU networks *with spectral accuracy and without curse of dimensionality.*

The proof of this result is based on a near explicit expression for the function $u_d(0,x)$ (see Section 2). It uses this expression in conjunction with regularity estimates in Section 3 and a neural network quadrature calculus and corresponding error estimates (which is of independent interest) in Section 4 to show that the function $u_d(0,x)$ possesses an approximate low-rank representation consisting of tensor products of cumulative normal distribution functions (Lemma 4.3) to which the low-rank approximation result mentioned above can be applied.

Related results have been shown in the recent work [20] which proves (by completely different methods) that solutions to general Kolmogorov equations with affine drift and diffusion terms can be approximated by neural networks of a size that scales polynomially in the dimension and the reciprocal of the desired accuracy as measured by the L^p norm with respect to a given probability measure. The approximation estimates developed in the present paper only apply to the European maximum option pricing problem for uncorrelated assets but hold with respect to the much stronger L^∞ norm and provide spectral accuracy in ϵ (as opposed to a low-order polynomial rate obtained in [20]), which is a considerable improvement. In summary, compared to [20], the present paper treats a more restricted problem but achieves stronger approximation results.

In order to give some context to our approximation results, we remark that solutions to Kolmogorov PDEs may, under reasonable assumptions, be approximated by empirical risk minimization over a neural network hypothesis class. The key here is the Feynman-Kac formula which allows to write the solution to the PDE as the expectation of an associated stochastic process. This expectation can be approximated by Monte-Carlo integration, i.e. one can view it as a neural network training problem where the data is generated by Monte-Carlo sampling methods which, under suitable conditions, are capable of avoiding the curse of dimensionality. For more information on this we refer to [4].

While we admit that the European maximum option pricing problem for uncorrelated assets constitutes a rather special problem, the proofs in this paper develop several novel deep neural network approximation results of independent interest that can be applied to more general settings where a low-rank structure is implicit in high-dimensional problems. For mostly numerical results on machine learning for pricing American options we refer to [18]. Lastly we note that after a first preprint of the present paper was submitted, a number of research articles related to this work have appeared [15, 16, 19, 21, 26, 27, 28, 30, 38].

1.3 Outline

The structure of this article is as follows. The following Section 2 provides a derivation of the semi-explicit formula for the price of European maximum options in a standard Black-Scholes setting. This formula consists of an integral of a tensor product function. In Section 3 we develop some auxiliary regularity results for the cumulative normal distribution that are of independent interest which will be used later on. In Section 4 we show that the integral appearing in the formula of Section 2 can be efficiently approximated by numerical quadrature. Section 5 introduces some basic facts related to deep ReLU networks and Section 6 develops basic approximation results for the approximation of functions which possess a tensor product structure. Finally, in Section 7 we show our main result, namely a spectral approximation rate for the approximation of European maximum options by deep ReLU networks without curse of dimensionality. In Appendix A we collect some auxiliary proofs.

2 High-dimensional derivative pricing

In this section, we briefly review the Black-Scholes differential equation (1.3) which arises, among others, as Kolmogorov equation for multivariate geometric Brownian Motion. This linear, parabolic equation is, for one particular type of financial contracts (so-called "European maximum option" on a basket of d stocks whose log-returns are assumed for simplicity as mutually uncorrelated) endowed with the terminal condition (1.4) and solved for $(t, x) \in [0, T] \times (0, \infty)^d$.

Proposition 2.1. *Let $d \in \mathbb{N}$, $\mu \in \mathbb{R}$, $\sigma, T, K_1, \ldots, K_d, \xi_1, \ldots, \xi_d \in (0, \infty)$, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, and let $W = (W^{(1)}, \ldots, W^{(d)}) \colon [0, T] \times \Omega \to \mathbb{R}^d$ be a standard Brownian motion and let $u \in C([0, T] \times (0, \infty)^d)$ satisfy* (1.3) *and* (1.4)*. Then for $x = (\xi_1, \ldots, \xi_d) \in (0, \infty)^d$ it holds that*

$$\begin{aligned} u(0,x) &= \mathbb{E}\Big[\max_{i\in\{1,2,\ldots,d\}} \Big(\max\Big\{\exp\big(\big[\mu - \tfrac{\sigma^2}{2}\big]T + \sigma W_T^{(i)}\big)\,\xi_i - K_i, 0\Big\}\Big)\Big] \\ &= \int_0^\infty 1 - \left[\prod_{i=1}^d \left(\int_{-\infty}^{\frac{1}{\sigma\sqrt{T}}\left[\ln\left(\frac{y+K_i}{\xi_i}\right) - \left(\mu - [\sigma^2/2]\right)T\right]} \tfrac{1}{\sqrt{2\pi}}\, \exp\!\left(-\tfrac{r^2}{2}\right) dr\right)\right] dy. \end{aligned} \tag{2.1}$$

For the proof of this Proposition, we require the following well-known result.

Lemma 2.2 (Complementary distribution function formula)**.** *Let $\mu \colon \mathcal{B}([0, \infty)) \to [0, \infty]$ be a sigma-finite measure. Then*

$$\int_0^\infty x\,\mu(dx) = \int_0^\infty \mu([x, \infty))\,dx. \tag{2.2}$$

We are now in position to provide a proof of Proposition 2.1.

Proof of Proposition 2.1. The first equality follows directly from the Feynman-Kac formula [22, Corollary 4.17]. We proceed with a proof of the second equality. Throughout this proof let $X_i \colon \Omega \to \mathbb{R}$, $i \in \{1, 2, \ldots, d\}$, be random variables which satisfy for every $i \in \{1, 2, \ldots, d\}$

$$X_i = \exp\big(\big[\mu - \tfrac{\sigma^2}{2}\big]T + \sigma W_T^{(i)}\big)\,\xi_i \tag{2.3}$$

and let $Y \colon \Omega \to \mathbb{R}$ be the random variable given by

$$Y = \max\{X_1 - K_1, \ldots, X_d - K_d, 0\}. \tag{2.4}$$

Observe that for every $y \in (0,\infty)$ it holds

$$\begin{aligned}
\mathbb{P}(Y \geq y) &= 1 - \mathbb{P}(Y < y) = 1 - \mathbb{P}\Big(\max_{i\in\{1,2,\ldots,d\}} (X_i - K_i) < y\Big) \\
&= 1 - \mathbb{P}\big(\cap_{i\in\{1,2,\ldots,d\}} \{X_i - K_i < y\}\big) = 1 - \prod_{i=1}^{d} \mathbb{P}(X_i - K_i < y) \\
&= 1 - \prod_{i=1}^{d} \mathbb{P}(X_i < y + K_i) \\
&= 1 - \prod_{i=1}^{d} \mathbb{P}\Big(\exp\big([\mu - \tfrac{\sigma^2}{2}]T + \sigma W_T^{(i)}\big)\, \xi_i < y + K_i\Big).
\end{aligned} \tag{2.5}$$

Hence, we obtain that for every $y \in (0,\infty)$ it holds

$$\begin{aligned}
\mathbb{P}(Y \geq y) &= 1 - \prod_{i=1}^{d} \mathbb{P}\Big(\exp\big([\mu - \tfrac{\sigma^2}{2}]T + \sigma W_T^{(i)}\big) < \tfrac{y+K_i}{\xi_i}\Big) \\
&= 1 - \prod_{i=1}^{d} \mathbb{P}\Big(\sigma W_T^{(i)} < \ln\Big(\tfrac{y+K_i}{\xi_i}\Big) - \big[\mu - \tfrac{\sigma^2}{2}\big]T\Big) \\
&= 1 - \prod_{i=1}^{d} \mathbb{P}\Big(\tfrac{1}{\sqrt{T}} W_T^{(i)} < \tfrac{1}{\sigma\sqrt{T}}\Big[\ln\Big(\tfrac{y+K_i}{\xi_i}\Big) - \big[\mu - \tfrac{\sigma^2}{2}\big]T\Big]\Big).
\end{aligned} \tag{2.6}$$

This shows that for every $y \in (0,\infty)$ it holds

$$\mathbb{P}(Y \geq y) = 1 - \Bigg[\prod_{i=1}^{d}\Bigg(\textstyle\int_{-\infty}^{\frac{1}{\sigma\sqrt{T}}\left[\ln\left(\frac{y+K_i}{\xi_i}\right) - \left(\mu - [\sigma^2/2]\right)T\right]} \tfrac{1}{\sqrt{2\pi}} \exp\Big(-\tfrac{r^2}{2}\Big)\, dr\Bigg)\Bigg]. \tag{2.7}$$

Combining this with Lemma 2.2 completes the proof of Proposition 2.1. □

With Lemma 2.2 and Proposition 2.1, we may write

$$u(0,x) = \mathbb{E}\Big[\varphi\Big(\exp\Big(\big[\mu - {\sigma^2}/{2}\big]\, T + \sigma W_T^{(1)}\Big)\, x_1, \ldots, \exp\Big(\big[\mu - {\sigma^2}/{2}\big]\, T + \sigma W_T^{(d)}\Big)\, x_d\Big)\Big] \tag{2.8}$$

("semi-explicit" formula). Let us consider the case $\mu = \sigma^2/2$, $T = \sigma = 1$, and $K_1 = \ldots = K_d = K \in (0,\infty)$. Then for every $x = (x_1,\ldots,x_d) \in (0,\infty)^d$

$$\begin{aligned}
u(0,x) &= \mathbb{E}\Big[\varphi\Big(e^{W_T^{(1)}} x_1, \ldots, e^{W_T^{(d)}} x_d\Big)\Big] = \mathbb{E}\Big[\varphi\Big(e^{W_1^{(1)}} x_1, \ldots, e^{W_1^{(d)}} x_d\Big)\Big] \\
&= \mathbb{E}\Big[\max\Big\{e^{W_1^{(1)}} x_1 - K, \ldots, e^{W_1^{(d)}} x_d - K, 0\Big\}\Big] \\
&= \int_0^\infty 1 - \Bigg[\prod_{i=1}^{d} \int_{-\infty}^{\ln(\frac{K+c}{x_i})} \tfrac{1}{\sqrt{2\pi}} \exp\Big(-\tfrac{r^2}{2}\Big)\, dr\Bigg]\, dc.
\end{aligned} \tag{2.9}$$

3 Regularity of the Cumulative Normal Distribution

Now that we have derived an semi-explicit formula for the solution, we establish regularity properties of the integrand function in (2.9). This will be required in order to approximate the multivariate integrals by quadratures (which are subsequently realized by neural networks) in Section 4 and to apply the neural network results from Section 6 to our problem. To this end, we analyze the derivatives of the factors in the tensor product, which essentially are compositions of the cumulative normal distribution with the natural logarithm. As this function appears in numerous closed-form option pricing formulae (see, e.g., [31]), the (Gevrey) type regularity estimates obtained in this section are of independent interest (they may, for example, also be used in the analysis of deep network expression rates and of spectral methods for option pricing).

Lemma 3.1. *Let $f\colon (0,\infty)\to\mathbb{R}$ be the function which satisfies for every $t\in(0,\infty)$ that*

$$f(t)=\tfrac{1}{\sqrt{2\pi}}\int_{-\infty}^{\ln(t)} e^{-\frac{1}{2}r^2}\mathrm{d}r, \tag{3.1}$$

let $g_{n,k}\colon (0,\infty)\to\mathbb{R}$, $n,k\in\mathbb{N}_0$, be the functions which satisfy for every $n,k\in\mathbb{N}_0$, $t\in(0,\infty)$ that

$$g_{n,k}(t)=t^{-n}e^{-\frac{1}{2}[\ln(t)]^2}[\ln(t)]^k, \tag{3.2}$$

and let $(\gamma_{n,k})_{n,k\in\mathbb{Z}}\subseteq\mathbb{Z}$ be the integers which satisfy for every $n,k\in\mathbb{Z}$ that

$$\gamma_{n,k}=\begin{cases}1 & : n=1,k=0\\ -\gamma_{n-1,k-1}-(n-1)\gamma_{n-1,k}+(k+1)\gamma_{n-1,k+1} & : n>1, 0\leq k<n\ .\\ 0 & : \text{else}\end{cases} \tag{3.3}$$

Then it holds for every $n\in\mathbb{N}$ that

(i) we have that f is n-times continuously differentiable and

(ii) we have for every $t\in(0,\infty)$ that

$$f^{(n)}(t)=\tfrac{1}{\sqrt{2\pi}}\left[\sum_{k=0}^{n-1}\gamma_{n,k}\,g_{n,k}(t)\right]. \tag{3.4}$$

Proof of Lemma 3.1. We prove (i) and (ii) by induction on $n\in\mathbb{N}$. For the base case $n=1$ note that (3.1), (3.2), (3.3), the fact that the function $\mathbb{R}\ni r\mapsto e^{-\frac{1}{2}r^2}\in(0,\infty)$ is continuous, the fundamental theorem of calculus, and the chain rule yield

(A) that f is differentiable and

(B) that for every $t\in(0,\infty)$ it holds

$$f'(t)=\tfrac{1}{\sqrt{2\pi}}\,e^{-\frac{1}{2}[\ln(t)]^2}t^{-1}=\tfrac{1}{\sqrt{2\pi}}\,g_{1,0}(t)=\tfrac{1}{\sqrt{2\pi}}\,\gamma_{1,0}\,g_{1,0}(t). \tag{3.5}$$

This establishes (i) and (ii) in the base case $n=1$. For the induction step $\mathbb{N}\ni n\to n+1\in\{2,3,4,\dots\}$ note that for every $t\in(0,\infty)$ we have

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\left[e^{-\frac{1}{2}[\ln(t)]^2}\right]=-t^{-1}e^{-\frac{1}{2}[\ln(t)]^2}\ln(t). \tag{3.6}$$

Combining this and (3.2) with the product rule establishes for every $n\in\mathbb{N}$, $k\in\{0,1,\dots,n-1\}$, $t\in(0,\infty)$ that

$$\begin{aligned}(g_{n,k})'(t)&=\tfrac{\mathrm{d}}{\mathrm{d}t}\left[t^{-n}e^{-\frac{1}{2}[\ln(t)]^2}[\ln(t)]^k\right]\\ &=-nt^{-(n+1)}e^{-\frac{1}{2}[\ln(t)]^2}[\ln(t)]^k-t^{-(n+1)}e^{-\frac{1}{2}[\ln(t)]^2}[\ln(t)]^{k+1}\\ &\quad+t^{-(n+1)}e^{-\frac{1}{2}[\ln(t)]^2}k[\ln(t)]^{\max\{k-1,0\}}\\ &=-g_{n+1,k+1}(t)-ng_{n+1,k}(t)+kg_{n+1,\max\{k-1,0\}}(t).\end{aligned} \tag{3.7}$$

Hence, we obtain that for every $n\in\mathbb{N}$, $t\in(0,\infty)$ it holds

$$\begin{aligned}&\sum_{k=0}^{n-1}\gamma_{n,k}(g_{n,k})'(t)\\ &=\sum_{k=0}^{n-1}\left[\gamma_{n,k}\left(-g_{n+1,k+1}(t)-ng_{n+1,k}(t)+kg_{n+1,\max\{k-1,0\}}(t)\right)\right]\\ &=\sum_{k=0}^{n-1}-\gamma_{n,k}\,g_{n+1,k+1}(t)+\sum_{k=0}^{n-1}-n\gamma_{n,k}\,g_{n+1,k}(t)+\sum_{k=1}^{n-1}k\gamma_{n,k}\,g_{n+1,\max\{k-1,0\}}(t)\\ &=\sum_{k=1}^{n}-\gamma_{n,k-1}\,g_{n+1,k}(t)+\sum_{k=0}^{n-1}-n\gamma_{n,k}\,g_{n+1,k}(t)+\sum_{k=0}^{n-2}(k+1)\gamma_{n,k+1}\,g_{n+1,k}(t).\end{aligned} \tag{3.8}$$

The fact that for every $n \in \mathbb{N}$ it holds that $\gamma_{n,-1} = \gamma_{n,n} = \gamma_{n,n+1} = 0$ and (3.3) therefore ensure that for every $n \in \mathbb{N}$, $t \in (0, \infty)$ we have

$$\begin{aligned} \sum_{k=0}^{n-1} \gamma_{n,k} (g_{n,k})'(t) &= \sum_{k=0}^{n} [(-\gamma_{n,k-1} - n\gamma_{n,k} + (k+1)\gamma_{n,k+1})\, g_{n+1,k}(t)] \\ &= \sum_{k=0}^{n} \gamma_{n+1,k}\, g_{n+1,k}(t). \end{aligned} \tag{3.9}$$

Induction thus establishes (i) and (ii). The proof of Lemma 3.1 is thus completed. □

Using the recursive formula from above we can now bound the derivatives of f. Note that the supremum of $f^{(n)}$ is actually attained on the interval $[e^{-4n}, 1]$ and scales with n like $e^{(cn^2)}$ for some $c \in (0, \infty)$. This can directly be seen by calulating the maximum of the $g_{n,k}$ from (3.2). For our purposes, however, it is sufficient to establish that all derivatives of f are bounded on $(0, \infty)$.

Lemma 3.2. *Let $f\colon (0, \infty) \to \mathbb{R}$ be the function which satisfies for every $t \in (0, \infty)$ that*

$$f(t) = \tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t)} e^{-\frac{1}{2}r^2} \mathrm{d}r. \tag{3.10}$$

Then it holds for every $n \in \mathbb{N}$ that

$$\sup_{t \in (0,\infty)} \left| f^{(n)}(t) \right| \leq \max\left\{ (n-1)!\, 2^{n-2} \,,\, \sup_{t \in [e^{-4n},1]} \left| f^{(n)}(t) \right| \right\} < \infty. \tag{3.11}$$

Proof of Lemma 3.2. Throughout this proof let $g_{n,k}\colon (0, \infty) \to \mathbb{R}$, $n, k \in \mathbb{N}_0$, be the functions introduced in (3.2) and let $(\gamma_{n,k})_{n,k \in \mathbb{Z}} \subseteq \mathbb{Z}$ be the integers introduced in (3.3). Then Lemma 3.1 shows for every $n \in \mathbb{N}$ that

(a) we have that f is n-times continuously differentiable and

(b) we have for every $t \in (0, \infty)$ that

$$f^{(n)}(t) = \tfrac{1}{\sqrt{2\pi}} \left[\sum_{k=0}^{n-1} \gamma_{n,k}\, g_{n,k}(t) \right]. \tag{3.12}$$

In addition, observe that for every $m \in \mathbb{N}$, $t \in (0, e^{-2m}]$ holds $\frac{1}{2}\ln(t) \leq -m$. This ensures that for every $m \in \mathbb{N}$, $t \in (0, e^{-2m}] \subseteq (0, 1]$ we have

$$e^{-\frac{1}{2}[\ln(t)]^2} = e^{\left[\ln(t)(-\frac{1}{2}\ln(t))\right]} = \left[e^{\ln(t)}\right]^{-\frac{1}{2}\ln(t)} = t^{-\frac{1}{2}\ln(t)} = \left(\tfrac{1}{t}\right)^{\frac{1}{2}\ln(t)} \leq \left(\tfrac{1}{t}\right)^{-m} = t^m. \tag{3.13}$$

Moreover, note that the fundamental theorem of calculus implies for every $t \in (0, 1]$ that

$$|\ln(t)| = |\ln(t) - \ln(1)| = |\ln(1) - \ln(t)| = \left| \int_t^1 \frac{1}{s}\, \mathrm{d}s \right| \leq \left| \frac{1}{t}(1-t) \right| \leq t^{-1}. \tag{3.14}$$

Combining (3.2), (3.12), and (3.13) therefore establishes that for every $n \in \mathbb{N}$, $t \in (0, e^{-4n}) \subseteq (0, 1]$ it holds

$$\begin{aligned} \left| f^{(n)}(t) \right| &= \tfrac{1}{\sqrt{2\pi}} \left| \sum_{k=0}^{n-1} \gamma_{n,k}\, g_{n,k}(t) \right| = \tfrac{1}{\sqrt{2\pi}} \left| \sum_{k=0}^{n-1} \gamma_{n,k} t^{-n} e^{-\frac{1}{2}[\ln(t)]^2} [\ln(t)]^k \right| \\ &\leq \tfrac{1}{\sqrt{2\pi}} \left[\sum_{k=0}^{n-1} |\gamma_{n,k}|\, t^{n-k} \right] \leq \tfrac{1}{\sqrt{2\pi}} \left[\sum_{k=0}^{n-1} |\gamma_{n,k}| \right]. \end{aligned} \tag{3.15}$$

In addition, observe that the fundamental theorem of calculus ensures that for every $t \in [1,\infty)$ we have

$$|\ln(t)| = |\ln(t) - \ln(1)| = \left|\int_1^t \frac{1}{s}\,\mathrm{d}s\right| \leq |t-1| \leq t. \tag{3.16}$$

This, (3.2), (3.12), and the fact that for every $t \in (0,\infty)$ it holds $|e^{-\frac{1}{2}[\ln(t)]^2}| \leq 1$ imply that for every $n \in \mathbb{N}$, $t \in (1,\infty)$ we have

$$\begin{aligned}\left|f^{(n)}(t)\right| &= \tfrac{1}{\sqrt{2\pi}}\left|\sum_{k=0}^{n-1}\gamma_{n,k}\,g_{n,k}(t)\right| = \tfrac{1}{\sqrt{2\pi}}\left|\sum_{k=0}^{n-1}\gamma_{n,k}t^{-n}e^{-\frac{1}{2}[\ln(t)]^2}[\ln(t)]^k\right| \\ &\leq \tfrac{1}{\sqrt{2\pi}}\left[\sum_{k=0}^{n-1}|\gamma_{n,k}|\,t^{-n}\,|\ln(t)|^k\right] \leq \tfrac{1}{\sqrt{2\pi}}\left[\sum_{k=0}^{n-1}|\gamma_{n,k}|\,t^{-n}t^k\right] \\ &= \tfrac{1}{\sqrt{2\pi}}\left[\sum_{k=0}^{n-1}|\gamma_{n,k}|\,t^{-n+k}\right] \leq \tfrac{1}{\sqrt{2\pi}}\left[\sum_{k=0}^{n-1}|\gamma_{n,k}|\right].\end{aligned} \tag{3.17}$$

Moreover, observe that (a) assures that for every $n \in \mathbb{N}$ it holds that the function $f^{(n)}$ is continuous. This and the boundedness of the set $[e^{-4n},1]$ ensure that for every $n \in \mathbb{N}$ we have

$$\sup_{t\in[e^{-4n},1]}\left|f^{(n)}(t)\right| < \infty. \tag{3.18}$$

Combining this with (3.15) and (3.17) establishes that for every $n \in \mathbb{N}$ we have

$$\sup_{t\in(0,\infty)}\left|f^{(n)}(t)\right| \leq \max\left\{\tfrac{1}{\sqrt{2\pi}}\left[\sum_{k=0}^{n-1}|\gamma_{n,k}|\right], \sup_{t\in[e^{-4n},1]}\left|f^{(n)}(t)\right|\right\} < \infty. \tag{3.19}$$

Furthermore, note that (3.3) implies that for every $n \in \{2,3,4,\dots\}$ it holds

$$\begin{aligned}\sum_{k=0}^{n-1}|\gamma_{n,k}| &= \sum_{k=0}^{n-1}|-\gamma_{n-1,k-1} - (n-1)\gamma_{n-1,k} + (k+1)\gamma_{n-1,k+1}| \\ &\leq \left[\sum_{k=0}^{n-1}|\gamma_{n-1,k-1}|\right] + \left[\sum_{k=0}^{n-1}(n-1)\,|\gamma_{n-1,k}|\right] + \left[\sum_{k=0}^{n-1}(k+1)\,|\gamma_{n-1,k+1}|\right] \\ &= \left[\sum_{k=-1}^{n-2}|\gamma_{n-1,k}|\right] + \left[\sum_{k=0}^{n-1}(n-1)\,|\gamma_{n-1,k}|\right] + \left[\sum_{k=1}^{n}k\,|\gamma_{n-1,k}|\right].\end{aligned} \tag{3.20}$$

Combining this with the fact that for every $n \in \{2,3,4,\dots\}$, $k \in \mathbb{Z}\backslash\{0,1,\dots,n-2\}$ we have $\gamma_{n-1,k} = 0$ implies that for every $n \in \{2,3,4,\dots\}$ it holds

$$\sum_{k=0}^{n-1}|\gamma_{n,k}| = \sum_{k=0}^{n-2}[(1+(n-1)+k)\,|\gamma_{n-1,k}|] \leq (2n-2)\left[\sum_{k=0}^{n-2}|\gamma_{n-1,k}|\right] = 2(n-1)\left[\sum_{k=0}^{n-2}|\gamma_{n-1,k}|\right]. \tag{3.21}$$

The fact that $\gamma_{1,0} = 1$ hence implies that for every $n \in \mathbb{N}$ we have

$$\sum_{k=0}^{n-1}|\gamma_{n,k}| \leq (n-1)!\,2^{n-1}\left[\sum_{k=0}^{0}|\gamma_{1,k}|\right] = (n-1)!\,2^{n-1}. \tag{3.22}$$

Combining this and (3.19) ensures that for every $n \in \mathbb{N}$ it holds

$$\sup_{t\in(0,\infty)}\left|f^{(n)}(t)\right| \leq \max\left\{\tfrac{1}{\sqrt{2\pi}}(n-1)!\,2^{n-1}, \sup_{t\in[e^{-4n},1]}\left|f^{(n)}(t)\right|\right\} < \infty. \tag{3.23}$$

The proof of Lemma 3.2 is thus completed. □

In the following corollary we estimate the derivatives of the function $x \to f(\frac{K+c}{x})$ required to approximate this function by neural networks.

Corollary 3.3. *Let $n \in \mathbb{N}$, $K \in [0, \infty)$, $c, a \in (0, \infty)$, $b \in (a, \infty)$, let $f\colon (0, \infty) \to \mathbb{R}$ be the function which satisfies for every $t \in (0, \infty)$ that*

$$f(t) = \tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t)} e^{-\frac{1}{2}r^2} \mathrm{d}r, \tag{3.24}$$

and let $h\colon [a, b] \to \mathbb{R}$ be the function which satisfies for every $x \in [a, b]$ that

$$h(x) = f(\tfrac{K+c}{x}). \tag{3.25}$$

Then it holds

(i) that f and h are infinitely often differentiable and

(ii) that

$$\max_{k\in\{0,1,\ldots,n\}} \sup_{x\in[a,b]} \left|h^{(k)}(x)\right| \leq n2^{n-1}n! \left[\max_{k\in\{0,1,\ldots,n\}} \sup_{t\in[\frac{K+c}{b},\frac{K+c}{a}]} \left|f^{(k)}(t)\right|\right] \max\{a^{-2n}, 1\} \max\{(K+c)^n, 1\}. \tag{3.26}$$

Proof of Corollary 3.3. Throughout this proof let $\alpha_{m,j} \in \mathbb{Z}$, $m, j \in \mathbb{Z}$, be the integers which satisfy that for every $m, j \in \mathbb{Z}$ it holds

$$\alpha_{m,j} = \begin{cases} -1 & : m = j = 1 \\ -(m-1+j)\alpha_{m-1,j} - \alpha_{m-1,j-1} & : m > 1,\ 1 \leq j \leq m \ . \\ 0 & : \text{else} \end{cases} \tag{3.27}$$

Note that Lemma 3.1 and the chain rule ensure that the functions f and h are infinitely often differentiable. Next we claim that for every $m \in \mathbb{N}$, $x \in [a, b]$ it holds

$$h^{(m)}(x) = \tfrac{\mathrm{d}^m}{\mathrm{d}x^m}\big(f(\tfrac{K+c}{x})\big) = \sum_{j=1}^{m} \alpha_{m,j}(K+c)^j x^{-(m+j)}\big(f^{(j)}(\tfrac{K+c}{x})\big). \tag{3.28}$$

We prove (3.28) by induction on $m \in \mathbb{N}$. To prove the base case $m = 1$ we note that the chain rule ensures that for every $x \in [a, b]$ we have

$$\tfrac{\mathrm{d}}{\mathrm{d}x}\big(f(\tfrac{K+c}{x})\big) = -(K+c)x^{-2}\big(f'(\tfrac{K+c}{x})\big) = \alpha_{1,1}(K+c)x^{-2}\big(f'(\tfrac{K+c}{x})\big)\,. \tag{3.29}$$

This establishes (3.28) in the base case $m = 1$. For the induction step $\mathbb{N} \ni m \to m+1 \in \mathbb{N}$ observe that the chain rule implies for every $m \in \mathbb{N}$, $x \in [a, b]$ that

$$\begin{aligned}
&\tfrac{\mathrm{d}}{\mathrm{d}x}\left[\sum_{j=1}^{m} \alpha_{m,j}(K+c)^j x^{-(m+j)}\left(f^{(j)}(\tfrac{K+c}{x})\right)\right] \\
&= -\left[\sum_{j=1}^{m} \alpha_{m,j}(K+c)^{j+1} x^{-(m+j+2)}\left(f^{(j+1)}(\tfrac{K+c}{x})\right)\right] - \left[\sum_{j=1}^{m} \alpha_{m,j}(K+c)^j(m+j)x^{-(m+j+1)}\left(f^{(j)}(\tfrac{K+c}{x})\right)\right] \\
&= -\left[\sum_{j=2}^{m+1} \alpha_{m,j-1}(K+c)^j x^{-(m+j+1)}\left(f^{(j)}(\tfrac{K+c}{x})\right)\right] - \left[\sum_{j=1}^{m} \alpha_{m,j}(K+c)^j(m+j)x^{-(m+j+1)}\left(f^{(j)}(\tfrac{K+c}{x})\right)\right] \\
&= \sum_{j=1}^{m+1} (-(m+j)\alpha_{m,j} - \alpha_{m,j-1})(K+c)^j x^{-(m+1+j)}\left(f^{(j)}(\tfrac{K+c}{x})\right).
\end{aligned} \tag{3.30}$$

Induction thus establishes (3.28). Next note that (3.27) ensures that for every $m \in \{2, 3, \dots\}$ it holds

$$\begin{aligned} \max_{j\in\{1,2,\dots,m\}} |\alpha_{m,j}| &= \max_{j\in\{1,2,\dots,m\}} |-(m-1+j)\alpha_{m-1,j} - \alpha_{m-1,j-1}| \\ &\leq \left[\max_{j\in\{1,2,\dots,m-1\}} |(m-1+j)\alpha_{m-1,j}|\right] + \left[\max_{j\in\{1,2,\dots,m-1\}} |\alpha_{m-1,j}|\right] \\ &\leq (2m-1)\left[\max_{j\in\{1,2,\dots,m-1\}} |\alpha_{m-1,j}|\right] \leq 2m\left[\max_{j\in\{1,2,\dots,m-1\}} |\alpha_{m-1,j}|\right]. \end{aligned} \tag{3.31}$$

Induction hence proves that for every $m \in \mathbb{N}$ we have $\max_{j\in\{1,2,\dots,m\}} |\alpha_{m,j}| \leq 2^{m-1} m!$. Combining this with (3.28) implies that for every $m \in \{1, 2, \dots, n\}$, $x \in [a, b]$ we have

$$\begin{aligned} \left|h^{(m)}(x)\right| &= \left|\sum_{j=1}^{m} \alpha_{m,j}(K+c)^j x^{-(m+j)}\big(f^{(j)}(\tfrac{K+c}{x})\big)\right| \\ &\leq 2^{m-1}m!\left[\max_{j\in\{1,2,\dots,m\}} \sup_{t\in[\frac{K+c}{b},\frac{K+c}{a}]} \left|f^{(j)}(t)\right|\right] \max\{x^{-2m}, 1\}\left[\sum_{j=1}^{m}(K+c)^j\right] \\ &\leq m2^{m-1}m!\left[\max_{j\in\{1,2,\dots,m\}} \sup_{t\in[\frac{K+c}{b},\frac{K+c}{a}]} \left|f^{(j)}(t)\right|\right] \max\{x^{-2m}, 1\}\max\{(K+c)^m, 1\}. \end{aligned} \tag{3.32}$$

Combining this with the fact that $\sup_{x\in[a,b]} |h(x)| = \sup_{t\in[\frac{K+c}{b},\frac{K+c}{a}]} |f(t)|$ establishes that it holds

$$\max_{k\in\{0,1,\dots,n\}} \sup_{x\in[a,b]} \left|h^{(k)}(x)\right| \leq n2^{n-1}n!\left[\max_{k\in\{0,1,\dots,n\}} \sup_{t\in[\frac{K+c}{b},\frac{K+c}{a}]} \left|f^{(k)}(t)\right|\right] \max\{a^{-2n}, 1\}\max\{(K+c)^n, 1\}. \tag{3.33}$$

This completes the proof of Corollary 3.3. □

Next we consider the derivatives of the functions $c \mapsto f(\frac{K+c}{x_i})$, $i \in \{1, 2, \dots, d\}$, and their tensor product, which will be needed in order to approximate approximate the outer integral in (2.9) by composite Gaussian quadrature.

Corollary 3.4. *Let $n \in \mathbb{N}$, $K \in [0, \infty)$, $x \in (0, \infty)$, let $f\colon (0, \infty) \to \mathbb{R}$ be the function which satisfies for every $t \in (0, \infty)$ that*

$$f(t) = \tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t)} e^{-\frac{1}{2}r^2} \mathrm{d}r, \tag{3.34}$$

and let $g\colon (0, \infty) \to \mathbb{R}$ be the function which satisfies for every $t \in (0, \infty)$ that

$$g(t) = f\big(\tfrac{K+t}{x}\big). \tag{3.35}$$

Then it holds

(i) that f and g are infinitely often differentiable and

(ii) that

$$\sup_{t\in(0,\infty)} \left|g^{(n)}(t)\right| \leq \left[\sup_{t\in(0,\infty)} \left|f^{(n)}(t)\right|\right] |x|^{-n} < \infty. \tag{3.36}$$

Proof of Corollary 3.4. Combining Lemma 3.2 with the chain rule implies that for every $t \in (0, \infty)$ it holds

$$\left|g^{(n)}(t)\right| = \left|\tfrac{\mathrm{d}^n}{\mathrm{d}t^n}\big(f(\tfrac{K+t}{x})\big)\right| = \left|f^{(n)}\big(\tfrac{K+t}{x}\big)\,\tfrac{1}{x^n}\right| \leq \left[\sup_{t\in(0,\infty)} \left|f^{(n)}(t)\right|\right] |x|^{-n} < \infty. \tag{3.37}$$

This completes the proof of Corollary 3.4. □

Lemma 3.5. *Let $d, n \in \mathbb{N}$, $a \in (0,\infty)$, $b \in (a,\infty)$, $K = (K_1,\dots,K_d) \in [0,\infty)^d$, $x = (x_1,\dots,x_d) \in [a,b]^d$, let $f\colon (0,\infty) \to \mathbb{R}$ be the function which satisfies for every $t \in (0,\infty)$ that*

$$f(t) = \tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t)} e^{-\frac{1}{2}r^2}\,\mathrm{d}r, \tag{3.38}$$

and let $F\colon (0,\infty) \to \mathbb{R}$ be the function which satisfies for every $c \in (0,\infty)$ that

$$F(c) = 1 - \left[\prod_{i=1}^{d} f\!\left(\tfrac{K_i+c}{x_i}\right)\right]. \tag{3.39}$$

Then it holds

(i) that f and F are infinitely often differentiable and

(ii) that

$$\sup_{c\in(0,\infty)} \left|F^{(n)}(c)\right| \leq \left[\max_{k\in\{0,1,\dots,n\}} \sup_{t\in(0,\infty)} \left|f^{(k)}(t)\right|\right]^n d^n a^{-n} < \infty. \tag{3.40}$$

Proof of Lemma 3.5. Note that Lemma 3.1 ensures that f and F are infinitely often differentiable. Moreover, observe that (3.39) and the general Leibniz rule imply for every $c \in (0,\infty)$ that

$$\begin{aligned} F^{(n)}(c) &= -\tfrac{\mathrm{d}^n}{\mathrm{d}c^n}\left[\prod_{i=1}^{d} f\!\left(\tfrac{K_i+c}{x_i}\right)\right] \\ &= -\sum_{\substack{l_1,l_2,\dots,l_d\in\mathbb{N}_0,\\ \sum_{i=1}^{d} l_i = n}} \left[\binom{n}{l_1,l_2,\dots,l_d} \prod_{i=1}^{d} \left(\tfrac{\mathrm{d}^{l_i}}{\mathrm{d}c^{l_i}}\left[f\!\left(\tfrac{K_i+c}{x_i}\right)\right]\right)\right]. \end{aligned} \tag{3.41}$$

Next note that the fact that for every $r \in \mathbb{R}$ it holds that $e^{-\frac{1}{2}r^2} \geq 0$ ensures that

$$\sup_{t\in(0,\infty)} |f(t)| = \sup_{t\in(0,\infty)} \left|\tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t)} e^{-\frac{1}{2}r^2}\,\mathrm{d}r\right| = \left|\tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}r^2}\,\mathrm{d}r\right| = 1. \tag{3.42}$$

Corollary 3.4 hence establishes that for every $c \in [0,\infty)$, $l_1,\dots,l_d \in \mathbb{N}_0$ with $\sum_{i=1}^{d} l_i = n$ it holds

$$\begin{aligned} \left|\prod_{i=1}^{d} \left(\tfrac{\mathrm{d}^{l_i}}{\mathrm{d}c^{l_i}}\left[f\!\left(\tfrac{K_i+c}{x_i}\right)\right]\right)\right| &\leq \prod_{i=1}^{d} \left(\left[\sup_{t\in(0,\infty)} \left|f^{(l_i)}(t)\right|\right] |x_i|^{-l_i}\right) \\ &= \left[\prod_{i=1}^{d} |x_i|^{-l_i}\right] \left[\prod_{i=1}^{d} \left(\sup_{t\in(0,\infty)} \left|f^{(l_i)}(t)\right|\right)\right] \\ &\leq \left[\prod_{i=1}^{d} |x_i|^{-l_i}\right] \left[\prod_{\substack{i\in\{1,2,\dots,d\},\\ l_i>0}} \left(\max_{k\in\{1,2,\dots,n\}} \sup_{t\in(0,\infty)} \left|f^{(k)}(t)\right|\right)\right] \\ &\leq \left[\prod_{i=1}^{d} |x_i|^{-l_i}\right] \left[\prod_{\substack{i\in\{1,2,\dots,d\},\\ l_i>0}} \max\left\{1, \max_{k\in\{1,2,\dots,n\}} \sup_{t\in(0,\infty)} \left|f^{(k)}(t)\right|\right\}\right] \\ &\leq \left[\prod_{i=1}^{d} |x_i|^{-l_i}\right] \left[\max\left\{1, \max_{k\in\{1,2,\dots,n\}} \sup_{t\in(0,\infty)} \left|f^{(k)}(t)\right|\right\}\right]^{(l_1+\ldots+l_d)} \\ &= \left[\prod_{i=1}^{d} |x_i|^{-l_i}\right] \left[\max_{k\in\{0,1,\dots,n\}} \sup_{t\in(0,\infty)} \left|f^{(k)}(t)\right|\right]^n. \end{aligned} \tag{3.43}$$

Moreover, note that the multinomial theorem ensures that

$$d^n = \left[\sum_{i=1}^{d} 1\right]^n = \sum_{\substack{l_1,l_2,\dots,l_d\in\mathbb{N}_0,\\ \sum_{i=1}^d l_i=n}} \left[\binom{n}{l_1,l_2,\dots,l_d}\prod_{i=1}^{d} 1^{l_i}\right] = \sum_{\substack{l_1,l_2,\dots,l_d\in\mathbb{N}_0,\\ \sum_{i=1}^d l_i=n}} \left[\binom{n}{l_1,l_2,\dots,l_d}\right]. \tag{3.44}$$

Combining this with (3.41), (3.43), and the assumption that $x\in[a,b]^d$ implies that for every $c\in(0,\infty)$ we have

$$\begin{aligned}\left|F^{(n)}(c)\right| &\le \left|\sum_{\substack{l_1,l_2,\dots,l_d\in\mathbb{N}_0,\\ \sum_{i=1}^d l_i=n}} \left[\binom{n}{l_1,l_2,\dots,l_d}\left[\prod_{i=1}^{d}|x_i|^{-l_i}\right]\left[\max_{k\in\{0,1,\dots,n\}}\sup_{t\in(0,\infty)}\left|f^{(k)}(t)\right|\right]^n\right]\right| \\ &\le a^{-n}\left[\max_{k\in\{0,1,\dots,n\}}\sup_{t\in(0,\infty)}\left|f^{(k)}(t)\right|\right]^n\left|\sum_{\substack{l_1,l_2,\dots,l_d\in\mathbb{N}_0,\\ \sum_{i=1}^d l_i=n}}\binom{n}{l_1,l_2,\dots,l_d}\right| \\ &= a^{-n}\left[\max_{k\in\{0,1,\dots,n\}}\sup_{t\in(0,\infty)}\left|f^{(k)}(t)\right|\right]^n d^n.\end{aligned} \tag{3.45}$$

This completes the proof of Lemma 3.5. □

4 Quadrature

To approximate the function $x\mapsto u(0,x)$ from (2.9) by a neural network we need to evaluate, for arbitrary, given x, an expression of the form $\int_0^\infty F_x(c)\mathrm{d}c$ with F_x as defined in Lemma 4.2. We achieve this by proving in Lemma 4.2 that the functions F_x decay sufficiently fast for $c\to\infty$, and then employ numerical integration to show that the definite integral $\int_0^N F_x(c)\mathrm{d}c$ can be sufficiently well approximated by a weighted sum of $F_x(c_j)$ for suitable quadrature points $c_j\in(0,N)$. The representation of such a sum can be realized by neural networks. We show in Section 6 and 7 how the functions $x\mapsto F_x(c_j)$ for $(c_j)\in(0,N)$ can be realized efficiently due to their tensor product structure. We start by recalling an error bound for composite Gaussian quadrature which is explicit in the stepsize and quadrature order.

Lemma 4.1. *Let $n,M\in\mathbb{N}$, $N\in(0,\infty)$. Then there exist real numbers $(c_j)_{j=1}^{nM}\subseteq(0,N)$ and $(w_j)_{j=1}^{nM}\subseteq(0,\infty)$ such that for every $h\in C^{2n}([0,N],\mathbb{R})$ it holds*

$$\left|\int_0^N h(t)\,\mathrm{d}t-\sum_{j=1}^{nM}w_j h(c_j)\right| \le \tfrac{1}{(2n)!}N^{2n+1}M^{-2n}\left[\sup_{\xi\in[0,N]}\left|h^{(2n)}(\xi)\right|\right]. \tag{4.1}$$

Proof of Lemma 4.1. Throughout this proof let $h\in C^{2n}([0,N],\mathbb{R})$ and $\alpha_k\in[0,N]$, $k\in\{0,1,\dots,M\}$, such that for every $k\in\{0,1,\dots,M\}$ it holds $\alpha_k=\frac{kN}{M}$. Observe that [32, Theorems 4.17, 6.11, and 6.12] ensure that for every $k\in\{0,1,\dots,M-1\}$ there exist $(\gamma_i^k)_{i=1}^n\subseteq(\alpha_k,\alpha_{k+1})$, $(\omega_i^k)_{i=1}^n\subseteq(0,\infty)$, and $\xi^k\in[\alpha_k,\alpha_{k+1}]$ such that

$$\int_{\alpha_k}^{\alpha_{k+1}} h(t)\,\mathrm{d}t-\sum_{i=1}^{n}\omega_i^k h(\gamma_i^k)=\frac{h^{(2n)}(\xi^k)}{(2n)!}\int_{\alpha_k}^{\alpha_{k+1}}\left[\prod_{i=1}^{n}(t-\gamma_i^k)^2\right]\mathrm{d}t. \tag{4.2}$$

Next note that for every $k\in\{0,1,\dots,M-1\}$ it holds

$$\int_{\alpha_k}^{\alpha_{k+1}}\left[\prod_{i=1}^{n}(t-\gamma_i^k)^2\right]\mathrm{d}t\le\int_{\alpha_k}^{\alpha_{k+1}}\left[\prod_{i=1}^{n}(\alpha_k-\alpha_{k+1})^2\right]\mathrm{d}t=\left[\tfrac{N}{M}\right]^{2n+1}. \tag{4.3}$$

Combining this with (4.2) yields that for every $k \in \{0, 1, \ldots, M\}$ we have

$$\left| \int_{\alpha_k}^{\alpha_{k+1}} h(t)\,\mathrm{d}t - \sum_{i=1}^{n} \omega_i^k h(\gamma_i^k) \right| \leq \frac{\left| h^{(2n)}(\xi^k) \right|}{(2n)!} \left[\tfrac{N}{M} \right]^{2n+1} \leq \tfrac{1}{(2n)!} \left[\tfrac{N}{M} \right]^{2n+1} \left[\sup_{\xi \in [0,N]} \left| h^{(2n)}(\xi) \right| \right]. \tag{4.4}$$

Hence, we obtain

$$\begin{aligned} \left| \int_0^N h(t)\,\mathrm{d}t - \sum_{k=0}^{M-1} \sum_{i=1}^{n} \omega_i^k h(\gamma_i^k) \right| &= \left| \sum_{k=0}^{M-1} \left[\int_{\alpha_k}^{\alpha_{k+1}} h(t)\,\mathrm{d}t - \sum_{i=1}^{n} \omega_i^k h(\gamma_i^k) \right] \right| \\ &\leq \sum_{k=0}^{M-1} \left(\tfrac{1}{(2n)!} \left(\tfrac{N}{M} \right)^{2n+1} \left[\sup_{\xi \in [0,N]} \left| h^{(2n)}(\xi) \right| \right] \right) \\ &= \tfrac{1}{(2n)!} N^{2n+1} M^{-2n} \left[\sup_{\xi \in [0,N]} \left| h^{(2n)}(\xi) \right| \right]. \end{aligned} \tag{4.5}$$

Let $(c_j)_{j=1}^{nM} \subseteq (0, N)$, $(w_j)_{j=1}^{nM} \subseteq (0, \infty)$ such that for every $i \in \{1, 2, \ldots, n\}$, $k \in \{0, 1, \ldots, M-1\}$ it holds

$$c_{kn+i} = \gamma_i^k \quad \text{and} \quad w_{kn+i} = \omega_i^k. \tag{4.6}$$

Next observe that

$$\left| \int_0^N h(t)\,\mathrm{d}t - \sum_{j=1}^{nM} w_j h(c_j) \right| = \left| \int_0^N h(t)\,\mathrm{d}t - \sum_{k=0}^{M-1} \sum_{i=1}^{n} \omega_i^k h(\gamma_i^k) \right|. \tag{4.7}$$

This completes the proof of Lemma 4.1. □

In the following we bound the error due to truncating the domain of integration.

Lemma 4.2. *Let $d, n \in \mathbb{N}$, $a \in (0, \infty)$, $b \in (a, \infty)$, $K = (K_1, K_2, \ldots, K_d) \in [0, \infty)^d$, let $F_x \colon (0, \infty) \to \mathbb{R}$, $x \in [a, b]^d$, be the functions which satisfy for every $x = (x_1, x_2, \ldots, x_d) \in [a, b]^d$, $c \in (0, \infty)$ that*

$$F_x(c) = 1 - \prod_{i=1}^{d} \left[\tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(\frac{K_i + c}{x_i})} e^{-\frac{1}{2} r^2} \mathrm{d}r \right], \tag{4.8}$$

and for every $\varepsilon \in (0, 1]$ let $N_\varepsilon \in \mathbb{R}$ be given by $N_\varepsilon = 2e^{2(n+1)}(b+1)^{1+\frac{1}{n}} d^{\frac{1}{n}} \varepsilon^{-\frac{1}{n}}$. Then it holds for every $\varepsilon \in (0, 1]$ that

$$\sup_{x \in [a,b]^d} \left| \int_{N_\varepsilon}^{\infty} F_x(c)\,\mathrm{d}c \right| \leq \varepsilon. \tag{4.9}$$

Proof of Lemma 4.2. Throughout this proof let $g \colon (0, \infty) \to (0, 1)$ be the function given by

$$g(t) = 1 - \tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t)} e^{-\frac{1}{2} r^2} \mathrm{d}r. \tag{4.10}$$

Note that [6, Eq.(5)] ensures that for every $y \in [0, \infty)$ we have $\frac{2}{\sqrt{\pi}} \int_y^\infty e^{-r^2} \mathrm{d}r \leq e^{-y^2}$. This implies for every $t \in [1, \infty)$ that

$$0 < g(t) = 1 - \tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t)} e^{-\frac{1}{2} r^2} \mathrm{d}r = \tfrac{1}{\sqrt{2\pi}} \int_{\ln(t)}^{\infty} e^{-\frac{1}{2} r^2} \mathrm{d}r = \tfrac{1}{\sqrt{\pi}} \int_{\frac{\ln(t)}{\sqrt{2}}}^{\infty} e^{-r^2} \mathrm{d}r \leq \tfrac{1}{2} e^{-\frac{1}{2} [\ln(t)]^2}. \tag{4.11}$$

Furthermore, observe that for every $t \in [e^{2(n+1)}, \infty)$ it holds

$$e^{-\frac{1}{2} [\ln(t)]^2} = e^{\left[\ln(t)(-\frac{1}{2} \ln(t))\right]} = \left[e^{\ln(t)} \right]^{-\frac{1}{2} \ln(t)} = t^{-\frac{1}{2} \ln(t)} \leq t^{-(n+1)}. \tag{4.12}$$

This, (4.11), and the fact that for every $\varepsilon \in (0,1]$, $c \in [N_\varepsilon, \infty)$, $x \in [a,b]^d$, $i \in \{1,2,\ldots,d\}$ we have $\frac{K_i+c}{x_i} \geq \frac{c}{b} \geq e^{2(n+1)} \geq 1$ imply that for every $\varepsilon \in (0,1]$, $c \in [N_\varepsilon, \infty)$, $x \in [a,b]^d$ it holds

$$\begin{aligned} |F_x(c)| &= \left|1 - \prod_{i=1}^{d} \left[\tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(\frac{K_i+c}{x_i})} e^{-\frac{1}{2}r^2} \mathrm{d}r\right]\right| = \left|1 - \prod_{i=1}^{d} \left[1 - g(\tfrac{K_i+c}{x_i})\right]\right| \\ &\leq \left|1 - \prod_{i=1}^{d} \left[1 - \tfrac{1}{2}\left[\tfrac{K_i+c}{x_i}\right]^{-(n+1)}\right]\right| \leq \left|1 - \prod_{i=1}^{d} \left[1 - \tfrac{1}{2}\left[\tfrac{c}{b}\right]^{-(n+1)}\right]\right|. \end{aligned} \tag{4.13}$$

Combining this with the binomial theorem and the fact that for every $i \in \{1,2,\ldots,d\}$ we have $\binom{d}{i} \leq \frac{d^i}{i!} \leq \frac{d^i}{\exp(i\ln(i)-i+1)} \leq \frac{(de)^i}{i^i}$ establishes that for every $\varepsilon \in (0,1]$, $c \in [N_\varepsilon, \infty)$, $x \in [a,b]^d$ it holds

$$\begin{aligned} |F_x(c)| &\leq \left|1 - \left(1 - \tfrac{1}{2}\left[\tfrac{c}{b}\right]^{-(n+1)}\right)^d\right| = \left|1 - \sum_{i=0}^{d} \left[\binom{d}{i} \left[-\tfrac{1}{2}\left[\tfrac{c}{b}\right]^{-(n+1)}\right]^i\right]\right| \\ &\leq \sum_{i=1}^{d} \left[\binom{d}{i} \left[\tfrac{1}{2}\right]^i \left[\tfrac{b}{c}\right]^{(n+1)i}\right] \leq \sum_{i=1}^{d} \left[\tfrac{de}{2i}\right]^i \left[\tfrac{b}{c}\right]^{(n+1)i} \\ &= \sum_{i=1}^{d} \left[\tfrac{e}{2i}\right]^i \left[d\left[\tfrac{b}{c}\right]^{n+1}\right]^i \leq 2d\left[\tfrac{b}{c}\right]^{n+1} \left[\sum_{i=1}^{d} \left[d\left[\tfrac{b}{c}\right]^{n+1}\right]^{i-1}\right] \\ &= 2d\left[\tfrac{b}{c}\right]^{n+1} \left[\sum_{i=0}^{d-1} \left[d\left[\tfrac{b}{c}\right]^{n+1}\right]^{i}\right] \leq 2d\left[\tfrac{b}{c}\right]^{n+1} \left[\sum_{i=0}^{\infty} \left[d\left[\tfrac{b}{c}\right]^{n+1}\right]^{i}\right]. \end{aligned} \tag{4.14}$$

This, the geometric sum formula, and the fact that for every $\varepsilon \in (0,1]$ it holds that $N_\varepsilon \geq 2bd^{\frac{1}{n}}$ imply that for every $\varepsilon \in (0,1]$, $c \in [N_\varepsilon, \infty)$, $x \in [a,b]^d$ we have

$$|F_x(c)| \leq 2d\left[\tfrac{b}{c}\right]^{n+1} \left[\frac{1}{1 - d\left[\frac{b}{c}\right]^{n+1}}\right] \leq 4d\left[\tfrac{b}{c}\right]^{n+1}. \tag{4.15}$$

Hence, we obtain for every $\varepsilon \in (0,1]$, $x \in [a,b]^d$ that

$$\begin{aligned} \left|\int_{N_\varepsilon}^{\infty} F_x(c)\,\mathrm{d}c\right| &\leq 4db^{n+1} \left|\int_{N_\varepsilon}^{\infty} c^{-(n+1)}\mathrm{d}c\right| = 4db^{n+1}\tfrac{1}{n}(N_\varepsilon)^{-n} \\ &= \tfrac{4}{n}db^{n+1}\left[2e^{2(n+1)}(b+1)^{1+\frac{1}{n}}d^{\frac{1}{n}}\varepsilon^{-\frac{1}{n}}\right]^{-n} \\ &= \tfrac{4}{n}db^{n+1}2^{-n}e^{-(2n^2+2n)}(b+1)^{-(n+1)}d^{-1}\varepsilon \\ &= \tfrac{4}{n}2^{-n}e^{-(2n^2+n)}\left[\tfrac{b}{b+1}\right]^{n+1}\varepsilon \leq \varepsilon. \end{aligned} \tag{4.16}$$

This completes the proof of Lemma 4.2. □

Next we combine the result above with Lemma 4.1 in order to derive the number of terms needed in order to approximate the integral by a sum to within a prescribed error bound ε.

Lemma 4.3. *Let $n \in \mathbb{N}$, $a \in (0,\infty)$, $b \in (a,\infty)$, $(K_i)_{i\in\mathbb{N}} \subseteq [0,\infty)$, let $F_x^d\colon (0,\infty) \to \mathbb{R}$, $x \in [a,b]^d$, $d \in \mathbb{N}$, be the functions which satisfy for every $d \in \mathbb{N}$, $x = (x_1,x_2,\ldots,x_d) \in [a,b]^d$, $c \in (0,\infty)$ that*

$$F_x^d(c) = 1 - \prod_{i=1}^{d} \left[\tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(\frac{K_i+c}{x_i})} e^{-\frac{1}{2}r^2} \mathrm{d}r\right], \tag{4.17}$$

and for every $d \in \mathbb{N}$, $\varepsilon \in (0,1]$ let $N_{d,\varepsilon} \in \mathbb{R}$ be given by

$$N_{d,\varepsilon} = 2e^{2(n+1)}(b+1)^{1+\frac{1}{n}}d^{\frac{1}{n}}\left[\tfrac{\varepsilon}{2}\right]^{-\frac{1}{n}}. \tag{4.18}$$

Then there exist $Q_{d,\varepsilon} \in \mathbb{N}$, $c_{\varepsilon,j}^d \in (0,N_{d,\varepsilon})$, $w_{\varepsilon,j}^d \in [0,\infty)$, $j \in \{1,2,\ldots,Q_{d,\varepsilon}\}$, $d \in \mathbb{N}$, $\varepsilon \in (0,1]$, such *(i) that*

$$\sup_{\varepsilon\in(0,1],d\in\mathbb{N}}\left[\frac{Q_{d,\varepsilon}}{d^{1+\frac{2}{n}}\varepsilon^{-\frac{2}{n}}}\right]<\infty \tag{4.19}$$

and

(ii) that for every $d\in\mathbb{N}$, $\varepsilon\in(0,1]$ it holds $\sum_{j=1}^{Q_{d,\varepsilon}} w_{\varepsilon,j}^d = N_{d,\varepsilon}$ and

$$\sup_{x\in[a,b]^d}\left|\int_0^\infty F_x^d(c)\,\mathrm{d}c-\sum_{j=1}^{Q_{d,\varepsilon}} w_{\varepsilon,j}^d F_x^d(c_{\varepsilon,j}^d)\right|\leq\varepsilon. \tag{4.20}$$

Proof of Lemma 4.3. Note that Lemma 3.5 ensures the existence of $S_m\in\mathbb{R}$, $m\in\mathbb{N}$, such that for every $d,m\in\mathbb{N}$, $x\in[a,b]^d$ it holds

$$\sup_{c\,\in(0,\infty)}\left|(F_x^d)^{(m)}(c)\right|\leq S_m d^m. \tag{4.21}$$

Let $Q_{d,\varepsilon}\in\mathbb{R}$, $d\in\mathbb{N}$, $\varepsilon\in(0,1]$, be given by

$$Q_{d,\varepsilon}=n\left\lceil\left[\tfrac{1}{(2n)!}(N_{d,\varepsilon})^{2n+1}S_{2n}d^{2n}\tfrac{2}{\varepsilon}\right]^{\frac{1}{2n}}\right\rceil. \tag{4.22}$$

Next observe that Lemma 4.1 (with $N\leftrightarrow N_{d,\varepsilon}$ in the notation of Lemma 4.1) establishes the existence of $c_{\varepsilon,j}^d\in(0,N_{d,\varepsilon})$, $w_{\varepsilon,j}^d\in[0,\infty)$, $j\in\{1,2,\dots,Q_{d,\varepsilon}\}$, $d\in\mathbb{N}$, $\varepsilon\in(0,1]$, such that for every $d\in\mathbb{N}$, $\varepsilon\in(0,\infty)$, $x\in[a,b]^d$ we have $\sum_{j=1}^{Q_{d,\varepsilon}} w_{\varepsilon,j}^d=N_{d,\varepsilon}$ and

$$\begin{aligned}\left|\int_0^{N_{d,\varepsilon}}F_x^d(c)\mathrm{d}c-\sum_{j=1}^{Q_{d,\varepsilon}}w_{\varepsilon,j}^dF_x^d(c_{\varepsilon,j}^d)\right| &\leq \tfrac{1}{(2n)!}(N_{d,\varepsilon})^{2n+1}\left[\tfrac{Q_{d,\varepsilon}}{n}\right]^{-2n}S_{2n}d^{2n}\\ &\leq \tfrac{1}{(2n)!}(N_{d,\varepsilon})^{2n+1}\left[\tfrac{1}{(2n)!}(N_{d,\varepsilon})^{2n+1}S_{2n}d^{2n}\tfrac{2}{\varepsilon}\right]^{-1}S_{2n}d^{2n}=\tfrac{\varepsilon}{2}.\end{aligned} \tag{4.23}$$

Moreover, note that Lemma 4.2 (with $N_{d,\frac{\varepsilon}{2}}\leftrightarrow N_{d,\varepsilon}$ in the notation of Lemma 4.2) and (4.23) imply for every $d\in\mathbb{N}$, $\varepsilon\in(0,1]$, $x\in[a,b]^d$ that

$$\begin{aligned}&\left|\int_0^\infty F_x^d(c)\,\mathrm{d}c-\sum_{j=1}^{Q_{d,\varepsilon}}w_{\varepsilon,j}^dF_x^d(c_{\varepsilon,j}^d)\right|\\ &\leq\left|\int_0^{N_{d,\varepsilon}}F_x^d(c)\,\mathrm{d}c-\sum_{j=1}^{Q_{d,\varepsilon}}w_{\varepsilon,j}^dF_x^d(c_{\varepsilon,j}^d)\right|+\left|\int_{N_{d,\varepsilon}}^\infty F_x^d(c)\,\mathrm{d}c\right|\\ &\leq\tfrac{\varepsilon}{2}+\tfrac{\varepsilon}{2}=\varepsilon.\end{aligned} \tag{4.24}$$

Furthermore, we have for every $d\in\mathbb{N}$, $\varepsilon\in(0,1]$ that

$$\begin{aligned}Q_{d,\varepsilon}&\leq n\left(1+\left[\tfrac{1}{(2n)!}(N_{d,\varepsilon})^{2n+1}S_{2n}d^{2n}\tfrac{2}{\varepsilon}\right]^{\frac{1}{2n}}\right)\\ &=n+n\left[\tfrac{2S_{2n}}{(2n)!}\right]^{\frac{1}{2n}}d\varepsilon^{-\frac{1}{2n}}(N_{d,\varepsilon})^{1+\frac{1}{2n}}\\ &\leq n+n\left[\tfrac{2S_{2n}}{(2n)!}\right]^{\frac{1}{2n}}d\varepsilon^{-\frac{1}{2n}}\left[4e^{2(n+1)}(b+1)^{1+\frac{1}{n}}d^{\frac{1}{n}}\varepsilon^{-\frac{1}{n}}\right]^{1+\frac{1}{2n}}\\ &=n+4n\left[\tfrac{8S_{2n}}{(2n)!}\right]^{\frac{1}{2n}}e^{2n+3+\frac{1}{n}}\,[b+1]^{1+\frac{3}{2n}+\frac{1}{2n^2}}\,d^{1+\frac{1}{n}+\frac{1}{2n^2}}\varepsilon^{-\frac{3}{2n}-\frac{1}{2n^2}}\\ &\leq nd^{1+\frac{2}{n}}\varepsilon^{-\frac{2}{n}}+4n\left[\tfrac{8S_{2n}}{(2n)!}\right]^{\frac{1}{2n}}e^{2n+3+\frac{1}{n}}\,[b+1]^{1+\frac{3}{2n}+\frac{1}{2n^2}}\,d^{1+\frac{2}{n}}\varepsilon^{-\frac{2}{n}}.\end{aligned} \tag{4.25}$$

This implies

$$\sup_{\varepsilon\in(0,1],d\in\mathbb{N}}\left[\frac{Q_{d,\varepsilon}}{d^{1+\frac{2}{n}}\varepsilon^{-\frac{2}{n}}}\right] \leq n + 4n\left[\tfrac{8S_{2n}}{(2n)!}\right]^{\frac{1}{2n}} e^{2n+3+\frac{1}{n}}\left[b+1\right]^{1+\frac{3}{2n}+\frac{1}{2n^2}} < \infty. \tag{4.26}$$

The proof of Lemma 4.3 is thus completed. □

5 Basic ReLU DNN Calculus

In order to talk about neural networks we will, up to some minor changes and additions, adopt the notation of P. Petersen and F. Voigtlaender from [36]. This allows us to differentiate between a neural network, defined as a structured set of weights, and its realization, which is a function on $\mathbb{R}^d$. Note that this is almost necessary in order to talk about the complexity of neural networks, since notions like depth, size or architecture do not make sense for general functions on $\mathbb{R}^d$. Even if we know that a given function 'is' a neural network, i.e. can be written a series of affine transformations and componentwise non-linearities, there are, in general, multiple non-trivially different ways to do so.

Each of these structured sets we consider does however define a unique function. This enables us to explicitly and unambiguously construct complex neural networks from simple ones, and subsequently relate the approximation capability of a given network to its complexity. Further note that since the realization of neural network is unique we can still speak of a neural network approximating a given function when its realization does so.

Specifically, a neural network will be given by its architecture, i.e. number of layers L and layer dimensions[1] $N_0, N_1, \ldots, N_L$, as well as the weights determining the affine transformations used to compute each layer from the previous one. Note that our notion of neural networks does not attach the architecture and weights to a fixed activation function, but instead considers the realization of such a neural network with respect to a given activation function. This choice is a purely technical one here, as we always consider networks with ReLU activation function.

Setting 5.1 (Neural networks). *For every $L \in \mathbb{N}$, $N_0, N_1, \ldots, N_L \in \mathbb{N}$ let $\mathcal{N}_L^{N_0,N_1,\ldots,N_L}$ be the set given by*

$$\mathcal{N}_L^{N_0,N_1,\ldots,N_L} = \times_{l=1}^{L}\left(\mathbb{R}^{N_l\times N_{l-1}} \times \mathbb{R}^{N_l}\right), \tag{5.1}$$

let $\mathfrak{N}$ be the set given by

$$\mathfrak{N} = \bigcup_{\substack{L\in\mathbb{N},\\ N_0,N_1,\ldots,N_L\in\mathbb{N}}} \mathcal{N}_L^{N_0,N_1,\ldots,N_L}, \tag{5.2}$$

let $\mathcal{L}, \mathcal{M}, \mathcal{M}_l, \dim_{\mathrm{in}}, \dim_{\mathrm{out}} \colon \mathfrak{N} \to \mathbb{N}$, $l \in \{1, 2, \ldots, L\}$, be the functions which satisfy for every $L \in \mathbb{N}$ and every $N_0, N_1, \ldots, N_L \in \mathbb{N}$, $\Phi = (((A^1_{i,j})_{i,j=1}^{N_1,N_0}, (b^1_i)_{i=1}^{N_1}), \ldots, ((A^L_{i,j})_{i,j=1}^{N_L,N_{L-1}}, (b^L_i)_{i=1}^{N_L})) \in \mathcal{N}_L^{N_0,N_1,\ldots,N_L}$, $l \in \{1, 2, \ldots, L\}$ $\mathcal{L}(\Phi) = L$, $\dim_{\mathrm{in}}(\Phi) = N_0$, $\dim_{\mathrm{out}}(\Phi) = N_L$,

$$\mathcal{M}_l(\Phi) = \sum_{i=1}^{N_l}\left[\mathbb{1}_{\mathbb{R}\setminus\{0\}}(b^l_i) + \sum_{j=1}^{N_{l-1}}\mathbb{1}_{\mathbb{R}\setminus\{0\}}(A^l_{i,j})\right], \tag{5.3}$$

and

$$\mathcal{M}(\Phi) = \sum_{l=1}^{L}\mathcal{M}_l(\Phi). \tag{5.4}$$

For every $\varrho \in C(\mathbb{R}, \mathbb{R})$ let $\varrho^* \colon \cup_{d\in\mathbb{N}} \mathbb{R}^d \to \cup_{d\in\mathbb{N}}\mathbb{R}^d$ be the function which satisfies for every $d \in \mathbb{N}$, $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ that $\varrho^*(x) = (\varrho(x_1), \varrho(x_2), \ldots, \varrho(x_d))$, and for every $\varrho \in \mathcal{C}(\mathbb{R}, \mathbb{R})$ denote by

[1] Often phrased as input dimension N_0 and output dimension N_L with N_l, $l \in \{1, 2, \ldots, L-1\}$ many neurons in the l'th layer.

$R_\varrho \colon \mathfrak{N} \to \cup_{a,b\in\mathbb{N}} C(\mathbb{R}^a, \mathbb{R}^b)$ the function which satisfies for every $L \in \mathbb{N}$, $N_0, N_1, \ldots, N_L \in \mathbb{N}$, $x_0 \in \mathbb{R}^{N_0}$, and $\Phi = ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)) \in \mathcal{N}_L^{N_0, N_1, \ldots, N_L}$, with $x_1 \in \mathbb{R}^{N_1}, \ldots, x_{L-1} \in \mathbb{R}^{N_{L-1}}$ given by

$$x_l = \varrho^*(A_l x_{l-1} + b_l) \ , \qquad l = 1, ..., L-1 \ , \tag{5.5}$$

that

$$[R_\varrho(\Phi)]\,(x_0) = A_L x_{L-1} + b_L \ . \tag{5.6}$$

The quantity $\mathcal{M}(\Phi)$ simply denotes the number of non-zero entries of the network Φ, which together with its depth $\mathcal{L}(\Phi)$ will be how we measure the 'size' of a given neural network Φ. One could instead consider the number of all weights, i.e. including zeroes, of a neural network. Note, however, that for any non-degenerate neural network Φ the total number of weights is bounded from above by $\mathcal{M}(\Phi)^2 + \mathcal{M}(\Phi)$. Here, the terminology "degenerate" refers to a neural network which has neurons that can be removed without changing the realization of the NN. This implies for any neural network there also exists a non-degenerate one of smaller or equal size, which has the exact same realization. Since our primary goal is to approximate d-variate functions by networks the size of which only depends polynomially on the dimension, the above means that the qualitatively same results hold regardless of which notion of 'size' is used.

We start by introducing two basic tools for constructing new neural networks from known ones and, in Lemma 5.3 and Lemma 5.4, consider how the properties of a derived network depend on its parts. Note that techniques like these have already been used in [36] and [39].

The first tool will be the 'composition' of neural networks in (5.7), which takes two networks and provides a new network whose realization is the composition of the realizations of the two constituent functions.

The second tool will be the 'parallelization' of neural networks in (5.12), which will be useful when considering linear combinations or tensor products of functions which we can already approximate. While parallelization of same-depth networks (5.10) works with arbitrary activation functions, we use for the general case that any ReLU network can easily be extended (5.11) to an arbitrary depth without changing its realization.

Setting 5.2. *Assume Setting 5.1, for every $L_1, L_2 \in \mathbb{N}$, $\Phi^i = \big((A_1^i, b_1^i), (A_2^i, b_2^i), \ldots, (A_{L_i}^i, b_{L_i}^i)\big) \in \mathfrak{N}$, $i \in \{1, 2\}$, with $\dim_{\text{in}}(\Phi^1) = \dim_{\text{out}}(\Phi^2)$ let $\Phi^1 \odot \Phi^2 \in \mathfrak{N}$ be the neural network given by*

$$\Phi^1 \odot \Phi^2 = \left((A_1^2, b_1^2), \ldots, (A_{L_2-1}^2, b_{L_2-1}^2), \left(\begin{pmatrix} A_{L_2}^2 \\ -A_{L_2}^2 \end{pmatrix}, \begin{pmatrix} b_{L_2}^2 \\ -b_{L_2}^2 \end{pmatrix}\right), \left(\begin{pmatrix} A_1^1 & -A_1^1 \end{pmatrix}, b_1^1\right), (A_2^1, b_2^1), \ldots, (A_{L_1}^1, b_{L_1}^1)\right), \tag{5.7}$$

for every $d \in \mathbb{N}$, $L \in \mathbb{N} \cap [2, \infty)$ let $\Phi_{d,L}^{\text{Id}} \in \mathfrak{N}$ be the neural network given by

$$\Phi_{d,L}^{\text{Id}} = \left(\left(\begin{pmatrix} \text{Id}_{\mathbb{R}^d} \\ -\text{Id}_{\mathbb{R}^d} \end{pmatrix}, 0\right), \underbrace{(\text{Id}_{\mathbb{R}^{2d}}, 0), \ldots, (\text{Id}_{\mathbb{R}^{2d}}, 0)}_{\textit{L-2 times}}, \left(\begin{pmatrix} \text{Id}_{\mathbb{R}^d} & -\text{Id}_{\mathbb{R}^d} \end{pmatrix}, 0\right)\right), \tag{5.8}$$

for every $d \in \mathbb{N}$ let $\Phi_{d,1}^{\text{Id}} \in \mathfrak{N}$ be the neural network given by

$$\Phi_{d,1}^{\text{Id}} = ((\text{Id}_{\mathbb{R}^d}, 0)), \tag{5.9}$$

for every $n, L \in \mathbb{N}$, $\Phi^j = ((A_1^j, b_1^j), (A_2^j, b_2^j), \ldots, (A_L^j, b_L^j)) \in \mathfrak{N}$, $j \in \{1, 2, \ldots, n\}$, let $\mathcal{P}_s(\Phi^1, \Phi^2, \ldots, \Phi^n) \in \mathfrak{N}$ be the neural network which satisfies

$$\mathcal{P}_s(\Phi^1, \Phi^2, \ldots, \Phi^n) = \left(\left(\begin{pmatrix} A_1^1 & & & \\ & A_1^2 & & \\ & & \ddots & \\ & & & A_1^n \end{pmatrix}, \begin{pmatrix} b_1^1 \\ b_1^2 \\ \vdots \\ b_1^n \end{pmatrix}\right), \ldots, \left(\begin{pmatrix} A_L^1 & & & \\ & A_L^2 & & \\ & & \ddots & \\ & & & A_L^n \end{pmatrix}, \begin{pmatrix} b_L^1 \\ b_L^2 \\ \vdots \\ b_L^n \end{pmatrix}\right)\right), \tag{5.10}$$

for every $L, d \in \mathbb{N}$, $\Phi \in \mathfrak{N}$ with $\mathcal{L}(\Phi) \leq L$, $\dim_{\text{out}}(\Phi) = d$, let $\mathcal{E}_L(\Phi) \in \mathfrak{N}$ be the neural network given by

$$\mathcal{E}_L(\Phi) = \begin{cases} \Phi_{d, L-\mathcal{L}(\Phi)}^{\text{Id}} \odot \Phi & : \mathcal{L}(\Phi) < L \\ \Phi & : \mathcal{L}(\Phi) = L \end{cases}, \tag{5.11}$$

and for every $n, L \in \mathbb{N}$, $\Phi^j \in \mathfrak{N}$, $j \in \{1, 2, \dots, n\}$ with $\max_{j\in\{1,2,\dots,n\}} \mathcal{L}(\Phi^j) = L$, let $\mathcal{P}(\Phi^1, \Phi^2, \dots, \Phi^n) \in \mathfrak{N}$ denote the neural network given by

$$\mathcal{P}(\Phi^1, \Phi^2, \dots, \Phi^n) = \mathcal{P}_s(\mathcal{E}_L(\Phi^1), \mathcal{E}_L(\Phi^2), \dots, \mathcal{E}_L(\Phi^n)). \tag{5.12}$$

Lemma 5.3. *Assume Setting 5.2, let $\Phi^1, \Phi^2 \in \mathfrak{N}$, and let $\varrho\colon \mathbb{R} \to \mathbb{R}$ be the function which satisfies for every $t \in \mathbb{R}$ that $\varrho(t) = \max\{0, t\}$. Then*

(i) for every $x \in \mathbb{R}^{\dim_{\text{in}}(\Phi^2)}$ it holds

$$[R_\varrho(\Phi^1 \odot \Phi^2)](x) = ([R_\varrho(\Phi^1)] \circ [R_\varrho(\Phi^2)])(x) = [R_\varrho(\Phi^1)]([R_\varrho(\Phi^2)](x)), \tag{5.13}$$

(ii) $\mathcal{L}(\Phi^1 \odot \Phi^2) = \mathcal{L}(\Phi^1) + \mathcal{L}(\Phi^2)$,

(iii) $\mathcal{M}(\Phi^1 \odot \Phi^2) \le \mathcal{M}(\Phi^1) + \mathcal{M}(\Phi^2) + \mathcal{M}_1(\Phi^1) + \mathcal{M}_{\mathcal{L}(\Phi^2)}(\Phi^2) \le 2(\mathcal{M}(\Phi^1) + \mathcal{M}(\Phi^2))$,

(iv) $\mathcal{M}_1(\Phi^1 \odot \Phi^2) = \mathcal{M}_1(\Phi^2)$,

(v) $\mathcal{M}_{\mathcal{L}(\Phi^1\odot\Phi^2)}(\Phi^1 \odot \Phi^2) = \mathcal{M}_{\mathcal{L}(\Phi^1)}(\Phi^1)$,

(vi) $\dim_{\text{in}}(\Phi^1 \odot \Phi^2) = \dim_{\text{in}}(\Phi^2)$,

(vii) $\dim_{\text{out}}(\Phi^1 \odot \Phi^2) = \dim_{\text{out}}(\Phi^1)$,

(viii) for every $d, L \in \mathbb{N}$, $x \in \mathbb{R}^d$ it holds that $[R_\varrho(\Phi^{\text{Id}}_{d,L})](x) = x$, and

(ix) for every $L \in \mathbb{N}$, $\Phi \in \mathfrak{N}$ with $\mathcal{L}(\Phi) \le L$, $x \in \mathbb{R}^{\dim_{\text{in}}(\Phi)}$ it holds that $[R_\varrho(\mathcal{E}_L(\Phi))](x) = [R_\varrho(\Phi)](x)$.

Proof of Lemma 5.3. For every $i \in \{1, 2\}$ let $L_i \in \mathbb{N}$, $N^i_1, N^i_2, \dots, N^i_{L_i}$, $(A^i_l, b^i_l) \in \mathbb{R}^{N^i_l \times N^i_{l-1}} \times \mathbb{R}^{N^i_l}$, $l \in \{1, 2, \dots, L_i\}$ such that $\Phi^i = ((A^i_1, b^i_1), \dots, (A^i_{L_i}, b^i_{L_i}))$. Furthermore, let $(A_l, b_l) \in \mathbb{R}^{N_l \times N_{l-1}} \times \mathbb{R}^{N_l}$, $l \in \{1, 2, \dots, L_1 + L_2\}$, be the matrix-vector tuples which satisfy $\Phi_1 \odot \Phi_2 = ((A_1, b_1), \dots, (A_{L_1+L_2}, b_{L_1+L_2}))$ and let $r_l\colon \mathbb{R}^{N_0} \to \mathbb{R}^{N_l}$, $l \in \{1, 2, \dots, L_1 + L_2\}$, be the functions which satisfy for every $x \in \mathbb{R}^{N_0}$ that

$$r_l(x) = \begin{cases} \varrho^*(A_1 x + b_1) & : l = 1 \\ \varrho^*(A_l r_{l-1}(x) + b_l) & : 1 < l < L_1 + L_2 \\ A_l r_{l-1}(x) + b_l & : l = L_1 + L_2 \end{cases}. \tag{5.14}$$

Observe that for every $l \in \{1, 2, \dots, L_2 - 1\}$ holds $(A_l, b_l) = (A^2_l, b^2_l)$. This implies that for every $x \in \mathbb{R}^{N_0}$ holds

$$A^2_{L_2} r_{L_2-1}(x) + b^2_{L_2} = [R_\varrho(\Phi_2)](x). \tag{5.15}$$

Combining this with (5.7) implies for every $x \in \mathbb{R}^{N_0}$ that

$$\begin{aligned} r_{L_2}(x) &= \varrho^*(A_{L_2} r_{L_2-1}(x) + b_{L_2}) = \varrho^*\left(\begin{pmatrix} A^2_{L_2} \\ -A^2_{L_2} \end{pmatrix} r_{L_2-1}(x) + \begin{pmatrix} b^2_{L_2} \\ -b^2_{L_2} \end{pmatrix}\right) \\ &= \varrho^*\left(\begin{pmatrix} A^2_{L_2} r_{l-1}(x) + b^2_{L_2} \\ -A^2_{L_2} r_{l-1}(x) - b^2_{L_2} \end{pmatrix}\right) = \begin{pmatrix} \varrho^*([R_\varrho(\Phi^2)](x)) \\ \varrho^*(-[R_\varrho(\Phi^2)](x)) \end{pmatrix} \end{aligned} \tag{5.16}$$

In addition, for every $d \in \mathbb{N}$, $y = (y_1, y_2, \dots, y_d) \in \mathbb{R}^d$ holds

$$\varrho^*(y) - \varrho^*(-y) = (\varrho(y_1) - \varrho(-y_1), \varrho(y_2) - \varrho(-y_2), \dots, \varrho(y_d) - \varrho(-y_d)) = y. \tag{5.17}$$

This, (5.7), and (5.16) ensure that for every $x \in \mathbb{R}^{N_0}$ holds

$$\begin{aligned} r_{L_2+1}(x) &= A_{L_2+1}\begin{pmatrix} \varrho^*([R_\varrho(\Phi^2)](x)) \\ \varrho^*(-[R_\varrho(\Phi^2)](x)) \end{pmatrix} + b_{L_2+1} \\ &= A^1_1 \varrho^*([R_\varrho(\Phi^2)](x)) - A^1_1 \varrho^*(-[R_\varrho(\Phi^2)](x)) + b_{L_2+1} \\ &= A^1_1 [R_\varrho(\Phi^2)](x) + b^1_1. \end{aligned} \tag{5.18}$$

Combining this with (5.14) establishes (i). Moreover, (ii)-(vii) follow directly from (5.7). Furthermore, (5.8), (5.9), and (5.17) imply (viii). Finally, (ix) follows from (5.11) and (viii). This completes the proof of Lemma 5.3. □

Lemma 5.4. *Assume Setting 5.2, let $\varrho\colon \mathbb{R}\to\mathbb{R}$ be the function which satisfies for every $t\in\mathbb{R}$ that $\varrho(t)=\max\{0,t\}$, let $n\in\mathbb{N}$, let $\varphi^j\in\mathfrak{N}$, $j\in\{1,2,\dots,n\}$, let $d_j\in\mathbb{N}$, $j\in\{1,2,\dots,n\}$, be given by $d_j=\dim_{\mathrm{in}}(\varphi^j)$, let $D\in\mathbb{N}$ be given by $D=\sum_{j=1}^n d_j$, and let $\Phi\in\mathfrak{N}$ be given by $\Phi=\mathcal{P}(\varphi^1,\varphi^2,\dots,\varphi^n)$. Then*

(i) for every $x\in\mathbb{R}^D$ it holds

$$[R_\varrho(\Phi)](x)=\big([R_\varrho(\varphi^1)](x_1,\dots,x_{d_1}),[R_\varrho(\varphi^2)](x_{d_1+1},\dots,x_{d_1+d_2}),\dots,[R_\varrho(\varphi^n)](x_{D-d_n+1},\dots,x_D)\big)\,, \tag{5.19}$$

(ii) $\mathcal{L}(\Phi)=\max_{j\in\{1,2,\dots,n\}}\mathcal{L}(\varphi^j)$,

(iii) $\mathcal{M}(\Phi)\le 2\left(\sum_{j=1}^n\mathcal{M}(\varphi^j)\right)+4\left(\sum_{j=1}^n\dim_{\mathrm{out}}(\varphi^j)\right)\max_{j\in\{1,2,\dots,n\}}\mathcal{L}(\varphi^j)$,

(iv) $\mathcal{M}(\Phi)=\sum_{j=1}^n\mathcal{M}(\varphi^j)$ *provided for every $j,j'\in\{1,2,\dots,n\}$ holds $\mathcal{L}(\varphi^j)=\mathcal{L}(\varphi^{j'})$,*

(v) $\mathcal{M}_{\mathcal{L}(\Phi)}(\Phi)\le\sum_{j=1}^n\max\{2\dim_{\mathrm{out}}(\varphi^j),\mathcal{M}_{\mathcal{L}(\varphi^j)}(\varphi^j)\}$,

(vi) $\mathcal{M}_1(\Phi)=\sum_{j=1}^n\mathcal{M}_1(\varphi^j)$,

(vii) $\dim_{\mathrm{in}}(\Phi)=\sum_{j=1}^n\dim_{\mathrm{in}}(\varphi^j)$, *and*

(viii) $\dim_{\mathrm{out}}(\Phi)=\sum_{j=1}^n\dim_{\mathrm{out}}(\varphi^j)$.

Proof of Lemma 5.4. Observe that Lemma 5.3 implies that for every $j\in\{1,2,\dots,n\}$ holds

$$R_\varrho(\mathcal{E}_{\mathcal{L}(\Phi)}(\varphi^j))=R_\varrho(\varphi^j). \tag{5.20}$$

Combining this with (5.10) and (5.12) establishes (i). Furthermore, note that that (ii), (vi), (vii), and (viii) follow directly from (5.10) and (5.12). Moreover, (5.10) demonstrates that for every $m\in\mathbb{N}$, $\psi_i\in\mathfrak{N}$, $i\in\{1,2,\dots,m\}$, with $\forall i,i'\in\{1,2,\dots,m\}\colon\mathcal{L}(\psi^i)=\mathcal{L}(\psi^{i'})$ holds

$$\mathcal{M}(\mathcal{P}_s(\psi^1,\psi^2,\dots,\psi^m))=\sum_{i=1}^m\mathcal{M}(\psi^i). \tag{5.21}$$

This establishes (iv). Next, observe that Lemma 5.3, (5.11), and the fact that for every $d\in$, $L\in\mathbb{N}$ holds $\mathcal{M}(\Phi^{\mathrm{Id}}_{d,L})\le 2dL$ imply that for every $j\in\{1,2,\dots,n\}$ we have

$$\begin{aligned}\mathcal{M}(\mathcal{E}_{\mathcal{L}(\Phi)}(\varphi^j))&\le 2\mathcal{M}(\Phi^{\mathrm{Id}}_{\dim_{\mathrm{out}}(\varphi^j),\mathcal{L}(\Phi)-\mathcal{L}(\varphi^j)})+2\mathcal{M}(\varphi^j)\\&\le 4\dim_{\mathrm{out}}(\varphi^j)\mathcal{L}(\Phi)+2\mathcal{M}(\varphi^j).\end{aligned} \tag{5.22}$$

Combining this with (5.21) establishes (iii). In addition, note that (5.8), (5.9), and (5.11) ensure for every $j\in\{1,2,\dots,n\}$ that

$$\mathcal{M}_{\mathcal{L}(\Phi)}(\mathcal{E}_{\mathcal{L}(\Phi)}(\varphi^j))\le\max\{2\dim_{\mathrm{out}}(\varphi^j),\mathcal{M}_{\mathcal{L}(\varphi^j)}(\varphi^j)\}. \tag{5.23}$$

Combining this with (5.10) establishes (v). The proof of Lemma 5.4 is thus completed. □

6 Basic Expression Rate Results

Here we begin by establishing an expression rate result for a very simple function, namely $x\mapsto x^2$ on $[0,1]$. Our approach is based on the observation by M. Telgarsky [42], that neural networks with ReLU activation function can efficiently compute high-frequent sawtooth functions, and the idea of D. Yarotsky in [44] to use this in order to approximate the function $x\mapsto x^2$ by networks computing its linear interpolations. This can then be used to derive networks capable of efficiently approximating $(x,y)\mapsto xy$, which leads to tensor products as well as polynomials and subsequently smooth function. Note that [44] uses a slightly different notion of neural networks, where connections between non-adjacent layers are permitted. This does, however, only require a technical modification of the proof, which does not significantly change the result. Nonetheless, the respective proofs are provided in the appendix for completeness.

Lemma 6.1. *Assume Setting 5.1 and let $\varrho\colon \mathbb{R} \to \mathbb{R}$ be the ReLU activation function given by $\varrho(t) = \max\{0, t\}$. Then there exist neural networks $(\sigma_\varepsilon)_{\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ such that for every $\varepsilon \in (0,\infty)$*

(i) $\mathcal{L}(\sigma_\varepsilon) \leq \begin{cases} \frac{1}{2}\,|\log_2(\varepsilon)| + 1 & : \varepsilon < 1 \\ 1 & : \varepsilon \geq 1 \end{cases}$,

(ii) $\mathcal{M}(\sigma_\varepsilon) \leq \begin{cases} 15(\frac{1}{2}\,|\log_2(\varepsilon)| + 1) & : \varepsilon < 1 \\ 0 & : \varepsilon \geq 1 \end{cases}$,

(iii) $\sup_{t\in[0,1]} \left|t^2 - [R_\varrho(\sigma_\varepsilon)](t)\right| \leq \varepsilon$,

(iv) $[R_\varrho(\sigma_\varepsilon)](0) = 0$.

We can now derive the following result on approximate multiplication by neural networks, by observing that $xy = 2B^2(|(x+y)/2B|^2 - |x/2B|^2 - |y/2B|^2)$ for every $B \in (0,\infty)$, $x, y \in \mathbb{R}$.

Lemma 6.2. *Assume Setting 5.1, let $B \in (0,\infty)$, and let $\varrho\colon \mathbb{R} \to \mathbb{R}$ be the ReLU activation function given by $\varrho(t) = \max\{0, t\}$. Then there exist neural networks $(\mu_\varepsilon)_{\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ which satisfy for every $\varepsilon \in (0,\infty)$ that*

(i) $\mathcal{L}(\mu_\varepsilon) \leq \begin{cases} \frac{1}{2}\log_2(\frac{1}{\varepsilon}) + \log_2(B) + 6 & : \varepsilon < B^2 \\ 1 & : \varepsilon \geq B^2 \end{cases}$,

(ii) $\mathcal{M}(\mu_\varepsilon) \leq \begin{cases} 45\log_2(\frac{1}{\varepsilon}) + 90\log_2(B) + 259 & : \varepsilon < B^2 \\ 0 & : \varepsilon \geq B^2 \end{cases}$,

(iii) $\sup_{(x,y)\in[-B,B]^2} |xy - [R_\varrho(\mu_\varepsilon)](x,y)| \leq \varepsilon$,

(iv) $\mathcal{M}_1(\mu_\varepsilon) = 8$, $\mathcal{M}_{\mathcal{L}(\mu_\varepsilon)}(\mu_\varepsilon) = 3$, *and*

(v) for every $x \in \mathbb{R}$ it holds that $R_\varrho[\mu_\varepsilon](0,x) = R_\varrho[\mu_\varepsilon](x,0) = 0$.

Next we extend this result to products of any number of factors by hierarchical, pairwise multiplication.

Theorem 6.3. *Assume Setting 5.1, let $\varrho\colon \mathbb{R} \to \mathbb{R}$ be the ReLU activation function given by $\varrho(t) = \max\{0, t\}$, let $m \in \mathbb{N} \cap [2,\infty)$, and let $B \in [1,\infty)$. Then there exists a constant $C \in \mathbb{R}$ (which is independent of m, B) and neural networks $(\Pi_\varepsilon)_{\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ which satisfy*

(i) $\mathcal{L}(\Pi_\varepsilon) \leq C\ln(m)\,(|\ln(\varepsilon)| + m\ln(B) + \ln(m))$,

(ii) $\mathcal{M}(\Pi_\varepsilon) \leq Cm\,(|\ln(\varepsilon)| + m\ln(B) + \ln(m))$,

(iii) $\displaystyle\sup_{x\in[-B,B]^m} \left| \left[\prod_{j=1}^m x_j\right] - [R_\varrho(\Pi_\varepsilon)](x) \right| \leq \varepsilon$, *and*

(iv) $R_\varrho\,[\Pi_\varepsilon]\,(x_1, x_2, \ldots, x_m) = 0$, *if there exists $i \in \{1, 2, \ldots, m\}$ with $x_i = 0$.*

Proof of Theorem 6.3. Throughout this proof assume Setting 5.2, let $l = \lceil \log_2 m \rceil$, and let $\theta \in \mathcal{N}_1^{1,1}$ be the neural network given by $\theta = (0,0)$, let $(A,b) \in \mathbb{R}^{l\times m} \times \mathbb{R}^l$ be the matrix-vector tuple given by

$$A_{i,j} = \begin{cases} 1 & : i = j, j \leq m \\ 0 & : \text{else} \end{cases} \quad \text{and} \quad b_i = \begin{cases} 0 & : i \leq m \\ 1 & : i > m \end{cases}. \tag{6.1}$$

Let further $\omega \in \mathcal{N}_2^{m,2^l}$ be the neural network given by $\omega = ((A,b))$. Note that Lemma 6.2 (with B^m as B in the notation of Lemma 6.2) ensures that there exist neural networks $(\mu_\eta)_{\eta\in(0,\infty)} \subseteq \mathfrak{N}$ such that for every $\eta \in (0, [B^m]^2)$ it holds

(A) $\mathcal{L}(\mu_\eta) \leq \frac{1}{2}\log_2(\frac{1}{\eta}) + \log_2(B^m) + 6$,

(B) $\mathcal{M}(\mu_\eta) \leq 45\log_2(\frac{1}{\eta}) + 90\log_2(B^m) + 259$,

(C) $\sup_{x,y\in[-B^m,B^m]} |xy - [R_\varrho(\mu_\eta)](x,y)| \leq \eta$,

(D) $\mathcal{M}_1(\mu_\eta) = 8,\ \mathcal{M}_{\mathcal{L}(\mu_\eta)}(\mu_\eta) = 3$, and

(E) for every $x \in \mathbb{R}$ it holds that $R_\varrho[\mu_\eta](0,x) = R_\varrho[\mu_\eta](x,0) = 0$.

Let $(\nu_\varepsilon)_{\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ be the neural networks which satisfy for every $\varepsilon \in (0,\infty)$

$$\nu_\varepsilon = \mu_{m^{-2}B^{-2m}\varepsilon}. \tag{6.2}$$

Observe that (A) implies that for every $\varepsilon \in (0,B^m) \subseteq (0,m^2B^{4m})$ it holds

$$\begin{aligned}\mathcal{L}(\nu_\varepsilon) &\leq \tfrac{1}{2}\log_2(\tfrac{1}{m^{-2}B^{-2m}\varepsilon}) + \log_2(B^m) + 6\\ &= \tfrac{1}{2}(\log_2(\tfrac{1}{\varepsilon}) + 2\log_2(m) + 2m\log_2(B)) + m\log_2(B) + 6\\ &= \tfrac{1}{2}\log_2(\tfrac{1}{\varepsilon}) + 2m\log_2(B) + \log_2(m) + 6.\end{aligned} \tag{6.3}$$

In addition, note that (B) implies that for every $\varepsilon \in (0,B^m) \subseteq (0,m^2B^{4m})$

$$\begin{aligned}\mathcal{M}(\nu_\varepsilon) &\leq 45\log_2(\tfrac{1}{m^{-2}B^{-2m}\varepsilon}) + 90\log_2(B^m) + 259\\ &= 45\log_2(\tfrac{1}{\varepsilon}) + 180m\log_2(B) + 90\log_2(m) + 259.\end{aligned} \tag{6.4}$$

Furthermore, (C) implies that for every $\varepsilon \in (0,B^m) \subseteq (0,m^2B^{4m})$ holds

$$\sup_{x,y\in[-B^m,B^m]} |xy - [R_\varrho(\nu_\eta)](x,y)| \leq m^{-2}B^{-2m}\varepsilon. \tag{6.5}$$

Let $\pi_{k,\varepsilon} \in \mathfrak{N}$, $\varepsilon \in (0,\infty)$, $k \in \mathbb{N}$, be the neural networks which satisfy for every $\varepsilon \in (0,\infty)$, $k \in \mathbb{N}$

$$\pi_{k,\varepsilon} = \begin{cases}\nu_\varepsilon & : k = 1\\ \nu_\varepsilon \odot \mathcal{P}(\pi_{k-1,\varepsilon}, \pi_{k-1,\varepsilon}) & : k > 1\end{cases} \tag{6.6}$$

and let $(\Pi_\varepsilon)_{\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ be neural networks given by

$$\Pi_\varepsilon = \begin{cases}\pi_{l,\varepsilon} \odot \omega & : \varepsilon < B^m\\ \theta & : \varepsilon \geq B^m\end{cases}. \tag{6.7}$$

Note that for every $\varepsilon \in (B^m,\infty)$ it holds

$$\begin{aligned}\sup_{x\in[-B,B]^m}\left|\left[\prod_{j=1}^m x_j\right] - [R_\varrho(\Pi_\varepsilon)](x)\right| &= \sup_{x\in[-B,B]^m}\left|\left[\prod_{j=1}^m x_j\right] - [R_\varrho(\theta)](x)\right|\\ &= \sup_{x\in[-B,B]^m}\left|\left[\prod_{j=1}^m x_j\right] - 0\right| = B^m \leq \varepsilon.\end{aligned} \tag{6.8}$$

We claim that for every $k \in \{1,2,\dots,l\}$, $\varepsilon \in (0,B^m)$ it holds

(a) that

$$\sup_{x\in[-B,B]^{(2^k)}}\left|\left[\prod_{j=1}^{2^k} x_j\right] - [R_\varrho(\pi_{k,\varepsilon})](x)\right| \leq 4^{k-1}m^{-2}B^{(2^k-2m)}\varepsilon, \tag{6.9}$$

(b) that $\mathcal{L}(\pi_{k,\varepsilon}) \leq k\mathcal{L}(\nu_\varepsilon)$, and

(c) that $\mathcal{M}(\pi_{k,\varepsilon}) \leq (2^k - 1)\mathcal{M}(\nu_\varepsilon) + (2^{k-1} - 1)20$.

We prove (a), (b), and (c) by induction on $k \in \{1, 2, \dots, l\}$. Observe that (6.5) and the fact that $B \in [1, \infty)$ establishes (a) for $k = 1$. Moreover, note that (6.6) establishes (b) and (c) in the base case $k = 1$.

For the induction step $\{1, 2, \dots, l-1\} \ni k \to k+1 \in \{2, 3, \dots, l\}$ note that Lemma 5.3, Lemma 5.4, (6.5) and (6.6) imply that for every $k \in \{1, 2, \dots, l-1\}$, $\varepsilon \in (0, B^m)$

$$\begin{aligned}
&\sup_{x\in[-B,B]^{(2^{k+1})}} \left| \left[\prod_{j=1}^{2^{k+1}} x_j\right] - [R_\varrho(\pi_{k+1,\varepsilon})](x) \right| \\
&= \sup_{x,x'\in[-B,B]^{(2^k)}} \left| \left[\prod_{j=1}^{2^k} x_j\right]\left[\prod_{j=1}^{2^k} x'_j\right] - [R_\varrho(\pi_{k+1,\varepsilon})]\left((x,x')\right) \right| \\
&= \sup_{x,x'\in[-B,B]^{(2^k)}} \left| \left[\prod_{j=1}^{2^k} x_j\right]\left[\prod_{j=1}^{2^k} x'_j\right] - [R_\varrho(\nu_\varepsilon)]\left([R_\varrho(\pi_{k,\varepsilon})](x), [R_\varrho(\pi_{k,\varepsilon})](x')\right) \right| \\
&\leq \sup_{x,x'\in[-B,B]^{(2^k)}} \left| \left[\prod_{j=1}^{2^k} x_j\right]\left[\prod_{j=1}^{2^k} x'_j\right] - \left([R_\varrho(\pi_{k,\varepsilon})](x)\right)\left([R_\varrho(\pi_{k,\varepsilon})](x')\right) \right| \\
&\quad + \sup_{x,x'\in[-B,B]^{(2^k)}} \left| \left([R_\varrho(\pi_{k,\varepsilon})](x)\right)\left([R_\varrho(\pi_{k,\varepsilon})](x')\right) - [R_\varrho(\nu_\varepsilon)]\left([R_\varrho(\pi_{k,\varepsilon})](x), [R_\varrho(\pi_{k,\varepsilon})](x')\right) \right| \\
&\leq \sup_{x,x'\in[-B,B]^{(2^k)}} \left| \left[\prod_{j=1}^{2^k} x_j\right]\left[\prod_{j=1}^{2^k} x'_j\right] - \left([R_\varrho(\pi_{k,\varepsilon})](x)\right)\left([R_\varrho(\pi_{k,\varepsilon})](x')\right) \right| + m^{-2}B^{-2m}\varepsilon.
\end{aligned} \tag{6.10}$$

Next, for every $c, \delta \in (0, \infty)$, $y, z \in [-c, c]$, $\tilde{y}, \tilde{z} \in \mathbb{R}$ with $|y - \tilde{y}|, |z - \tilde{z}| \leq \delta$ it holds

$$|yz - \tilde{y}\tilde{z}| \leq 2(|y| + |z|)\delta + \delta^2 \leq 2c\delta + \delta^2. \tag{6.11}$$

Moreover, for every $k \in \{1, 2, \dots, l\}$

$$4^{k-1} \leq 4^{l-1} = 4^{\lceil \log_2 m \rceil - 1} \leq 4^{\log_2 m} = m^2. \tag{6.12}$$

The fact that $B \in [1, \infty)$ therefore ensures that for every $k \in \{1, 2, \dots, l-1\}$, $\varepsilon \in (0, B^m)$

$$\left[4^{k-1}m^{-2}B^{(2^k-2m)}\varepsilon\right]^2 = \left[4^{k-1}m^{-2}B^{(2^{k+1}-2m)}\varepsilon\right]\left[4^{k-1}m^{-2}B^{-2m}\varepsilon\right] \leq \left[4^{k-1}m^{-2}B^{(2^{k+1}-2m)}\varepsilon\right]. \tag{6.13}$$

This and (6.11) imply that for every $k \in \{1, 2, \dots, l-1\}$, $\varepsilon \in (0, B^m)$, $x, x' \in [-B, B]^{(2^k)}$

$$\begin{aligned}
&\left| \left[\prod_{j=1}^{2^k} x_j\right]\left[\prod_{j=1}^{2^k} x'_j\right] - \left([R_\varrho(\pi_{k,\varepsilon})](x)\right)\left([R_\varrho(\pi_{k,\varepsilon})](x')\right) \right| \\
&\leq 2B^{(2^k)}4^{k-1}m^{-2}B^{(2^k-2m)}\varepsilon + \left[4^{k-1}m^{-2}B^{(2^k-2m)}\varepsilon\right]^2 \\
&\leq 3\left[4^{k-1}m^{-2}B^{(2^{k+1}-2m)}\varepsilon\right].
\end{aligned} \tag{6.14}$$

Combining this, (6.10), and the fact that $B \in [1, \infty)$ demonstrates that for every $k \in \{1, 2, \dots, l-1\}$, $\varepsilon \in (0, B^m)$

$$\begin{aligned}
&\sup_{x\in[-B,B]^{(2^{k+1})}} \left| \left[\prod_{j=1}^{2^{k+1}} x_j\right] - [R_\varrho(\pi_{k+1,\varepsilon})](x) \right| \\
&\leq 3\left[4^{k-1}m^{-2}B^{(2^{k+1}-2m)}\varepsilon\right] + m^{-2}B^{-2m}\varepsilon \\
&\leq 4^k m^{-2}B^{(2^{k+1}-2m)}\varepsilon.
\end{aligned} \tag{6.15}$$

This establishes the claim (a). Moreover, Lemma 5.3 and Lemma 5.4 imply for every $k \in \{1, 2, \ldots, l-1\}$, $\varepsilon \in (0, B^m)$ with $\mathcal{L}(\pi_{k,\varepsilon}) \leq k\mathcal{L}(\nu_\varepsilon)$ holds

$$\begin{aligned} \mathcal{L}(\pi_{k+1,\varepsilon}) &= \mathcal{L}(\nu_\varepsilon) + \max\{\mathcal{L}(\pi_{k,\varepsilon}), \mathcal{L}(\pi_{k,\varepsilon})\} \\ &\leq \mathcal{L}(\nu_\varepsilon) + k\mathcal{L}(\nu_\varepsilon) = (k+1)\mathcal{L}(\nu_\varepsilon). \end{aligned} \tag{6.16}$$

This establishes the claim (b). Furthermore, Lemma 5.3, Lemma 5.4, (B), and (D) imply for every $k \in \{1, 2, \ldots, l-1\}$, $\varepsilon \in (0, B^m)$ with $\mathcal{M}(\pi_{k,\varepsilon}) \leq (2^k - 1)\mathcal{M}(\nu_\varepsilon) + (2^{k-1} - 1)20$ holds

$$\begin{aligned} \mathcal{M}(\pi_{k+1,\varepsilon}) &\leq \mathcal{M}(\nu_\varepsilon) + (\mathcal{M}(\pi_{k,\varepsilon}) + \mathcal{M}(\pi_{k,\varepsilon})) + \mathcal{M}_1(\nu_\varepsilon) + \mathcal{M}_{\mathcal{L}(\mathcal{P}(\pi_{k,\varepsilon},\pi_{k,\varepsilon}))}(\mathcal{P}(\pi_{k,\varepsilon}, \pi_{k,\varepsilon})) \\ &\leq \mathcal{M}(\nu_\varepsilon) + 2\mathcal{M}(\pi_{k,\varepsilon}) + 14 + 2\mathcal{M}_{\mathcal{L}(\nu_\varepsilon)}(\nu_\varepsilon) \leq \mathcal{M}(\nu_\varepsilon) + 2\mathcal{M}(\pi_{k,\varepsilon}) + 20 \\ &\leq \mathcal{M}(\nu_\varepsilon) + 2((2^k - 1)\mathcal{M}(\nu_\varepsilon) + (2^{k-1} - 1)20) + 20 \\ &= (2^{k+1} - 1)\mathcal{M}(\nu_\varepsilon) + (2^k - 1)20. \end{aligned} \tag{6.17}$$

This establishes the claim (c).

Combining (a) with Lemma 5.3 and (6.7) implies for every $\varepsilon \in (0, B^m)$ the bound

$$\begin{aligned} \sup_{x \in [-B,B]^m} \left| \left[\prod_{j=1}^{m} x_j\right] - [R_\varrho(\Pi_\varepsilon)](x) \right| &\leq \sup_{x \in [-B,B]^{(2^l)}} \left| \left[\prod_{j=1}^{2^l} x_j\right] - [R_\varrho(\pi_{l,\varepsilon})](x) \right| \\ &\leq 4^{l-1} m^{-2} B^{(2^l - 2m)} \varepsilon \\ &\leq 4^{\lceil \log_2(m) \rceil - 1} m^{-2} B^{(2^{\lceil \log_2(m) \rceil} - 2m)} \varepsilon \\ &\leq 4^{\log_2(m)} m^{-2} B^{(2^{\log_2(m)+1} - 2m)} \varepsilon \\ &\leq \left[2^{\log_2(m)}\right]^2 m^{-2} B^{(2m-2m)} \varepsilon \leq \varepsilon. \end{aligned} \tag{6.18}$$

This and (6.8) establish that the neural networks $(\Pi_\varepsilon)_{\varepsilon \in (0,\infty)}$ satisfy (iii). Combining (b) with Lemma 5.3, (6.3), and (6.7) ensures that for every $\varepsilon \in (0, B^m)$

$$\begin{aligned} \mathcal{L}(\Pi_\varepsilon) &= \mathcal{L}(\pi_{l,\varepsilon}) + \mathcal{L}(\omega) \leq l\mathcal{L}(\nu_\varepsilon) + 1 \leq (\log_2(m) + 1)\mathcal{L}(\nu_\varepsilon) + 1 \\ &\leq \log_2(m) \log_2(\tfrac{1}{\varepsilon}) + 4\log_2(m) m \log_2(B) + 2[\log_2(m)]^2 + 12\log_2(m) + 1. \end{aligned} \tag{6.19}$$

and that for every $\varepsilon \in (B^m, \infty)$ it holds $\mathcal{L}(\Pi_\varepsilon) = \mathcal{L}(\theta) = 1$. This establishes that the neural networks $(\Pi_\varepsilon)_{\varepsilon \in (0,\infty)}$ satisfy (i). Furthermore, note that (c), Lemma 5.3, (6.3), and (6.7) demonstrate that for every $\varepsilon \in (0, B^m)$

$$\begin{aligned} \mathcal{M}(\Pi_\varepsilon) &\leq 2(\mathcal{M}(\pi_{l,\varepsilon}) + \mathcal{M}(\omega)) \leq 2\left[(2^l - 1)\mathcal{M}(\nu_\varepsilon) + (2^{l-1} - 1)20\right] + 4m \\ &\leq 2^{l+1}\mathcal{M}(\nu_\varepsilon) + (2^l)20 + 4m \leq 4m\mathcal{M}(\nu_\varepsilon) + 44m \\ &\leq 180m \log_2(\tfrac{1}{\varepsilon}) + 720m^2 \log_2(B) + 360m \log_2(m) + 1080m. \end{aligned} \tag{6.20}$$

and that for every $\varepsilon \in (B^m, \infty)$ holds $\mathcal{M}(\Pi_\varepsilon) = \mathcal{M}(\theta) = 0$. This establishes that the neural networks $(\Pi_\varepsilon)_{\varepsilon \in (0,\infty)}$ satisfy (ii). Note that (iv) follows from (E) by construction. The proof of Theorem 6.3 is thus completed. □

With the above established, it is quite straightforward to get the following result for the approximation of tensor products. Note that the exponential term B^{m-1} in (iii) is unavoidable as result from multiplying m many inaccurate values of magnitude B. For our purposes this will not be an issue since the functions we consider are bounded in absolute value by $B = 1$. This is further not an issue in cases, where the h_j can be approximated by networks whose size scales logarithmically with ε.

Proposition 6.4. *Assume Setting 5.2, let $\varrho\colon \mathbb{R} \to \mathbb{R}$ be the ReLU activation function given by $\varrho(t) = \max\{0, t\}$, let $B \in [1, \infty)$, $m \in \mathbb{N}$, for every $j \in \{1, 2, \ldots, m\}$ let $d_j \in \mathbb{N}$, $\Omega_j \subseteq \mathbb{R}^{d_j}$, and $h_j : \Omega_j \to [-B, B]$, let $(\Phi^j_\varepsilon)_{\varepsilon \in (0,\infty)} \in \mathfrak{N}$, $j \in \{1, 2, \ldots, m\}$, be neural networks which satisfy for every $\varepsilon \in (0, \infty)$, $j \in \{1, 2, \ldots, m\}$*

$$\sup_{t \in \Omega_j} \left| h_j(x) - \left[R_\varrho(\Phi^j_\varepsilon)\right](x) \right| \leq \varepsilon, \tag{6.21}$$

let $\Phi^{\mathcal{P}}_\varepsilon \in \mathfrak{N}$, $\varepsilon \in (0,\infty)$ be given by $\Phi^{\mathcal{P}}_\varepsilon = \mathcal{P}(\Phi^1_\varepsilon, \Phi^2_\varepsilon, \dots, \Phi^m_\varepsilon)$, and let $L_\varepsilon \in \mathbb{N}$, $\varepsilon \in (0,\infty)$ be given by $L_\varepsilon = \max_{j\in\{1,2,\dots,m\}} \mathcal{L}(\Phi^j_\varepsilon)$.
Then there exists a constant $C \in \mathbb{R}$ (which is independent of m, B, ε) and neural networks $(\Psi_\varepsilon)_{\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ which satisfy

(i) $\mathcal{L}(\Psi_\varepsilon) \leq C \ln(m) \left(|\ln(\varepsilon)| + m\ln(B) + \ln(m)\right) + L_\varepsilon$,

(ii) $\mathcal{M}(\Psi_\varepsilon) \leq Cm \left(|\ln(\varepsilon)| + m\ln(B) + \ln(m)\right) + \mathcal{M}(\Phi^{\mathcal{P}}_\varepsilon) + \mathcal{M}_{L_\varepsilon}(\Phi^{\mathcal{P}}_\varepsilon)$, *and*

(iii) $\sup_{t=(t_1,t_2,\dots,t_m)\in \times_{j=1}^m \Omega_j} \left| \left[\prod_{j=1}^m h_j(t_j)\right] - [R_\varrho(\Psi_\varepsilon)](t)\right| \leq 3mB^{m-1}\varepsilon.$

Proof of Proposition 6.4. In the case of $m = 1$ the neural networks $(\Phi^1_\varepsilon)_{\varepsilon\in(0,\infty)} \in \mathfrak{N}$ satisfy (i), (ii), and (iii) by assumption. Throughout the remainder of this proof assume $m \geq 2$, and let $\theta \in \mathcal{N}^{1,1}_1$ denote the trivial neural network $\theta = (0,0)$. Observe that Theorem 6.3 (with $\varepsilon \leftrightarrow \eta$, $C' \leftrightarrow C$ in the notation Theorem 6.3) ensures that there exist $C' \in \mathbb{R}$ and neural networks $(\Pi_\eta)_{\eta\in(0,\infty)} \subseteq \mathfrak{N}$ which satisfy for every $\eta \in (0,\infty)$ that

(a) $\mathcal{L}(\Pi_\eta) \leq C' \ln(m) \left(|\ln(\eta)| + m\ln(B) + \ln(m)\right)$,

(b) $\mathcal{M}(\Pi_\eta) \leq C'm \left(|\ln(\eta)| + m\ln(B) + \ln(m)\right)$, and

(c) $\sup_{x\in[-B,B]^m} \left| \left[\prod_{j=1}^m x_j\right] - [R_\varrho(\Pi_\eta)](x)\right| \leq \eta.$

Let $(\Psi_\varepsilon)_{\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ be the neural networks which satisfy for every $\varepsilon \in (0,\infty)$ that

$$\Psi_\varepsilon = \begin{cases} \Pi_\varepsilon \odot \mathcal{P}(\Phi^1_\varepsilon, \Phi^2_\varepsilon, \dots, \Phi^m_\varepsilon) & : \varepsilon < \frac{B}{2m} \\ \theta & : \varepsilon \geq \frac{B}{2m} \end{cases}. \tag{6.22}$$

Note that for every $\varepsilon \in (0, \frac{B}{2m})$

$$\begin{aligned} \max_{\substack{x\in[-B,B]^m, x'\in\mathbb{R}^m \\ \|x'-x\|_\infty \leq \varepsilon}} \left| \prod_{j=1}^m x'_j - \prod_{j=1}^m x_j \right| &= (B+\varepsilon)^m - B^m = \sum_{k=1}^m \binom{m}{k} B^{m-k}\varepsilon^k \leq \varepsilon \sum_{k=1}^m \frac{m^k}{k!} B^{m-k} \varepsilon^{k-1} \\ &\leq \varepsilon \sum_{k=1}^m \frac{m^k}{k!} B^{m-k} \left(\frac{B}{2m}\right)^{k-1} = mB^{m-1}\varepsilon \sum_{k=1}^m \frac{1}{2^{k-1}k!} \\ &\leq 2mB^{m-1}\varepsilon. \end{aligned} \tag{6.23}$$

Combining this with Lemma 5.3, Lemma 5.4, (6.21), and (c) implies that for every $\varepsilon \in (0, \frac{B}{2m})$, $t = (t_1, t_2, \dots, t_m) \in \Omega$ it holds

$$\begin{aligned} \left| \left[\prod_{j=1}^m h_j(t_j)\right] - [R_\varrho(\Psi_\varepsilon)](t) \right| &= \left| \left[\prod_{j=1}^m h_j(t_j)\right] - \left[R_\varrho(\Pi_\varepsilon \odot \mathcal{P}(\Phi^1_\varepsilon, \Phi^2_\varepsilon, \dots, \Phi^m_\varepsilon))\right](t) \right| \\ &\leq \left| \left[\prod_{j=1}^m h_j(t_j)\right] - \left[\prod_{j=1}^m \left[R_\varrho(\Phi^j_\varepsilon)\right](t_j)\right] \right| \\ &\quad + \left| \left[\prod_{j=1}^m \left[R_\varrho(\Phi^j_\varepsilon)\right](t_j)\right] - [R_\varrho(\Pi_\varepsilon)]\left([R_\varrho(\Phi^1_\varepsilon)](t_1), \dots, [R_\varrho(\Phi^m_\varepsilon)](t_j)\right) \right| \\ &\leq 2mB^{m-1}\varepsilon + \varepsilon \leq 3mB^{m-1}\varepsilon. \end{aligned} \tag{6.24}$$

Moreover, for every $\varepsilon \in [\frac{B}{2m}, \infty)$, $t = (t_1, t_2, \dots, t_m) \in \Omega$ it holds that

$$\begin{aligned} \left| \left[\prod_{j=1}^m h_j(t_j)\right] - [R_\varrho(\Psi_\varepsilon)](t) \right| &= \left| \left[\prod_{j=1}^m h_j(t_j)\right] - [R_\varrho(\theta)](t) \right| \\ &= \left| \left[\prod_{j=1}^m h_j(t_j)\right] \right| \leq B^m \leq 2mB^{m-1}\varepsilon. \end{aligned} \tag{6.25}$$

This and (6.24) establish that the neural networks $(\Psi_\varepsilon)_{\varepsilon,c\in(0,\infty)}$ satisfy (iii). Next observe that Lemma 5.3, Lemma 5.4, and (a) demonstrate that for every $\varepsilon \in (0, \frac{B}{2m})$

$$\begin{aligned}\mathcal{L}(\Psi_\varepsilon) &= \mathcal{L}(\Pi_\varepsilon \odot \mathcal{P}(\Phi^1_\varepsilon, \Phi^2_\varepsilon, \ldots, \Phi^m_\varepsilon)) = \mathcal{L}(\Pi_\varepsilon) + \max_{j\in\{1,2,\ldots,m\}} \mathcal{L}(\Phi^j_\varepsilon)\\ &\le C' \ln(m)\left(|\ln(\varepsilon)| + m\ln(B) + \ln(m)\right) + L_\varepsilon.\end{aligned} \tag{6.26}$$

This and the fact that for every $\varepsilon \in [\frac{B}{2m}, \infty)$ it holds that $\mathcal{L}(\Psi_\varepsilon) = \mathcal{L}(\theta) = 1$ establish that the neural networks $(\Psi_\varepsilon)_{\varepsilon,c\in(0,\infty)}$ satisfy (i). Furthermore note that Lemma 5.3, Lemma 5.4, and (b) ensure that for every $\varepsilon \in (0, \frac{B}{2m})$

$$\begin{aligned}\mathcal{M}(\Psi_\varepsilon) &= \mathcal{M}(\Pi_\varepsilon \odot \mathcal{P}(\Phi^1_\varepsilon, \Phi^2_\varepsilon, \ldots, \Phi^m_\varepsilon))\\ &\le 2\mathcal{M}(\Pi_\varepsilon) + \mathcal{M}(\mathcal{P}(\Phi^1_\varepsilon, \Phi^2_\varepsilon, \ldots, \Phi^m_\varepsilon)) + \mathcal{M}_{\mathcal{L}(\mathcal{P}(\Phi^1_\varepsilon,\Phi^2_\varepsilon,\ldots,\Phi^m_\varepsilon))}(\mathcal{P}(\Phi^1_\varepsilon, \Phi^2_\varepsilon, \ldots, \Phi^m_\varepsilon))\\ &\le 2C'm\left(|\ln(\varepsilon)| + m\ln(B) + \ln(m)\right) + \mathcal{M}(\Phi^{\mathcal{P}}_\varepsilon) + \mathcal{M}_{L_\varepsilon}(\Phi^{\mathcal{P}}_\varepsilon).\end{aligned} \tag{6.27}$$

This and the fact that for every $\varepsilon \in [\frac{B}{2m}, \infty)$ it holds that $\mathcal{M}(\Psi_\varepsilon) = \mathcal{M}(\theta) = 0$ imply the neural networks $(\Psi_\varepsilon)_{\varepsilon,c\in(0,\infty)}$ satisfy (ii). The proof of Proposition 6.4 is completed. □

Another way to use the multiplication results is to consider the approximation of smooth functions by polynomials. This can be done for functions of arbitrary dimension using the multivariate Taylor expansion (see [44] and [33, Thm. 2.3]). Such a direct approach, however, yields networks whose size depends exponentially on the dimension of the function. As our goal is to show that high-dimensional functions with a tensor product structure can be approximated by networks with only polynomial dependence on the dimension, we only consider univariate smooth functions here. In the appendix we present a detailed and explicit construction of this Taylor approximation by neural networks. In the following results we employ an auxiliary parameter r, so that the bounds on the depth and connectivity of the networks may be stated for all $\varepsilon \in (0,\infty)$. Note that this parameter does not influence the construction of the networks themselves.

Theorem 6.5. *Assume Setting 5.1, let $n \in \mathbb{N}$, $r \in (0,\infty)$, let $\varrho\colon \mathbb{R} \to \mathbb{R}$ be the ReLU activation function given by $\varrho(t) = \max\{0,t\}$, and let $B^n_1 \subseteq C^n([0,1],\mathbb{R})$ be the set given by*

$$B^n_1 = \left\{ f \in C^n([0,1],\mathbb{R})\colon \max_{k\in\{0,1,\ldots,n\}} \left[\sup_{t\in[0,1]} \left|f^{(k)}(t)\right|\right] \le 1 \right\}. \tag{6.28}$$

Then there exist neural networks $(\Phi_{f,\varepsilon})_{f\in B^n_1,\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ which satisfy

(i) $\displaystyle\sup_{f\in B^n_1,\varepsilon\in(0,\infty)} \left[\frac{\mathcal{L}(\Phi_{f,\varepsilon})}{\max\{r, |\ln(\varepsilon)|\}}\right] < \infty$,

(ii) $\displaystyle\sup_{f\in B^n_1,\varepsilon\in(0,\infty)} \left[\frac{\mathcal{M}(\Phi_{f,\varepsilon})}{\varepsilon^{-\frac{1}{n}} \max\{r, |\ln(\varepsilon)|\}}\right] < \infty$, *and*

(iii) for every $f \in B^n_1$, $\varepsilon \in (0,\infty)$ that

$$\sup_{t\in[0,1]} |f(t) - [R_\varrho(\Phi_{f,\varepsilon})](t)| \le \varepsilon. \tag{6.29}$$

For convenience of use we also provide the following more general corollary.

Corollary 6.6. *Assume Setting 5.1, let $r \in (0,\infty)$ and let $\varrho\colon \mathbb{R} \to \mathbb{R}$ be the ReLU activation function given by $\varrho(t) = \max\{0,t\}$. Let further the set $\mathcal{C}^n$ be given by $\mathcal{C}^n = \cup_{[a,b]\subseteq\mathbb{R}_+} C^n([a,b],\mathbb{R})$, and let $\|\cdot\|_{n,\infty} : \mathcal{C}^n \to [0,\infty)$ satisfy for every $[a,b] \subseteq \mathbb{R}_+$, $f \in C^n([a,b],\mathbb{R})$*

$$\|f\|_{n,\infty} = \max_{k\in\{0,1,\ldots,n\}} \left[\sup_{t\in[a,b]} \left|f^{(k)}(t)\right|\right]. \tag{6.30}$$

Then there exist neural networks $(\Phi_{f,\varepsilon})_{f\in\mathcal{C}^n,\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ which satisfy *(i)* $$\sup_{f\in\mathcal{C}^n,\varepsilon\in(0,\infty)}\left[\frac{\mathcal{L}(\Phi_{f,\varepsilon})}{\max\{r,|\ln(\frac{\varepsilon}{\max\{1,b-a\}\|f\|_{n,\infty}})|\}}\right]<\infty,$$

(ii) $$\sup_{f\in\mathcal{C}^n,\varepsilon\in(0,\infty)}\left[\frac{\mathcal{M}(\Phi_{f,\varepsilon})}{\max\{1,b-a\}\,\|f\|_{n,\infty}^{\frac{1}{n}}\,\varepsilon^{-\frac{1}{n}}\max\{r,|\ln(\frac{\varepsilon}{\max\{1,b-a\}\|f\|_{n,\infty}})|\}}\right]<\infty,\text{ and}$$

(iii) for every $[a,b]\subseteq\mathbb{R}_+$, $f\in C^n([a,b],\mathbb{R})$, $\varepsilon\in(0,\infty)$ that

$$\sup_{t\in[a,b]}|f(t)-[R_\varrho(\Phi_{f,\varepsilon})](t)|\leq\varepsilon. \tag{6.31}$$

7 DNN Expression Rates for High-Dimensional Basket prices

Now that we have established a number of general expression rate results, we can apply them to our specific problem. Using the regularity result (3.3) we obtain the following.

Corollary 7.1. *Assume Setting 5.1, let $n\in\mathbb{N}$, $r\in(0,\infty)$, $a\in(0,\infty)$, $b\in(a,\infty)$, let $\varrho\colon\mathbb{R}\to\mathbb{R}$ be the ReLU activation function given by $\varrho(t)=\max\{0,t\}$, let $f\colon(0,\infty)\to\mathbb{R}$ be as defined in* (3.1), *and let $h_{c,K}\colon[a,b]\to\mathbb{R}$, $c\in(0,\infty)$, $K\in[0,\infty)$, denote the functions which satisfy for every $c\in(0,\infty)$, $K\in[0,\infty)$, $x\in[a,b]$ that*

$$h_{c,K}(x)=f(\tfrac{K+c}{x}). \tag{7.1}$$

Then there exist neural networks $(\Phi_{\varepsilon,c,K})_{\varepsilon,c\in(0,\infty),K\in[0,\infty)}\subseteq\mathfrak{N}$ which satisfy

(i) $$\sup_{\varepsilon,c\in(0,\infty),K\in[0,\infty)}\left[\frac{\mathcal{L}(\Phi_{\varepsilon,c,K})}{\max\{r,|\ln(\varepsilon)|\}+\max\{0,\ln(K+c)\}}\right]<\infty,$$

(ii) $$\sup_{\varepsilon,c\in(0,\infty),K\in[0,\infty)}\left[\frac{\mathcal{M}(\Phi_{\varepsilon,c,K})}{(K+c+1)^{\frac{1}{n}}\varepsilon^{-\frac{1}{n^2}}}\right]<\infty,\text{ and}$$

(iii) for every $\varepsilon,c\in(0,\infty)$, $K\in[0,\infty)$ that

$$\sup_{x\in[a,b]}|h_{c,K}(x)-[R_\varrho(\Phi_{\varepsilon,c,K})](x)|\leq\varepsilon. \tag{7.2}$$

Proof of Corollary 7.1. We observe Corollary 3.3 ensures the existence of a constant $C\in\mathbb{R}$ with

$$\max_{k\leq n}\sup_{x\in[a,b]}\left|h_{c,K}^{(k)}(x)\right|\leq C\max\{(K+c)^n,1\}. \tag{7.3}$$

Moreover, observe for every $\varepsilon,c\in(0,\infty)$, $K\in[0,\infty)$ it holds

$$\begin{aligned}&\max\{r,|\ln(\tfrac{\varepsilon}{\max\{1,b-a\}C\max\{(K+c)^n,1\}})|\}\\&\leq\max\{r,|\ln(\varepsilon)|\}+|\ln(\max\{1,b-a\})|+|\ln(C\max\{(K+c)^n,1\})|\\&\leq\max\{r,|\ln(\varepsilon)|\}+\ln(\max\{1,b-a\})+|\ln(C)|+|\ln(\max\{(K+c)^n,1\})|\\&\leq\max\{r,|\ln(\varepsilon)|\}+\ln(\max\{1,b-a\})+|\ln(C)|+n\max\{\ln(K+c),0\}\\&\leq n(1+\max\{1,\tfrac{1}{r}\}(|\ln(C)|+\ln(\max\{1,b-a\})))(\max\{r,|\ln(\varepsilon)|\}+\max\{\ln(K+c),0\}).\end{aligned} \tag{7.4}$$

Furthermore, note for every $\varepsilon,c\in(0,\infty)$, $K\in[0,\infty)$ it holds

$$\begin{aligned}\left[\frac{\varepsilon}{\max\{1,b-a\}C\max\{(K+c)^n,1\}}\right]^{-\frac{1}{2n^2}}&=[\max\{1,b-a\}]^{-\frac{1}{2n^2}}\varepsilon^{-\frac{1}{2n^2}}C^{\frac{1}{2n^2}}\max\{(K+c)^{\frac{1}{2n}},1\}\\&\leq[\max\{1,b-a\}]^{-\frac{1}{2n^2}}C^{\frac{1}{2n^2}}(K+c+1)^{\frac{1}{2n}}\varepsilon^{-\frac{1}{2n^2}}.\end{aligned} \tag{7.5}$$

Combining this, (7.3), (7.4) with Lemma A.1 and Corollary 6.6 (with $n\leftrightarrow 2n^2$ in the notation of Corollary 6.6) completes the proof of Corollary 7.1. □

We can then employ Proposition 6.4 in order to approximate the required tensor product.

Corollary 7.2. *Assume Setting 5.1, let $\varrho\colon \mathbb{R}\to\mathbb{R}$ be the ReLU activation function given by $\varrho(t)=\max\{0,t\}$, let $n\in\mathbb{N}$, $a\in(0,\infty)$, $b\in(a,\infty)$, $(K_i)_{i\in\mathbb{N}}\subseteq[0,K_{\max})$, and consider, for $h_{c,K}\colon[a,b]\to\mathbb{R}$, $c\in(0,\infty)$, $K\in[0,K_{\max})$, the functions which are, for every $c\in(0,\infty)$, $K\in[0,K_{\max})$, $x\in[a,b]$, given by*

$$h_{c,K}(x)=\tfrac{1}{\sqrt{2\pi}}\int_{-\infty}^{\ln(\frac{K+c}{x})}e^{-\frac{1}{2}r^2}\mathrm{d}r. \tag{7.6}$$

For any $c\in(0,\infty)$, $d\in\mathbb{N}$ let the function $F_c^d(x)\colon[a,b]^d\to\mathbb{R}$ be given by

$$F_c^d(x)=1-\left[\prod_{i=1}^d h_{c,K_i}(x_i)\right]. \tag{7.7}$$

Then there exist neural networks $(\Psi^d_{\varepsilon,c})_{\varepsilon,c\in(0,\infty),d\in\mathbb{N}}\subseteq\mathfrak{N}$ which satisfy

(i) $\sup_{\varepsilon,c\in(0,\infty),d\in\mathbb{N}}\left[\frac{\mathcal{L}(\Psi^d_{\varepsilon,c})}{\max\{1,\ln(d)\}(|\ln(\varepsilon)|+\ln(d)+1)+\ln(c+1)}\right]<\infty$,

(ii) $\sup_{\varepsilon,c\in(0,\infty),d\in\mathbb{N}}\left[\frac{\mathcal{M}(\Psi^d_{\varepsilon,c})}{(c+1)^{\frac{1}{n}}d^{1+\frac{1}{n}}\varepsilon^{-\frac{1}{n}}}\right]<\infty$, *and*

(iii) for every $\varepsilon,c\in(0,\infty)$, $d\in\mathbb{N}$ that

$$\sup_{x\in[a,b]^d}\left|F_c^d(x)-\left[R_\varrho(\Psi^d_{\varepsilon,c})\right](x)\right|\le\varepsilon. \tag{7.8}$$

Proof of Corollary 7.2. Throughout this proof assume Setting 5.2. Property Corollary 7.1 ensures there exist constants $b_L,b_M\in(0,\infty)$ and neural networks $\left(\Phi^i_{\eta,c}\right)_{\eta,c\in(0,\infty)}\subseteq\mathfrak{N}$, $i\in\mathbb{N}$ such that for every $i\in\mathbb{N}$ it holds

(a) $\sup_{\eta,c\in(0,\infty)}\left[\frac{\mathcal{L}(\Phi^i_{\eta,c})}{\max\{1,|\ln(\eta)|\}+\max\{0,\ln(K_{\max}+c)\}}\right]<b_L$,

(b) $\sup_{\eta,c\in(0,\infty)}\left[\frac{\mathcal{M}(\Phi^i_{\eta,c})}{(K_{\max}+c+1)^{\frac{1}{n}}\eta^{-\frac{1}{n^2}}}\right]<b_M$, and

(c) for every $\eta,c\in(0,\infty)$ that

$$\sup_{x\in[a,b]}\left|h_{c,K_i}(x)-\left[R_\varrho(\Phi^i_{\eta,c})\right](x)\right|\le\eta. \tag{7.9}$$

Furthermore, for every $c\in(0,\infty)$, $i\in\mathbb{N}$, $x\in[a,b]$ holds

$$|h_{c,K_i}(x)|=\left|\tfrac{1}{\sqrt{2\pi}}\int_{-\infty}^{\ln(\frac{K_i+c}{x})}e^{-\frac{1}{2}r^2}\mathrm{d}r\right|\le\tfrac{1}{\sqrt{2\pi}}\left|\int_{-\infty}^{\infty}e^{-\frac{1}{2}r^2}\mathrm{d}r\right|=1. \tag{7.10}$$

Combining this with (a) and Proposition 6.4 and Lemma 5.4 implies there exist $C\in\mathbb{R}$ and neural networks $(\psi^d_{\eta,c})_{\eta\in(0,\infty)}\subseteq\mathfrak{N}$, $c\in(0,\infty)$, $d\in\mathbb{N}$, such that for every $c\in(0,\infty)$, $d\in\mathbb{N}$ it holds

(A) $\mathcal{L}(\psi^d_{\eta,c})\le C\ln(d)\,(|\ln(\eta)|+\ln(d))+\max_{i\in\{1,2,\dots,d\}}\mathcal{L}(\Phi^i_{\eta,c})$,

(B) $\mathcal{M}(\psi^d_{\eta,c})\le Cd\,(|\ln(\eta)|+\ln(d))+4\sum_{i=1}^d\mathcal{M}(\Phi^i_{\eta,c})+8d\max_{i\in\{1,2,\dots,d\}}\mathcal{L}(\Phi^i_{\eta,c})$, and

(C) for every $\eta \in (0,\infty)$ that

$$\sup_{x\in[a,b]^d}\left|\left[\prod_{i=1}^{d} h_{c,K_i}(x_i)\right] - \big[R_\varrho(\psi^d_{\eta,c})\big](x)\right| \le 3d\eta. \tag{7.11}$$

Let $\lambda \in \mathcal{N}_1^{1,1}$ be the neural network given by $\lambda = ((-1,1))$, let $\theta \in \mathcal{N}_1^{1,1}$ be the neural network given by $\theta = (0,0)$, and let $(\Psi^d_{\varepsilon,c})_{\varepsilon,c\,\in(0,\infty),d\in\mathbb{N}} \subseteq \mathfrak{N}$ be the neural networks given by

$$\Psi^d_{\varepsilon,c} = \begin{cases} \lambda \odot \psi^d_{\varepsilon/(3d),c} & : \varepsilon \le 2 \\ \theta & : \varepsilon > 2 \end{cases}. \tag{7.12}$$

Observe that this and (B) imply for every $\varepsilon \in (0,2]$, $c \in (0,\infty)$, $d \in \mathbb{N}$, $x \in [a,b]^d$ it holds

$$\begin{aligned}\big|F^d_c(x) - \big[R_\varrho(\Psi^d_{\varepsilon,c})\big](x)\big| &= \left|\left(1 - \left[\prod_{i=1}^{d} h_{c,K_i}(x_i)\right]\right) - \left(1 - \left[R_\varrho(\psi^d_{\varepsilon/(3d),c})\right](x)\right)\right| \\ &\le 3d\tfrac{\varepsilon}{3d} = \varepsilon.\end{aligned} \tag{7.13}$$

Moreover, (7.12) and (7.10) ensure for every $\varepsilon \in (2,\infty)$, $c \in (0,\infty)$, $d \in \mathbb{N}$, $x \in [a,b]^d$ it holds

$$\big|F^d_c(x) - \big[R_\varrho(\Psi^d_{\varepsilon,c})\big](x)\big| = \left|\left(1 - \left[\prod_{i=1}^{d} h_{c,K_i}(x_i)\right]\right)\right| \tag{7.14}$$

This and (7.13) establish the neural networks $(\Psi^d_{\varepsilon,c})_{\varepsilon,c\,\in(0,\infty),d\in\mathbb{N}}$ satisfy (iii). Next observe that for every $c \in (0,\infty)$ it holds

$$\begin{aligned}\max\{0, \ln(K_{\max} + c)\} &\le \max\{0, \ln(\max\{1, K_{\max}\} + \max\{1, K_{\max}\}c)\} \\ &= \ln(\max\{1, K_{\max}\}(1+c)) = \ln(\max\{1, K_{\max}\}) + \ln(1+c) \\ &\le \ln(c+1) + |\ln(K_{\max})|.\end{aligned} \tag{7.15}$$

Hence, we obtain that for every $\varepsilon, c \in (0,\infty)$, $d \in \mathbb{N}$ it holds

$$\begin{aligned}&\max\{1, |\ln(\tfrac{\varepsilon}{3d})|\} + \max\{0, \ln(K_{\max} + c)\} \\ &\le |\ln(\varepsilon)| + \ln(d) + \ln(3) + \ln(c+1) + |\ln(K_{\max})| \\ &\le (\ln(3) + |\ln(K_{\max})|)\,[\max\{1, \ln(d)\}(|\ln(\varepsilon)| + \ln(d) + 1) + \ln(c+1)]\,.\end{aligned} \tag{7.16}$$

In addition, for every $\varepsilon, c \in (0,\infty)$, $d \in \mathbb{N}$ it holds

$$C\ln(d)\left(\left|\ln(\tfrac{\varepsilon}{3d})\right| + \ln(d)\right) \le 4C\,[\max\{1, \ln(d)\}(|\ln(\varepsilon)| + \ln(d) + 1) + \ln(c+1)]\,. \tag{7.17}$$

Combining this with Lemma 5.3, (a), (A), and (7.16) yields

$$\begin{aligned}&\sup_{\substack{\varepsilon\in(0,2],c\,\in(0,\infty),\\ d\in\mathbb{N}}}\left[\frac{\mathcal{L}(\Psi^d_{\varepsilon,c})}{\max\{1, \ln(d)\}(|\ln(\varepsilon)| + \ln(d) + 1) + \ln(c+1)}\right] \\ &\le \sup_{\substack{\varepsilon\in(0,2],c\,\in(0,\infty),\\ d\in\mathbb{N}}}\left[\frac{1 + C\ln(d)\left(\left|\ln(\frac{\varepsilon}{3d})\right| + \ln(d)\right) + \max_{i\in\{1,2,\ldots,d\}}\mathcal{L}(\Phi^i_{\varepsilon/(3d),c})}{\max\{1, \ln(d)\}(|\ln(\varepsilon)| + \ln(d) + 1) + \ln(c+1)}\right] \\ &\le 2 + 4C + (\ln(3) + |\ln(K_{\max})|)b_L < \infty.\end{aligned} \tag{7.18}$$

Moreover, (7.12) shows

$$\begin{aligned}&\sup_{\substack{\varepsilon\in(2,\infty),c\,\in(0,\infty),\\ d\in\mathbb{N}}}\left[\frac{\mathcal{L}(\Psi^d_{\varepsilon,c})}{\max\{1, \ln(d)\}(|\ln(\varepsilon)| + \ln(d) + 1) + \ln(c+1)}\right] \\ &= \sup_{\substack{\varepsilon\in(2,\infty),c\,\in(0,\infty),\\ d\in\mathbb{N}}}\left[\frac{1}{\max\{1, \ln(d)\}(|\ln(\varepsilon)| + \ln(d) + 1) + \ln(c+1)}\right] < \infty.\end{aligned} \tag{7.19}$$

This and (7.18) establish that $(\Psi^d_{\varepsilon,c})_{\varepsilon,c\,\in(0,\infty),d\in\mathbb{N}}$ satisfy (i). Next observe Lemma A.1 implies that

- for every $\varepsilon \in (0, 2]$ it holds

$$|\ln(\varepsilon)| \le \left[\sup_{\delta \in [\exp(-2n^2), 2]} \ln(\delta) \right] \varepsilon^{-\frac{1}{n}} = 2n^2 \varepsilon^{-\frac{1}{n}}, \tag{7.20}$$

- for every $d \in \mathbb{N}$ it holds

$$\ln(d) \le \left[\max_{k \in \{1,2,\ldots,\exp(2n^2)\}} \ln(k) \right] d^{\frac{1}{n}} = 2n^2 d^{\frac{1}{n}}, \tag{7.21}$$

- and for every $c \in (0, \infty)$ it holds

$$\ln(c+1) \le \left[\sup_{t \in (0, \exp(2n^2 - 1)]} \ln(t+1) \right] (c+1)^{\frac{1}{n}} = 2n^2 (c+1)^{\frac{1}{n}}. \tag{7.22}$$

For every $m \in \mathbb{N}$, $x_i \in [1, \infty)$, $i \in \{1, 2, \ldots, m\}$, it holds

$$\sum_{i=1}^{m} x_i \le \prod_{i=1}^{m} (x_i + 1) \le 2^m \prod_{i=1}^{m} x_i. \tag{7.23}$$

Combining this with (7.20), (7.21), and (7.22) shows for every $\varepsilon \in (0, 2]$, $d \in \mathbb{N}$, $c \in (0, \infty)$ it holds

$$\begin{aligned} 2Cd(|\ln(\tfrac{\varepsilon}{3d})| + \ln(d)) &\le 2Cd(|\ln(\varepsilon)| + 2\ln(d) + \ln(3) + \ln(c+1)) \\ &\le 4n^2 Cd(2\varepsilon^{-\frac{1}{n}} + 2d^{\frac{1}{n}} + \ln(3) + (c+1)^{\frac{1}{n}}) \\ &\le 1024 n^2 C (c+1)^{\frac{1}{n}} d^{1+\frac{1}{n}} \varepsilon^{-\frac{1}{n}}. \end{aligned} \tag{7.24}$$

Furthermore, note (7.15), (7.20), (7.21), (7.22), and (7.23) ensure for every $\varepsilon \in (0, 2]$, $d \in \mathbb{N}$, $c \in (0, \infty)$ it holds

$$\begin{aligned} &16d(\max\{1, |\ln(\tfrac{\varepsilon}{3d})|\} + \max\{0, \ln(K_{\max} + c)\}) \\ &\le 16d(|\ln(\varepsilon)| + \ln(d) + \ln(3) + \ln(c+1) + |\ln(K_{\max})|) \\ &\le 32n^2 d(2\varepsilon^{-\frac{1}{n}} + d^{\frac{1}{n}} + (c+1)^{\frac{1}{n}} + \ln(3) + |\ln(K_{\max})|) \\ &\le 2048 n^2 (\ln(3) + |\ln(K_{\max})|)(c+1)^{\frac{1}{n}} d^{1+\frac{1}{n}} \varepsilon^{-\frac{1}{n}}. \end{aligned} \tag{7.25}$$

In addition, observe that for every $\varepsilon \in (0, 2]$, $d \in \mathbb{N}$, $c \in (0, \infty)$ it holds

$$4d(K_{\max} + c + 1)^{\frac{1}{n}} (\tfrac{\varepsilon}{3d})^{-\frac{1}{n^2}} \le 96 \max\{1, K_{\max}\}(c+1)^{\frac{1}{n}} d^{1+\frac{1}{n}} \varepsilon^{-\frac{1}{n}}. \tag{7.26}$$

Combining this with Lemma 5.3, (a), (b), (B), (7.24), and (7.25) yield

$$\begin{aligned} &\sup_{\substack{\varepsilon \in (0,2], c \in (0,\infty), \\ d \in \mathbb{N}}} \left[\frac{\mathcal{M}(\Psi^d_{\varepsilon,c})}{(c+1)^{\frac{1}{n}} d^{1+\frac{1}{n}} \varepsilon^{-\frac{1}{n}}} \right] \\ &\le \sup_{\substack{\varepsilon \in (0,2], c \in (0,\infty), \\ d \in \mathbb{N}}} \left[\frac{4 + 2Cd(|\ln(\frac{\varepsilon}{3d})| + \ln(d)) + 8 \sum_{i=1}^{d} \mathcal{M}(\Phi^i_{\varepsilon/(3d),c}) + 16d \max_{i \in \{1,2,\ldots,d\}} \mathcal{L}(\Phi^i_{\varepsilon/(3d),c})}{(c+1)^{\frac{1}{n}} d^{1+\frac{1}{n}} \varepsilon^{-\frac{1}{n}}} \right] \\ &\le 8 + 1024 n^2 C + 96 \max\{1, K_{\max}\} b_M + 2048 n^2 (\ln(3) + |\ln(K_{\max})|) b_L < \infty. \end{aligned} \tag{7.27}$$

Furthermore, note that (7.12) ensures

$$\sup_{\substack{\varepsilon \in (2,\infty), c \in (0,\infty), \\ d \in \mathbb{N}}} \left[\frac{\mathcal{M}(\Psi^d_{\varepsilon,c})}{(c+1)^{\frac{1}{n}} d^{1+\frac{1}{n}} \varepsilon^{-\frac{1}{n}}} \right] = \sup_{\substack{\varepsilon \in (2,\infty), c \in (0,\infty), \\ d \in \mathbb{N}}} \left[\frac{\mathcal{M}(\theta)}{(c+1)^{\frac{1}{n}} d^{1+\frac{1}{n}} \varepsilon^{-\frac{1}{n}}} \right] = 0. \tag{7.28}$$

This and (7.27) establish that the neural networks $(\Psi^d_{\varepsilon,c})_{\varepsilon, c \in (0,\infty), d \in \mathbb{N}}$ satisfy (ii). Thus the proof of Corollary 7.2 is completed. □

Finally, we add the quadrature estimates from Section 4 to achieve approximation with networks whose size only depends polynomially on the dimension of the problem.

Theorem 7.3. *Assume Setting 5.1, let $\varrho\colon \mathbb{R} \to \mathbb{R}$ be the ReLU activation function given by $\varrho(t) = \max\{0,t\}$, let $n \in \mathbb{N}$, $a \in (0,\infty)$, $b \in (a,\infty)$, $(K_i)_{i\in\mathbb{N}} \subseteq [0, K_{\max})$, and let $F_d\colon (0,\infty) \times [a,b]^d \to \mathbb{R}$, $d \in \mathbb{N}$, be the functions which satisfy for every $d \in \mathbb{N}$, $c \in (0,\infty)$, $x \in [a,b]^d$*

$$F_d(c,x) = 1 - \prod_{i=1}^{d} \left[\tfrac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(\frac{K_i+c}{x_i})} e^{-\frac{1}{2}r^2} \mathrm{d}r \right]. \tag{7.29}$$

Then there exists neural networks $(\Gamma_{d,\varepsilon})_{\varepsilon\in(0,1],d\in\mathbb{N}} \in \mathfrak{N}$ which satisfy

(i) $\sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{\mathcal{L}(\Gamma_{d,\varepsilon})}{\max\{1,\ln(d)\}\,(|\ln(\varepsilon)| + \ln(d) + 1)} \right] < \infty$,

(ii) $\sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{\mathcal{M}(\Gamma_{d,\varepsilon})}{d^{2+\frac{1}{n}}\varepsilon^{-\frac{1}{n}}} \right] < \infty$, *and*

(iii) for every $\varepsilon \in (0,1]$, $d \in \mathbb{N}$ that

$$\sup_{x\in[a,b]^d} \left| \int_0^\infty F_d(c,x)\mathrm{d}c - [R_\varrho(\Gamma_{d,\varepsilon})](x) \right| \le \varepsilon. \tag{7.30}$$

Proof of Theorem 7.3. Throughout this proof assume Setting 5.2, let $S_{b,n} \in \mathbb{R}$ be given by

$$S_{b,n} = 2e^{2(4n+1)}(b+1)^{1+\frac{1}{4n}} \tag{7.31}$$

and let $N_{d,\varepsilon} \in \mathbb{R}$, $d \in \mathbb{N}$, $\varepsilon \in (0,1]$, be given by

$$N_{d,\varepsilon} = S_{b,n} d^{\frac{1}{4n}} \left[\tfrac{\varepsilon}{4}\right]^{-\frac{1}{4n}}. \tag{7.32}$$

Note Lemma 4.3 (with $4n \leftrightarrow n$, $F_x^d(c) \leftrightarrow F_d(x,c)$, $N_{d,\frac{\varepsilon}{2}} \leftrightarrow N_{d,\varepsilon}$, $Q_{d,\frac{\varepsilon}{2}} \leftrightarrow Q_{d,\varepsilon}$ in the notation of Lemma 4.3) ensures that there exist $Q_{d,\varepsilon} \in \mathbb{R}$, $c_{\varepsilon,j}^d \in (0, N_{d,\varepsilon})$, $w_{\varepsilon,j}^d \in [0,\infty)$, $j \in \{1,2,\dots,Q_{d,\varepsilon}\}$, $d \in \mathbb{N}$, $\varepsilon \in (0,1]$ with

$$\sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{Q_{d,\varepsilon}}{d^{1+\frac{1}{2n}}\varepsilon^{-\frac{1}{2n}}} \right] < \infty \tag{7.33}$$

and for every $d \in \mathbb{N}$, $\varepsilon \in (0,1]$ it holds

$$\sup_{x\in[a,b]^d} \left| \int_0^\infty F_d(c,x)\mathrm{d}c - \sum_{j=0}^{Q_{d,\varepsilon}} w_{\varepsilon,j}^d F_d(c_{\varepsilon,j}^d, x) \right| \le \tfrac{\varepsilon}{2} \tag{7.34}$$

and

$$\sum_{j=1}^{Q_{d,\varepsilon}} w_{\varepsilon,j}^d = N_{d,\varepsilon}. \tag{7.35}$$

Furthermore, Corollary 7.2 (with $4n \leftrightarrow n$, $F_{c_{\varepsilon,j}^d}^d(x) \leftrightarrow F_d(x, c_{\varepsilon,j}^d)$) ensures there exist neural networks $(\Psi_{\varepsilon,j}^d)_{\varepsilon\in(0,\infty),d\in\mathbb{N},j\in\{1,2,\dots,Q_{d,\varepsilon}\}} \subseteq \mathfrak{N}$ which satisfy

(a) $\sup_{\varepsilon\in(0,\infty),d\in\mathbb{N}} \left[\frac{\max_{j\in\{1,2,\dots,Q_{d,\varepsilon}\}} \mathcal{L}(\Psi_{\varepsilon,j}^d)}{\max\{1,\ln(d)\}\left(|\ln(\frac{\varepsilon}{2N_{d,\varepsilon}})| + \ln(d) + 1\right) + \ln(N_{d,\varepsilon}+1)} \right] < \infty$,

(b) $\sup_{\varepsilon\in(0,\infty),d\in\mathbb{N}} \left[\frac{\max_{j\in\{1,2,\dots,Q_{d,\varepsilon}\}} \mathcal{M}(\Psi^d_{\varepsilon,j})}{(N_{d,\varepsilon}+1)^{\frac{1}{4n}} d^{1+\frac{1}{4n}} \left[\frac{\varepsilon}{2N_{d,\varepsilon}}\right]^{-\frac{1}{4n}}} \right] < \infty$, and

(c) for every $\varepsilon \in (0,\infty)$, $d \in \mathbb{N}$ that

$$\sup_{x\in[a,b]^d} \left| F_d(c^d_{\varepsilon,j}, x) - \left[R_\varrho(\Psi^d_{\varepsilon,j})\right](x)\right| \le \tfrac{\varepsilon}{2N_{d,\varepsilon}}. \tag{7.36}$$

Let $\mathrm{Id}_{\mathbb{R}^d} \in \mathbb{R}^{d\times d}$, $d \in \mathbb{N}$, be the matrices given by $\mathrm{Id}_{\mathbb{R}^d} = \mathrm{diag}(1,1,\dots,1)$, let $\nabla_{d,q} \in \mathcal{N}_1^{d,dq}$, $d,q\in\mathbb{N}$, be the neural networks given by

$$\nabla_{d,q} = \left((\begin{pmatrix} \mathrm{Id}_d \\ \vdots \\ \mathrm{Id}_d \end{pmatrix}, 0) \right), \tag{7.37}$$

let $\Sigma_{d,\varepsilon} \in \mathcal{N}_1^{d,1}$, $d\in\mathbb{N}$, $\varepsilon\in(0,1]$, be the neural networks given by

$$\Sigma_{d,\varepsilon} = \left(\left(\begin{pmatrix} w^d_{\varepsilon,1} & w^d_{\varepsilon,2} & \dots & w^d_{\varepsilon,Q_{d,\varepsilon}}\end{pmatrix}, 0\right)\right), \tag{7.38}$$

and let $(\Gamma_{d,\varepsilon})_{\varepsilon\in(0,1],d\in\mathbb{N}} \in \mathfrak{N}$ be the neural networks given by

$$\Gamma_{d,\varepsilon} = \Sigma_{d,\varepsilon} \odot \mathcal{P}(\Psi^d_{\varepsilon,1}, \Psi^d_{\varepsilon,2}, \dots, \Psi^d_{\varepsilon,Q_{d,\varepsilon}}) \odot \nabla_{d,Q_{d,\varepsilon}}. \tag{7.39}$$

Combining Lemma 5.3, Lemma 5.4, (7.34), (7.35), and (c) implies for every $\varepsilon\in(0,\infty)$ and $d\in\mathbb{N}$, $x\in[a,b]^d$ it holds

$$\begin{aligned}
&\left| \int_0^\infty F_d(c,x)\mathrm{d}c - [R_\varrho(\Gamma_{d,\varepsilon})](x)\right| \\
&\le \left| \int_0^\infty F_d(c,x)\mathrm{d}c - \sum_{j=0}^{Q_{d,\varepsilon}} w^d_{\varepsilon,j} F_d(c^d_{\varepsilon,j},x)\right| + \left| \sum_{j=0}^{Q_{d,\varepsilon}} w^d_{\varepsilon,j} F_d(c^d_{\varepsilon,j},x) - [R_\varrho(\Gamma_{d,\varepsilon})](x)\right| \\
&\le \tfrac{\varepsilon}{2} + \left| \sum_{j=0}^{Q_{d,\varepsilon}} w^d_{\varepsilon,j} F_d(c^d_{\varepsilon,j},x) - \sum_{j=0}^{Q_{d,\varepsilon}} w^d_{\varepsilon,j} \left[R_\varrho(\Psi^d_{\varepsilon,j})\right](x)\right| \\
&\le \tfrac{\varepsilon}{2} + \sum_{j=0}^{Q_{d,\varepsilon}} w^d_{\varepsilon,j} \left| F_d(c^d_{\varepsilon,j},x) - \left[R_\varrho(\Psi^d_{\varepsilon,j})\right](x)\right| \le \tfrac{\varepsilon}{2} + N_{d,\varepsilon}\tfrac{\varepsilon}{2N_{d,\varepsilon}} = \varepsilon.
\end{aligned} \tag{7.40}$$

This establishes that the neural networks $(\Gamma_{d,\varepsilon})_{\varepsilon\in(0,1],d\in\mathbb{N}}$ satisfy (iii). Next, observe for every $\varepsilon\in(0,\infty)$, $d\in\mathbb{N}$

$$\begin{aligned}
&\max\{1,\ln(d)\}\left(|\ln(\frac{\varepsilon}{2N_{d,\varepsilon}})| + \ln(d) + 1\right) + \ln(N_{d,\varepsilon}+1) \\
&\le \max\{1,\ln(d)\}\left(|\ln(\varepsilon)| + \ln(d) + 3\ln(N_{d,\varepsilon}) + \ln(2) + 1\right) \\
&\le \max\{1,\ln(d)\}\left(|\ln(\varepsilon)| + \ln(d) + 3\left(\ln(S_{b,n}) + \frac{1}{4n}\ln(d) + \frac{1}{4n}|\ln(\varepsilon)| + \frac{1}{4n}\ln(4)\right) + 2\right) \\
&\le \max\{1,\ln(d)\}\left(4|\ln(\varepsilon)| + 4\ln(d) + 3\ln(S_{b,n}) + 8\right) \\
&\le \left(3\ln(S_{b,n}) + 8\right)\max\{1,\ln(d)\}\left(|\ln(\varepsilon)| + \ln(d) + 1\right).
\end{aligned} \tag{7.41}$$

Combining this with Lemma 5.3, Lemma 5.4, and (a) implies

$$\begin{aligned}
&\sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{\mathcal{L}(\Gamma_{d,\varepsilon})}{\max\{1,\ln(d)\}\,(|\ln(\varepsilon)|+\ln(d)+1)}\right] \\
&\le \sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{\mathcal{L}(\Sigma_{d,\varepsilon})+\max_{j\in\{1,2,\dots,Q_{d,\varepsilon}\}}\mathcal{L}(\Psi^d_{\varepsilon,j})+\mathcal{L}(\nabla_{d,Q_{d,\varepsilon}})}{\max\{1,\ln(d)\}\,(|\ln(\varepsilon)|+\ln(d)+1)}\right] \\
&\le 2+\sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{\max_{j\in\{1,2,\dots,Q_{d,\varepsilon}\}}\mathcal{L}(\Psi^d_{\varepsilon,j})}{\max\{1,\ln(d)\}\,(|\ln(\varepsilon)|+\ln(d)+1)}\right] \\
&\le 2+(3\ln(S_{b,n})+8)\sup_{\varepsilon\in(0,\infty),d\in\mathbb{N}} \left[\frac{\max_{j\in\{1,2,\dots,Q_{d,\varepsilon}\}}\mathcal{L}(\Psi^d_{\varepsilon,j})}{\max\{1,\ln(d)\}\Big(|\ln(\frac{\varepsilon}{2N_{d,\varepsilon}})|+\ln(d)+1\Big)+\ln(N_{d,\varepsilon}+1)}\right] \\
&< \infty.
\end{aligned} \tag{7.42}$$

This establishes $(\Gamma_{d,\varepsilon})_{\varepsilon\in(0,1],d\in\mathbb{N}}$ satisfy (i). In addition, for every $\varepsilon\in(0,\infty)$, $d\in\mathbb{N}$ it holds

$$\begin{aligned}
(N_{d,\varepsilon}+1)^{\frac{1}{4n}}\, d^{1+\frac{1}{4n}} \left[\frac{\varepsilon}{2N_{d,\varepsilon}}\right]^{-\frac{1}{4n}} &\le 4N_{d,\varepsilon}^{\frac{1}{2n}}\, d^{1+\frac{1}{4n}}\,\varepsilon^{-\frac{1}{4n}} \\
&\le 4\left[S_{b,n} d^{\frac{1}{4n}} \left[\tfrac{\varepsilon}{4}\right]^{-\frac{1}{4n}}\right]^{\frac{1}{2n}} d^{1+\frac{1}{4n}}\,\varepsilon^{-\frac{1}{4n}} \\
&\le 16 S_{b,n} d^{1+\frac{1}{4n}+\frac{1}{4n^2}}\,\varepsilon^{-(\frac{1}{4n}+\frac{1}{8n^2})} \\
&\le 16 S_{b,n} d^{1+\frac{1}{2n}}\,\varepsilon^{-\frac{1}{2n}}.
\end{aligned} \tag{7.43}$$

Combining this with Lemma 5.3, Lemma 5.4, (7.33), (b), and the fact that for every $\psi\in\mathfrak{N}$ which satisfies $\min_{l\in\{1,2,\dots,\mathcal{L}(\psi)\}}\mathcal{M}_l(\psi)>0$ it holds $\mathcal{L}(\psi)\le\mathcal{M}(\psi)$ ensures

$$\begin{aligned}
&\sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{\mathcal{M}(\Gamma_{d,\varepsilon})}{d^{(2+\frac{1}{n})}\varepsilon^{-\frac{1}{n}}}\right] \\
&\le \sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{2\mathcal{M}(\Sigma_{d,\varepsilon})+4\left(2\sum_{j=1}^{Q_{d,\varepsilon}}\mathcal{M}(\Psi^d_{\varepsilon,j})+4Q_{d,\varepsilon}\max_{j\in\{1,2,\dots,Q_{d,\varepsilon}\}}\mathcal{L}(\Psi^d_{\varepsilon,j})\right)+4\mathcal{M}(\nabla_{d,Q_{d,\varepsilon}})}{d^{(2+\frac{1}{n})}\varepsilon^{-\frac{1}{n}}}\right] \\
&\le \sup_{\varepsilon\in(0,1],d\in\mathbb{N}} \left[\frac{24Q_{d,\varepsilon}\max_{j\in\{1,2,\dots,Q_{d,\varepsilon}\}}\mathcal{M}(\Psi^d_{\varepsilon,j})}{d^{(2+\frac{1}{n})}\varepsilon^{-\frac{1}{n}}}\right]+\sup_{\varepsilon\in(0,1],d\in\mathbb{N}}\left[\frac{2Q_{d,\varepsilon}+4dQ_{d,\varepsilon}}{d^{(2+\frac{1}{n})}\varepsilon^{-\frac{1}{n}}}\right] \\
&\le 24\left(\sup_{\varepsilon\in(0,1],d\in\mathbb{N}}\left[\frac{Q_{d,\varepsilon}}{d^{(1+\frac{1}{2n})}\varepsilon^{-\frac{1}{2n}}}\right]\right)\left(\sup_{\varepsilon\in(0,1],d\in\mathbb{N}}\left[\frac{\max_{j\in\{1,2,\dots,Q_{d,\varepsilon}\}}\mathcal{M}(\Psi^d_{\varepsilon,j})}{d^{(1+\frac{1}{2n})}\varepsilon^{-\frac{1}{2n}}}\right]\right) \\
&\quad +4\sup_{\varepsilon\in(0,1],d\in\mathbb{N}}\left[\frac{Q_{d,\varepsilon}}{d^{(1+\frac{1}{n})}\varepsilon^{-\frac{1}{n}}}\right] \\
&\le 24\left(\sup_{\varepsilon\in(0,1],d\in\mathbb{N}}\left[\frac{Q_{d,\varepsilon}}{d^{(1+\frac{1}{2n})}\varepsilon^{-\frac{1}{2n}}}\right]\right)\left(1+16S_{b,n}\sup_{\varepsilon\in(0,1],d\in\mathbb{N}}\left[\frac{\max_{j\in\{1,2,\dots,Q_{d,\varepsilon}\}}\mathcal{M}(\Psi^d_{\varepsilon,j})}{(N_{d,\varepsilon}+1)^{\frac{1}{4n}}\,d^{1+\frac{1}{4n}}\left[\frac{\varepsilon}{2N_{d,\varepsilon}}\right]^{-\frac{1}{4n}}}\right]\right) \\
&< \infty.
\end{aligned} \tag{7.44}$$

This establishes the neural networks $(\Gamma_{d,\varepsilon})_{\varepsilon\in(0,1],d\in\mathbb{N}}$ satisfy (ii). The proof of Theorem 7.3 is thus completed. □

8 Discussion

While Theorem 7.3 only establishes formally that the solution of one specific high-dimensional PDE may be approximated by neural networks without curse of dimensionality, the constructive approach also serves to illustrate that neural networks are capable of accomplishing the same for any PDE solution which exhibits a similar low-rank structure. Note here, that the tensor product construction in Proposition 6.4 only introduces a logarithmic dependency on the approximation accuracy. That we end up with a spectral rate in this specific case is due to Proposition 6.4 and Lemma 4.3, i.e. the insufficient regularity of the univariate functions inside the tensor product, as well as the number of terms required by the Gaussian quadrature used to approximate the outer integral. In particular, this means that the approach in Section 6 might also be used to produce approximation results with connectivity growing only logarithmically in the inverse of the approximation error, given that one has a suitably well behaved low-rank structure.

The present result is a promising step towards higher order, numerical solution of high-dimensional PDEs, which are notoriously troublesome to handle with any of the classical approaches based on discretization of the domain, or with randomized (a.k.a. Monte-Carlo based) arguments. Of course answering the question of approximability can only ensure that there exist networks with a reasonable size-to-accuracy trade-off, whereas for any practical purpose it is also necessary to establish whether and how one can find these networks.

An analysis of the generalization error for linear Kolmogorov equations can be found in [4], which concludes that, under reasonable assumptions, the number of required Monte Carlo samples is free of the curse of dimensionality. Moreover, there are a number of empirical results [2, 3, 10, 23, 41], which suggest that the solutions of various high dimensional PDEs may be learned efficiently using standard stochastic gradient descent based methods. However, a satisfying formal analysis of this training procesdoes not seem to be available at the present.

Lastly we would like to point out that, even though we had a semi-explicit formula available, the ReLU networks we used for approximation were in no way adapted to use this knowledge and have been shown to exhibit excellent approximation properties for, e.g., piecewise smooth functions [36], affine and Gabor systems [12], and even fractal structures [9]. So, while a spline dictionary based approach specifically designed for the approximation of this one PDE solution may have similar rates, it would most certainly lack the remarkable universality of neural networks.

References

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

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

[3] Beck, C., E, W., and Jentzen, A. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science* (2017).

[4] Berner, J., Grohs, P., and Jentzen, A. 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 Journal on Mathematics of Data Science 2* (2020), 631–657.

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

[6] Chiani, M., Dardari, D., and Simon, M. K. New exponential bounds and approximations for the computation of error probability in fading channels. *IEEE Trans. Wireless Communications 2*, 4 (2003), 840–845.

[7] Chui, C., Li, X., and Mhaskar, H. Neural networks for localized approximation. *Mathematics of Computation 63*, 208 (1994), 607–623.

[8] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems 2*, 4 (1989), 303–314.

[9] Dym, N., Sober, B., and Daubechies, I. Expression of fractals through neural network functions. *IEEE Journal on Selected Areas in Information Theory 1*, 1 (2020), 57–66.

[10] E, W., Han, J., and Jentzen, A. 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.

[11] E, W., and Yu, B. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Commun. Math. Stat. 6*, 1 (2018), 1–12.

[12] Elbrächter, D., Perekrestenko, D., Grohs, P., and Bölcskei, H. Deep neural network approximation theory. *arXiv:1901.02220* (2019).

[13] Freidlin, M. *Functional integration and partial differential equations*, vol. 109 of *Annals of Mathematics Studies.* Princeton University Press, Princeton, NJ, 1985.

[14] Fujii, M., Takahashi, A., and Takahashi, M. Asymptotic Expansion as Prior Knowledge in Deep Learning Method for high dimensional BSDEs. *Asia-Pacific Financial Markets 29* (2017), 1563–1619.

[15] Gonon, L., Grohs, P., Jentzen, A., Kofler, D., and Šiška, D. Uniform error estimates for artificial neural network approximations for heat equations. *arXiv:1911.09647* (2019).

[16] Gonon, L., and Schwab, C. Deep ReLU network expression rates for option prices in high-dimensional, exponential Lévy models. Tech. Rep. 2020-52, Seminar for Applied Mathematics, ETH Zürich, 2020.

[17] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[18] Goudenège, L., Molent, A., and Zanette, A. Machine learning for pricing American options in high-dimensional Markovian and non-Markovian models. *Quantitative Finance 20*, 4 (2020), 573–591.

[19] Grohs, P., and Herrmann, L. Deep neural network approximation for high-dimensional elliptic PDEs with boundary conditions. *arXiv:2007.05384* (2020).

[20] Grohs, P., Hornung, F., Jentzen, A., and von Wurstemberger, P. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *arXiv:1809.02362* (2019). Accepted in Mem. Amer. Math. Soc.

[21] Grohs, P., Jentzen, A., and Salimova, D. Deep neural network approximations for Monte Carlo algorithms. *arXiv:1908.10828* (2019).

[22] Hairer, M., Hutzenthaler, M., and Jentzen, A. Loss of regularity for Kolmogorov equations. *Annals of Probability 2015, Vol. 43, No. 2, 468-527* (Mar. 2015).

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

[24] Henry-Labordere, P. Deep Primal-Dual Algorithm for BSDEs: Applications of Machine Learning to CVA and IM. Available at SSRN: https://ssrn.com/abstract=3071506.

[25] Hornik, K., Stinchcombe, M., and White, H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks 3*, 5 (1990), 551–560.

[26] Hornung, F., Jentzen, A., and Salimova, D. Space-time deep neural network approximations for high-dimensional partial differential equations. *arXiv:2006.02199* (2020).

[27] Hutzenthaler, M., Jentzen, A., Kruse, T., and Nguyen, T. A. A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations. *SN Partial Differential Equations and Applications 1*, 10 (2020).

[28] Jentzen, A., Salimova, D., and Welti, T. 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:1809.07321* (2018).

[29] Khoo, Y., Lu, J., and Ying, L. Solving parametric PDE problems with artificial neural networks. *European Journal of Applied Mathematics* (2020).

[30] Kutyniok, G., Petersen, P., Raslan, M., and Schneider, R. A theoretical analysis of deep neural networks and parametric PDEs. *arXiv:1904.00377* (2019).

[31] Kwok, Y.-K. *Mathematical models of financial derivatives*, second ed. Springer Finance. Springer, Berlin, 2008.

[32] Levy, D. Introduction to Numerical Analysis, 2010. Available: https://api.semanticscholar.org/CorpusID:123255603.

[33] Mhaskar, H. N. Neural Networks for optimal approximation of smooth and analytic functions. *Neural Computation 8* (1996), 164–177.

[34] Mishra, S. A machine learning framework for data driven acceleration of computations of differential equations. *Math. in Engg. 1*, 1 (2018), 118–146.

[35] Perekrestenko, D., Grohs, P., Elbrächter, D., and Bölcskei, H. The universal approximation power of finite-width deep ReLU networks. *arXiv:1806.01528* (2018).

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

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

[38] Reisinger, C., and Zhang, Y. Rectified deep neural networks overcome the curse of dimensionality for nonsmooth value functions in zero-sum games of nonlinear stiff systems. *arXiv:1903.06652* (2019).

[39] Schmidt-Hieber, J. Nonparametric regression using deep neural networks with ReLU activation function. *Ann. Statist. 48*, 4 (2020), 1875–1897.

[40] Schwab, C., and Zech, J. Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ. *Analysis and Applications 17*, 01 (2019), 19–55.

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

[42] Telgarsky, M. Representation benefits of deep feedforward networks. *arXiv:1509.0810* (2015).

[43] Wilmott, P. *Paul Wilmott introduces quantitative finance*, 2 ed. Wiley, 2007.

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

A Additional Proofs

A.1 Technical Lemma

Lemma A.1. *It holds for every $r \in (0,\infty)$, $t \in (0, \exp(-2r^2)]$ that*

$$|\ln(t)| \le t^{-1/r} \tag{A.1}$$

and for every $r \in (0,\infty)$, $t \in [\exp(2r^2), \infty)$ that

$$\ln(t) \le t^{1/r}. \tag{A.2}$$

Proof of Lemma A.1. First, observe that for every $r \in (0,\infty)$, $y \in [2r^2,\infty)$ it holds that

$$\exp\Big(\frac{y}{r}\Big) = \sum_{k=0}^{\infty} \left[\frac{y^k}{k! r^k}\right] \ge \frac{y^2}{2! r^2} = y \left[\frac{y}{2r^2}\right] \ge y. \tag{A.3}$$

This implies that for every $r \in (0,\infty)$, $x \in [\exp(2r^2),\infty)$ it holds that

$$x^{1/r} = \exp\Big(\ln\Big(x^{1/r}\Big)\Big) = \exp\Big(\tfrac{\ln(x)}{r}\Big) \ge \ln(x). \tag{A.4}$$

Hence, we obtain that for every $r \in (0,\infty)$, $t \in (0, \exp(-2r^2)] \subseteq (0,1]$ it holds that

$$t^{-1/r} = \left[\tfrac{1}{t}\right]^{1/r} \ge \ln(\tfrac{1}{t}) = |\ln(t)|\,. \tag{A.5}$$

This completes the proof of Lemma A.1. □

A.2 Proof of Lemma 6.1

Proof of Lemma 6.1. The proof follows [44]. We provide it in order to provide values of constants in the bounds on depth and width, and to reveal the dependence on the scaling parameter B. Throughout this proof let $\theta \in \mathcal{N}_1^{1,1}$ be the neural network given by $\theta = (0,0)$, let $g_s\colon [0,1] \to [0,1]$, $s \in \mathbb{N}$, be the functions which satisfy for every $s \in \mathbb{N}$, $t \in [0,1]$ that

$$g_s(t) = \begin{cases} 2t & : s = 1, t < \frac{1}{2} \\ 2 - 2t & : s = 1, t \ge \frac{1}{2} \\ g_1(g_{s-1}(t)) & : s \ge 1 \end{cases}, \tag{A.6}$$

and let $f_m\colon [0,1] \to [0,1]$, $m \in \mathbb{N}$, be the functions which satisfy for every $m \in \mathbb{N}$, $k \in \{0,1,\dots,2^m\}$, $x \in \left[\frac{k}{2^m}, \frac{k+1}{2^m}\right]$ that

$$f_m(x) = \left[\frac{2k+1}{2^m}\right] x - \frac{k^2+k}{2^{2m}}. \tag{A.7}$$

We claim for every $s \in \mathbb{N}$, $k \in \{0,1,\dots,2^{s-1}-1\}$ it holds

$$g_s(x) = \begin{cases} 2^s(x - \frac{2k}{2^s}) & : x \in \left[\frac{2k}{2^s}, \frac{2k+1}{2^s}\right] \\ 2^s(\frac{2k+2}{2^s} - x) & : x \in \left[\frac{2k+1}{2^s}, \frac{2k+2}{2^s}\right] \end{cases}. \tag{A.8}$$

We now prove (A.8) by induction on $s \in \mathbb{N}$. Equation (A.6) establishes (A.8) in the base case $s = 1$. For the induction step $\mathbb{N} \ni s \to s+1 \in \{2,3,\dots\}$ observe that (A.6) implies for every $s \in \mathbb{N}$, $l \in \{0,1,\dots,2^{s-1}-1\}$ that

(a) it holds for every $x \in \left[\frac{2l}{2^s}, \frac{2l+(1/2)}{2^s}\right]$

$$\begin{aligned} g_{s+1}(x) &= g(g_s(x)) = g(2^s(x - \tfrac{2l}{2^s})) = 2\left[2^s(x - \tfrac{2l}{2^s})\right] \\ &= 2^{s+1}(x - \tfrac{2l}{2^s}) = 2^{s+1}(x - \tfrac{2(2l)}{2^{s+1}}). \end{aligned} \tag{A.9}$$

(b) it holds for every $x \in \left[\frac{2l+(1/2)}{2^s}, \frac{2l+1}{2^s}\right]$

$$\begin{aligned} g_{s+1}(x) &= g(g_s(x)) = g(2^s(x - \tfrac{2l}{2^s})) = 2 - 2\left[2^s(x - \tfrac{2l}{2^s})\right] \\ &= 2 - 2^{s+1}x + 4l = 2^{s+1}(\tfrac{4l+2}{2^{s+1}} - x) \\ &= 2^{s+1}(\tfrac{2(2l+1)}{2^{s+1}} - x). \end{aligned} \tag{A.10}$$

(c) it holds for every $x \in \left[\frac{2l+1}{2^s}, \frac{2l+(3/2)}{2^s}\right]$

$$\begin{aligned} g_{s+1}(x) &= g(g_s(x)) = g(2^s(\tfrac{2l+2}{2^s} - x)) = 2 - 2\left[2^s(\tfrac{2l+2}{2^s} - x)\right] \\ &= 2 - 2(2l+2) + 2^{s+1}x = 2^{s+1}x - 2(2l+1) \\ &= 2^{s+1}(x - \tfrac{2(2l+1)}{2^{s+1}}). \end{aligned} \tag{A.11}$$

(d) it holds for every $x \in \left[\frac{2l+(3/2)}{2^s}, \frac{2l+2}{2^s}\right]$

$$\begin{aligned} g_{s+1}(x) &= g(g_s(x)) = g(2^s(\tfrac{2l+2}{2^s} - x)) = 2\left[2^s(\tfrac{2l+2}{2^s} - x)\right] \\ &= 2^{s+1}(\tfrac{2l+2}{2^s} - x) = 2^{s+1}(\tfrac{2(2l+2)}{2^{s+1}} - x). \end{aligned} \tag{A.12}$$

Next observe that for every $s \in \mathbb{N}$, $k \in \{0, 1, \ldots, 2^s - 1\}$ there exists $l \in \{0, 1, \ldots, 2^{s-1} - 1\}$ such that

$$\left[\tfrac{2k}{2^{s+1}}, \tfrac{2k+1}{2^{s+1}}\right] = \left[\tfrac{2l}{2^s}, \tfrac{2l+(1/2)}{2^s}\right] \quad \text{or} \quad \left[\tfrac{2k}{2^{s+1}}, \tfrac{2k+1}{2^{s+1}}\right] = \left[\tfrac{2l+1}{2^s}, \tfrac{2l+(3/2)}{2^s}\right]. \tag{A.13}$$

Furthermore, for every $s \in \mathbb{N}$, $k \in \{0, 1, \ldots, 2^s - 1\}$ there exists $l \in \{0, 1, \ldots, 2^{s-1} - 1\}$ such that

$$\left[\tfrac{2k+1}{2^{s+1}}, \tfrac{2k+2}{2^{s+1}}\right] = \left[\tfrac{2l+(1/2)}{2^s}, \tfrac{2l+1}{2^s}\right] \quad \text{or} \quad \left[\tfrac{2k+1}{2^{s+1}}, \tfrac{2k+2}{2^{s+1}}\right] = \left[\tfrac{2l+(3/2)}{2^s}, \tfrac{2l+2}{2^s}\right]. \tag{A.14}$$

Combining this with (A.9), (A.10), (A.11), (A.12), and (A.13) completes the induction step $\mathbb{N} \ni s \to s+1 \in \{2, 3, \ldots\}$ and thus establishes the claim (A.8).

Next, for every $m \in \mathbb{N}$, $k \in \{0, 1, \ldots, 2^{m-1}\}$ it holds

$$f_{m-1}(\tfrac{2k}{2^m}) - f_m(\tfrac{2k}{2^m}) = f_{m-1}(\tfrac{k}{2^{m-1}}) - f_m(\tfrac{2k}{2^m}) = \left[\tfrac{k}{2^{m-1}}\right]^2 - \left[\tfrac{2k}{2^m}\right]^2 = 0. \tag{A.15}$$

In addition, note that (A.7) implies that for every $m \in \mathbb{N}$, $k \in \{0, 1, \ldots, 2^m - 1\}$ it holds

$$\begin{aligned} f_{m-1}(\tfrac{2k+1}{2^m}) &= f_{m-1}\left(\tfrac{k+\frac{1}{2}}{2^{m-1}}\right) = \left\lceil \frac{2k+1}{2^{m-1}} \right\rceil \frac{k+\frac{1}{2}}{2^{m-1}} - \frac{k^2+k}{2^{2(m-1)}} \\ &= \frac{(2k+1)(k+\frac{1}{2}) - (k^2+k)}{2^{2m-2}} = \frac{k^2+k+\frac{1}{2}}{2^{2m-2}} = \frac{4k^2+4k+2}{2^{2m}} \end{aligned} \tag{A.16}$$

and

$$f_m(\tfrac{2k+1}{2^m}) = \left\lceil \frac{2(2k+1)+1}{2^m} \right\rceil \frac{2k+1}{2^m} - \frac{(2k+1)^2 + (2k+1)}{2^{2m}} = \frac{4k^2+4k+1}{2^{2m}}. \tag{A.17}$$

For every $m \in \mathbb{N}$, $k \in \{0, 1, \ldots, 2^m - 1\}$ it holds

$$f_{m-1}(\tfrac{2k+1}{2^m}) - f_m(\tfrac{2k+1}{2^m}) = \frac{4k^2+4k+2}{2^{2m}} - \frac{4k^2+4k+1}{2^{2m}} = \frac{1}{2^{2m}}. \tag{A.18}$$

Combining this with (A.8), (A.7), and (A.15) demonstrates that for every $m \in \mathbb{N}$, $x \in [0, 1]$ it holds

$$f_{m-1}(x) - f_m(x) = 2^{-2m} g_m(x). \tag{A.19}$$

The fact that for every $x \in [0,1]$ it holds that $f_0(x) = x$ therefore implies that for every $m \in \mathbb{N}_0$, $x \in [0,1]$ it holds

$$f_m(x) = x - \sum_{s=1}^{m} 2^{-2s} g_s(x). \tag{A.20}$$

We observe f_m is the affine, linear interpolant of the twice continuously differentiable function $[0,1] \ni x \mapsto x^2 \in [0,1]$ at the points $\frac{k}{2^m}$, $k \in \{0,1,\dots,2^m\}$. This establishes that for every $m \in \mathbb{N}$

$$\begin{aligned}
\sup_{x\in[0,1]} \left|x^2 - f_m(x)\right| &= \max_{k\in\{0,1,\dots,2^m\}} \left(\sup_{x\in\left[\frac{k}{2^m},\frac{k+1}{2^m}\right]} \left|x^2 - f_m(x)\right| \right) \\
&\leq \max_{k\in\{0,1,\dots,2^m\}} \left(\frac{\left[\frac{k+1}{2^m} - \frac{k}{2^m}\right]^2}{8} \max_{x\in\left[\frac{k}{2^m},\frac{k+1}{2^m}\right]} \left|\frac{\mathrm{d}^2}{\mathrm{d}t^2}\left[x^2\right]\right| \right) \\
&\leq \max_{k\in\{0,1,\dots,2^m\}} \left(\tfrac{1}{8}\left[\tfrac{1}{2^m}\right]^2 \max_{x\in\left[\frac{k}{2^m},\frac{k+1}{2^m}\right]} |2| \right) \\
&= 2^{-2m-2}.
\end{aligned} \tag{A.21}$$

Let $(A_k, b_k) \in \mathbb{R}^{4\times 4} \times \mathbb{R}^4$, $k \in \mathbb{N}$, be the matrix-vector tuples which satisfy for every $k \in \mathbb{N}$

$$A_k = \begin{pmatrix} 2 & -4 & 2 & 0 \\ 2 & -4 & 2 & 0 \\ 2 & -4 & 2 & 0 \\ -2^{-2k+3} & 2^{-2k+4} & -2^{-2k+3} & 1 \end{pmatrix} \quad \text{and} \quad b_k = \begin{pmatrix} 0 \\ -\frac{1}{2} \\ -1 \\ 0 \end{pmatrix}, \tag{A.22}$$

let $\varphi_m \in \mathfrak{N}$, $m \in \mathbb{N}$, be the neural networks which satisfy $\varphi_1 = (1,0)$ and, for every $m \in \mathbb{N}$,

$$\varphi_m = \left(\left(\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{1}{2} \\ -1 \\ 0 \end{pmatrix} \right), (A_2, b_2), \dots, (A_{m-1}, b_{m-1}), \left(\begin{pmatrix} -2^{-2m+3} \\ 2^{-2m+4} \\ -2^{-2m+3} \\ 1 \end{pmatrix}^T, 0 \right) \right). \tag{A.23}$$

Let further $r^k \colon \mathbb{R} \to \mathbb{R}$, $k \in \mathbb{N}$ denote the function which satisfies for every $x \in \mathbb{R}$

$$(r_1^1(x), r_2^1(x), r_3^1(x), r_4^1(x)) = r^1(x) = \varrho^*(x, x - \tfrac{1}{2}, x - 1, x) \tag{A.24}$$

and for every $x \in \mathbb{R}$, $k \in \mathbb{N}$

$$(r_1^k(x), r_2^k(x), r_3^k(x), r_4^k(x)) = r^k(x) = \varrho^*(A_k r_{k-1}(x) + b_k). \tag{A.25}$$

We claim that for every $k \in \{1,2,\dots,m-1\}$, $x \in [0,1]$ it holds

(a)

$$2r_1^k(x) - 4r_2^k(x) + 2r_3^k(x) = g_k(x) \tag{A.26}$$

and

(b)

$$r_4^k(x) = x - \sum_{j=1}^{k-1} 2^{-2j} g_j(x). \tag{A.27}$$

We prove (a) and (b) by induction over $k \in \{1, 2, \ldots, m-1\}$. For the base case $k = 1$ we note that for every $x \in [0, 1]$ it holds

$$g_1(x) = 2\varrho(x) - 4\varrho(x - \tfrac{1}{2}) + 2\varrho(x - 1). \tag{A.28}$$

Hence, we obtain that for every $x \in [0, 1]$ it holds

$$2r_1^1(x) - 4r_2^1(x) + 2r_3^1(x) = 2\varrho(x) - 4\varrho(x - \tfrac{1}{2}) + 2\varrho(x - 1) = g_1(x). \tag{A.29}$$

Furthermore, note that for every $x \in [0, 1]$ it holds that $r_4^1(x) = x$. This and (A.29) establish the base case $k = 1$. For the induction step $\{1, 2, \ldots, m-2\} \ni k-1 \to k \in \{2, 3, \ldots, m-1\}$ observe that (A.28) ensures for every $x \in [0, 1]$, $k \in \{2, 3, \ldots, m-1\}$, with $g_{k-1}(x) = 2r_1^{k-1}(x) - 4r_2^{k-1}(x) + 2r_3^{k-1}(x)$, it holds

$$\begin{aligned} 2r_1^k(x) - 4r_2^k(x) + 2r_3^k(x) = {} & 2\varrho(2r_1^{k-1}(x) - 4r_2^{k-1}(x) + 2r_3^{k-1}(x)) \\ & -4\varrho(2r_1^{k-1}(x) - 4r_2^{k-1}(x) + 2r_3^{k-1}(x) - \tfrac{1}{2}) \\ & +2\varrho(2r_1^{k-1}(x) - 4r_2^{k-1}(x) + 2r_3^{k-1}(x) - 1) \\ = {} & g_1(2r_1^{k-1}(x) - 4r_2^{k-1}(x) + 2r_3^{k-1}(x)) \\ = {} & g_1(g_{k-1}(k)) = g_k(x). \end{aligned} \tag{A.30}$$

Induction thus establishes (a). Moreover note that (A.7) and (A.20) for every $k \in \mathbb{N}$, $x \in [0, 1]$ it holds

$$x - \sum_{j=1}^{k-1} 2^{-2j} g_j(x) = f_{k-1}(x) \geq 0. \tag{A.31}$$

Combining this with (A.28) implies that for every $x \in [0, 1]$, $k \in \{2, 3, \ldots, m-1\}$ with $g_{k-1}(x) = 2r_1^{k-1}(x) - 4r_2^{k-1}(x) + 2r_3^{k-1}(x)$ and $r_4^{k-1}(x) = x - \sum_{j=1}^{k-2} 2^{-2j} g_j(x)$ it holds

$$\begin{aligned} r_4^k(x) &= \varrho(-2^{-2k+3} r_1^{k-1}(x) + 2^{-2k+4} r_2^{k-1}(x) - 2^{-2k+3} r_3^{k-1}(x) + r_4^{k-1}(x)) \\ &= \varrho(x - \sum_{j=1}^{k-2} 2^{-2j} g_j(x) - g_{k-1}(x)) = \varrho(x - \sum_{j=1}^{k-1} 2^{-2j} g_j(x)) \\ &= x - \sum_{j=1}^{k-1} 2^{-2j} g_j(x). \end{aligned} \tag{A.32}$$

Induction thus establishes (b). Next observe that (a) and (b) that for every $m \in \mathbb{N}$, $x \in [0, 1]$ it holds

$$\begin{aligned} [R_\varrho(\varphi_m)](x) &= -2^{-2m+3} r_1^{m-1}(x) + 2^{-2m+4} r_2^{m-1}(x) - 2^{-2m+3} r_3^{m-1}(x) + r_4^{m-1}(x) \\ &= -2^{-2(m-1)} \big(2r_1^{m-1}(x) - 4r_2^{m-1}(x) + 2r_3^{m-1}(x)\big) + x - \sum_{j=1}^{m-2} 2^{-2j} g_j(x) \\ &= x - \left[\sum_{j=1}^{m-2} 2^{-2j} g_j(x)\right] - 2^{-2(m-1)} g_{m-1}(x) = x - \sum_{j=1}^{m-1} 2^{-2j} g_j(x). \end{aligned} \tag{A.33}$$

Combining this with (A.20) establishes that for every $m \in \mathbb{N}$, $x \in [0, 1]$ it holds

$$[R_\varrho(\varphi_m)](x) = f_{m-1}(x). \tag{A.34}$$

This and (A.21) imply that for every $m \in \mathbb{N}$ it holds

$$\sup_{x \in [0,1]} \left| x^2 - [R_\varrho(\varphi_m)](x) \right| \leq 2^{-2m}. \tag{A.35}$$

Furthermore, observe that by construction it holds for every $m \in \mathbb{N}$

$$\mathcal{L}(\varphi_m) = m \quad \text{and} \quad \mathcal{M}(\varphi_m) = \max\{1, 10 + 15(m-2)\} \leq 15m. \tag{A.36}$$

Let $(\sigma_\varepsilon)_{\varepsilon \in (0,\infty)} \subseteq \mathfrak{N}$ be the neural networks which satisfy for $\varepsilon \in (0,1)$

$$\sigma_\varepsilon = \varphi_{\lceil \frac{1}{2} |\log_2(\varepsilon)| \rceil} \tag{A.37}$$

and for every $\varepsilon \in [1,\infty)$ that $\sigma_\varepsilon = \theta$. Observe that for every $\varepsilon \in [1,\infty)$ it holds

$$\sup_{x \in [0,1]} \left| x^2 - [R_\varrho(\sigma_\varepsilon)](x) \right| = \sup_{x \in [0,1]} \left| x^2 - [R_\varrho(\theta)](x) \right| \leq 1 \leq \varepsilon. \tag{A.38}$$

In addition note for every $\varepsilon \in (0,1)$ it holds

$$\begin{aligned} \sup_{x \in [0,1]} \left| x^2 - [R_\varrho(\sigma_\varepsilon)](x) \right| &= \sup_{x \in [0,1]} \left| x^2 - [R_\varrho(\varphi_{\lceil \frac{1}{2} |\log_2(\varepsilon)| \rceil})](x) \right| \\ &\leq 2^{-2\lceil \frac{1}{2} |\log_2(\varepsilon)| \rceil} \leq 2^{-2(\frac{1}{2} |\log_2(\varepsilon)|)} = 2^{\log_2(\varepsilon)} = \varepsilon. \end{aligned} \tag{A.39}$$

Moreover, observe that (A.36) implies for every $\varepsilon \in (0,1)$ it holds

$$\mathcal{L}(\sigma_\varepsilon) = \mathcal{L}(\varphi_{\lceil \frac{1}{2} |\log_2(\varepsilon)| \rceil}) = \left\lceil \tfrac{1}{2} \left|\log_2(\varepsilon)\right| \right\rceil \tag{A.40}$$

and

$$\mathcal{M}(\sigma_\varepsilon) = \mathcal{M}(\varphi_{\lceil \frac{1}{2} |\log_2(\varepsilon)| \rceil}) \leq 15 \left\lceil \tfrac{1}{2} \left|\log_2(\varepsilon)\right| \right\rceil. \tag{A.41}$$

Furthermore, for every $\varepsilon \in [1,\infty)$ it holds $\mathcal{L}(\sigma_\varepsilon) = \mathcal{L}(\theta) = 1$ and $\mathcal{M}(\sigma_\varepsilon) = \mathcal{M}(\theta) = 0$. This completes the proof of Lemma 6.1. □

A.3 Proof of Lemma 6.2

Proof of Lemma 6.2. Throughout this proof assume Setting 5.2, let $\theta \in \mathcal{N}_1^{1,1}$ be the neural network given by $\theta = (0,0)$, let $\alpha \in \mathcal{N}_2^{2,6,3}$ be the neural network given by

$$\alpha_1 = \left(\left(\begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right), \left(\tfrac{1}{2B} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right) \right), \tag{A.42}$$

and let $\Sigma \in \mathcal{N}_1^{3,1}$ be the neural network given by $\Sigma = \left(\left(\begin{pmatrix} 2B^2 & -2B^2 & -2B^2 \end{pmatrix}, 0 \right) \right)$. Observe that Lemma 6.1 ensures the existence of neural networks $(\sigma_\varepsilon)_{\varepsilon \in (0,\infty)} \subseteq \mathfrak{N}$ which satisfy Lemma 6.1, (i) – (iv). Let $(\mu_\varepsilon)_{\varepsilon \in (0,\infty)} \subseteq \mathfrak{N}$ be the neural networks which satisfy for every $\varepsilon \in (0,\infty)$

$$\mu_\varepsilon = \begin{cases} \Sigma \odot \mathcal{P}\left(\sigma_{\varepsilon/6B^2}, \sigma_{\varepsilon/6B^2}, \sigma_{\varepsilon/6B^2}\right) \odot \alpha & : \varepsilon < B^2 \\ \theta & : \varepsilon \geq B^2 \end{cases}. \tag{A.43}$$

Note first that for every $\varepsilon \in [B^2, \infty)$ it holds

$$\sup_{x,y \in [-B,B]} |xy - [R_\varrho(\mu_\varepsilon)](x,y)| = \sup_{x,y \in [-B,B]} |xy - [R_\varrho(\theta)](x,y)| = \sup_{x,y \in [-B,B]} |xy - 0| = B^2 \leq \varepsilon. \tag{A.44}$$

Next observe that for every $(x,y) \in \mathbb{R}^2$ it holds

$$[R_\varrho(\alpha)](x,y) = \tfrac{1}{2B} \begin{pmatrix} \varrho(x+y) + \varrho(-(x+y)) \\ \varrho(x) + \varrho(-x) \\ \varrho(y) + \varrho(-y) \end{pmatrix} = \tfrac{1}{2B} \begin{pmatrix} |x+y| \\ |x| \\ |y| \end{pmatrix}. \tag{A.45}$$

Furthermore, for every $(x, y, z) \in \mathbb{R}^3$ holds $[R_\varrho(\Sigma)](x, y, z) = 2B^2x - 2B^2y - 2B^2z$. Combining this with Lemma 5.3, Lemma 5.4, (A.43), and (A.45) establishes that for every $\varepsilon \in (0, B^2)$, $(x, y) \in [-B, B]^2$ it holds

$$[R_\varrho(\mu_\varepsilon)](x, y) = 2B^2 \left([R_\varrho(\sigma_{\varepsilon/6B^2})] \left(\tfrac{|x+y|}{2B} \right) - [R_\varrho(\sigma_{\varepsilon/6B^2})] \left(\tfrac{|x|}{2B} \right) - [R_\varrho(\sigma_{\varepsilon/6B^2})] \left(\tfrac{|y|}{2B} \right) \right). \tag{A.46}$$

With Lemma 6.1, Item iv, (A.46) establishes (v). In addition note that Lemma 6.1 demonstrates for every $\varepsilon \in (0, \infty)$ it holds

$$\begin{aligned}
&\sup_{z \in [-2B, 2B]} \left| \tfrac{1}{2} z^2 - 2B^2 \left[[R_\varrho(\sigma_{\varepsilon/6B^2})] \left(\tfrac{|z|}{2B} \right) \right] \right| \\
&= \sup_{z \in [-2B, 2B]} \left| 2B^2 \left[\tfrac{|z|}{2B} \right]^2 - 2B^2 \left[[R_\varrho(\sigma_{\varepsilon/6B^2})] \left(\tfrac{|z|}{2B} \right) \right] \right| \\
&= 2B^2 \left[\sup_{t \in [0,1]} \left| t^2 - \left[[R_\varrho(\sigma_{\varepsilon/6B^2})] \, (t) \right] \right| \right] \le 2B^2 \left[\frac{\varepsilon}{6B^2} \right] = \frac{\varepsilon}{3}.
\end{aligned} \tag{A.47}$$

This and (A.46) establish that for every $\varepsilon \in (0, B^2)$ it holds

$$\begin{aligned}
&\sup_{x, y \in [-B, B]} |xy - [R_\varrho(\mu_\varepsilon)](x, y)| \\
&= \sup_{x, y \in [-B, B]} \left| \frac{1}{2} \left[(x + y)^2 - x^2 - y^2 \right] - [R_\varrho(\mu_\varepsilon)](x, y) \right| \\
&\le \tfrac{\varepsilon}{3} + \tfrac{\varepsilon}{3} + \tfrac{\varepsilon}{3} = \varepsilon.
\end{aligned} \tag{A.48}$$

Next observe that $\mathcal{L}(\alpha) = 2$ and $\mathcal{L}(\Sigma) = 1$. Combining this with Lemma 5.3, Lemma 5.4, and Lemma 6.1(i) ensures for every $\varepsilon \in (0, B^2)$

$$\begin{aligned}
\mathcal{L}(\mu_\varepsilon) &= \mathcal{L}(\Sigma) + \mathcal{L}(\sigma_{\varepsilon/6B^2}) + \mathcal{L}(\alpha) \\
&\le \tfrac{1}{2} \left| \log_2(\tfrac{\varepsilon}{6B^2}) \right| + 4 = \tfrac{1}{2} \log_2(\tfrac{6B^2}{\varepsilon}) + 4 \\
&\le \tfrac{1}{2} (\log_2(\tfrac{1}{\varepsilon}) + 2 \log_2(B) + 3) + 4 \\
&= \tfrac{1}{2} \log_2(\tfrac{1}{\varepsilon}) + \log_2(B) + 6.
\end{aligned} \tag{A.49}$$

Combining $\mathcal{M}(\alpha) = 14$ and $\mathcal{M}(\Sigma) = 3$ with Lemma 5.3, Lemma 5.4, Lemma 6.1(ii), and (A.42) demonstrate that for every $\varepsilon \in (0, B^2)$ it holds

$$\begin{aligned}
\mathcal{M}(\mu_\varepsilon) &\le 2 \left(\mathcal{M}(\Sigma) + 3\mathcal{M}(\sigma_{\varepsilon/6B^2}) + \mathcal{M}(\alpha) \right) \\
&\le 34 + 90(\tfrac{1}{2} | \log_2(\tfrac{6B^2}{\varepsilon})| + 1) \\
&\le 45 \log_2(\tfrac{1}{\varepsilon}) + 90 \log_2(B) + 259.
\end{aligned} \tag{A.50}$$

Moreover, for every $\varepsilon \in (B^2, \infty)$ it holds $\mathcal{L}(\mu_\varepsilon) = 1$ and $\mathcal{M}(\mu_\varepsilon) = 0$. Next, observe Lemma 5.3 and Lemma 5.4 demonstrate that for every $\varepsilon \in (0, \infty)$ it holds that $\mathcal{M}_1(\mu_\varepsilon) = \mathcal{M}_1(\alpha) = 8$ and $\mathcal{M}_{\mathcal{L}(\mu_\varepsilon)}(\mu_\varepsilon) = \mathcal{M}(\Sigma) = 3$. This completes the proof of Lemma 6.2.

□

A.4 Proof of Theorem 6.5

Proof of Theorem 6.5. Throughout this proof assume Setting 5.2, let $h_{N,j} : \mathbb{R} \to \mathbb{R}$, $N \in \mathbb{N}$, $j \in \{0, 1, \ldots, N\}$, be the functions which satisfy for every $N \in \mathbb{N}$, $j \in \{0, 1, \ldots, N\}$, $x \in \mathbb{R}$

$$h_{N,j}(x) = \begin{cases} Nx + 1 - j & : \frac{j-1}{N} \le x \le \frac{j}{N} \\ -Nx + 1 + j & : \frac{j}{N} \le x \le \frac{j+1}{N} \\ 0 & : \text{else} \end{cases}, \tag{A.51}$$

let $T_{f,N,j}\colon \mathbb{R} \to \mathbb{R}$, $f \in B_1^n$, $N \in \mathbb{N}$, $j \in \{0,1,\dots,N\}$, be the functions which satisfy for every $f \in B_1^n$, $N \in \mathbb{N}$, $j \in \{0,1,\dots,N\}$, $x \in [0,1]$

$$T_{f,N,j}(x) = \sum_{k=0}^{n-1} \frac{f^{(k)}(\frac{j}{N})}{k!}(x - \tfrac{j}{N})^k. \tag{A.52}$$

For every $f \in B_1^n$, let $f_N : \mathbb{R} \to \mathbb{R}$, $N \in \mathbb{N}$ denote functions which satisfy for every $N \in \mathbb{N}$, $x \in [0,1]$

$$f_N(x) = \sum_{j=0}^{N} h_{N,j}(x) T_{f,N,j}(x). \tag{A.53}$$

Observe that Taylor's theorem (with Lagrange remainder term) ensures that for every $f \in B_1^n$, $N \in \mathbb{N}$, $j \in \{0,1,\dots,N\}$, $x \in [\max\{0, \frac{j-1}{N}\}, \min\{1, \frac{j+1}{N}\}]$

$$\begin{aligned} |f(x) - T_{f,N,j}(x)| &\leq \tfrac{1}{n!} \left|x - \tfrac{j}{N}\right|^n \sup_{\xi \in [\max\{0, \frac{j-1}{N}\}, \min\{1, \frac{j+1}{N}\}]} \left|f^{(n)}(\xi)\right| \\ &\leq \tfrac{1}{n!} N^{-n} \max_{k \in \{0,1,\dots,n\}} \left[\sup_{t \in [0,1]} \left|f^{(k)}(t)\right|\right] \leq \tfrac{1}{n!} N^{-n}. \end{aligned} \tag{A.54}$$

Moreover, for every $N \in \mathbb{N}$, $x \in [0,1]$, $j \notin \{\lceil Nx \rceil - 1, \lceil Nx \rceil\}$ it holds that $h_{N,j}(x) = 0$. We obtain for every $N \in \mathbb{N}$ and $x \in [0,1]$

$$\sum_{j=0}^{N} h_{N,j}(x) T_{f,N,j}(x) = h_{N,\lceil Nx \rceil - 1}(x) T_{f,N,\lceil Nx \rceil - 1}(x) + h_{N,\lceil Nx \rceil}(x) T_{f,N,\lceil Nx \rceil}(x). \tag{A.55}$$

Furthermore, (A.51) implies for every $N \in \mathbb{N}$, $j \in \{1,\dots,N-1\}$, $x \in [\frac{j-1}{N}, \frac{j}{N}]$ holds

$$h_{N,j-1}(x) + h_{N,j}(x) = -Nx + 1 + (j-1) + Nx + 1 - j = 1. \tag{A.56}$$

Combining this with (A.53), (A.54), and (A.55) establishes that for every $f \in B_1^n$, $N \in \mathbb{N}$, $x \in [0,1]$

$$\begin{aligned} &|f(x) - f_N(x)| \\ &= \left|f(x) - \sum_{j=0}^{N} h_{N,j}(x) T_{f,N,j}(x)\right| \\ &= \left|f(x) - \big(h_{N,\lceil Nx \rceil - 1}(x) T_{f,N,\lceil Nx \rceil - 1}(x) + h_{N,\lceil Nx \rceil}(x) T_{f,N,\lceil Nx \rceil}(x)\big)\right| \\ &\leq \left|h_{N,\lceil Nx \rceil - 1}(x) f(x) - h_{N,\lceil Nx \rceil - 1}(x) T_{f,N,\lceil Nx \rceil - 1}(x)\right| \\ &\quad + \left|h_{N,\lceil Nx \rceil}(x) f(x) - h_{N,\lceil Nx \rceil}(x) T_{f,N,\lceil Nx \rceil}(x)\right| \\ &= h_{N,\lceil Nx \rceil - 1}(x) \left|f(x) - T_{f,N,\lceil Nx \rceil - 1}(x)\right| + h_{N,\lceil Nx \rceil}(x) \left|f(x) - T_{f,N,\lceil Nx \rceil}(x)\right| \\ &\leq h_{N,\lceil Nx \rceil - 1}(x) \left[\tfrac{1}{n!} N^{-n}\right] + h_{N,\lceil Nx \rceil}(x) \left[\tfrac{1}{n!} N^{-n}\right] = \tfrac{1}{n!} N^{-n}. \end{aligned} \tag{A.57}$$

We now realize this local Taylor approximation using neural networks. To this end, note that Theorem 6.3 ensures that there exist $C \in \mathbb{R}$ and neural networks $(\Pi^k_\eta)_{\eta \in (0,\infty)}$, $k \in \mathbb{N} \cap [2,\infty)$ which satisfy

(A) $\mathcal{L}(\Pi^k_\eta) \leq C \ln(k) \left(|\ln(\eta)| + k \ln(3) + \ln(k)\right)$,

(B) $\mathcal{M}(\Pi^k_\eta) \leq Ck \left(|\ln(\eta)| + k \ln(3) + \ln(k)\right)$,

(C) $\displaystyle\sup_{x \in [-3,3]^k} \left| \left[\prod_{i=1}^{k} x_i\right] - \left[R_\varrho(\Pi^k_\eta)\right](x) \right| \leq \eta$ and

(D) $R_\varrho \left[\Pi^k_\eta\right](x_1, x_2, \dots, x_k) = 0$, if there exists $i \in \{1,2,\dots,k\}$ with $x_i = 0$.

To complete the proof, we introduce the following neural networks:

- $\nabla_{N,j,k} \in \mathcal{N}_1^{k,1}$, $N \in \mathbb{N}$, $j \in \{0, 1, \ldots, N\}$, $k \in \{2, 3, \ldots, n-1\}$ given by

$$\nabla_{N,j,k} = \left(\left(\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \begin{pmatrix} -\frac{j}{N} \\ \vdots \\ -\frac{j}{N} \end{pmatrix} \right) \right), \tag{A.58}$$

- $\xi^k_{\varepsilon,N,j} \in \mathfrak{N}$, $\varepsilon \in (0,\infty)$, $N \in \mathbb{N}$, $j \in \{0, 1, \ldots, N\}$, $k \in \{1, 2, \ldots, n-1\}$, given by

$$\xi^k_{\varepsilon,N,j} = \begin{cases} (1, 0) & : k = 1 \\ \Pi^k_{\varepsilon/8e} \odot \nabla_{N,j,k} & : k > 1 \end{cases}, \tag{A.59}$$

- $\Sigma_{f,N,j} \in \mathcal{N}_1^{1,n-1}$, $f \in B_1^n$, $N \in \mathbb{N}$, $j \in \{0, 1, \ldots, N\}$ given by

$$\Sigma_{f,N,j} = \left(\left(\begin{pmatrix} \frac{f^{(n-1)}(\frac{j}{N})}{(n-1)!} & \frac{f^{(n-2)}(\frac{j}{N})}{(n-2)!} & \ldots & \frac{f^{(1)}(\frac{j}{N})}{(1)!} \end{pmatrix}, f(\tfrac{j}{N}) \right) \right), \tag{A.60}$$

- $\tau_{f,\varepsilon,N,j} \in \mathfrak{N}$, $f \in B_1^n$, $\varepsilon \in (0,\infty)$, $N \in \mathbb{N}$, $j \in \{0, 1, \ldots, N\}$ given by

$$\tau_{f,\varepsilon,N,j} = \Sigma_{f,N,j} \odot \mathcal{P}(\xi^{n-1}_{\varepsilon,N,j}, \xi^{n-2}_{\varepsilon,N,j}, \ldots, \xi^1_{\varepsilon,N,j}) \odot \nabla_{1,0,n-1}, \tag{A.61}$$

- $\chi_{N,j} \in \mathcal{N}_2^{1,3,1}$, $N \in \mathbb{N}$, $j \in \{0, 1, \ldots, N\}$ given by

$$\chi_{N,j} = \left(\left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -{}^{(j-1)}/_{N} \\ -{}^{j}/_{N} \\ -{}^{(j+1)}/_{N} \end{pmatrix} \right), \left(\begin{pmatrix} 1 & -2 & 1 \end{pmatrix}, 0 \right) \right) \tag{A.62}$$

- $\lambda_N \in \mathcal{N}_1^{1,N+1}$, $N \in \mathbb{N}$ given by

$$\lambda_N = \left(\left(\begin{pmatrix} 1 & \ldots & 1 \end{pmatrix}, 0 \right) \right), \tag{A.63}$$

- $\psi_{f,\varepsilon,N,j} \in \mathfrak{N}$, $f \in B_1^n$, $\varepsilon \in (0,\infty)$, $N \in \mathbb{N}$, $j \in \{0, 1, \ldots, N\}$ given by

$$\psi_{f,\varepsilon,N,j} = \Pi^2_{\varepsilon/8} \odot \mathcal{P}(\chi_{N,j}, \tau_{f,\varepsilon,N,j}), \tag{A.64}$$

- $\varphi_{f,\varepsilon,N} \in \mathfrak{N}$, $f \in B_1^n$, $N \in \mathbb{N}$, $\varepsilon \in (0,\infty)$ given by

$$\varphi_{f,\varepsilon,N} = \lambda_N \odot \mathcal{P}\left(\psi_{f,\varepsilon,N,1}, \psi_{f,\varepsilon,N,2}, \ldots, \psi_{f,\varepsilon,N,N}\right) \odot \nabla_{1,0,2N+2}. \tag{A.65}$$

With these networks, we note Lemma 5.3, Lemma 5.4, (C), (A.58) and (A.59) ensure that for every $N \in \mathbb{N}$, $\varepsilon \in (0,\infty)$, $j \in \{0, 1, \ldots, N\}$, $k \in \{2, 3, \ldots, n-1\}$

$$\begin{aligned} &\sup_{x \in [0,1]} \left| (x - \tfrac{j}{N})^k - \left[R_\varrho(\xi^k_{\varepsilon,N,j}) \right](x) \right| \\ &\leq \sup_{x \in [0,1]} \left| (x - \tfrac{j}{N})^k - \left[R_\varrho(\Pi^k_{\varepsilon/8e}) \right] \left(\left[R_\varrho(\nabla_{N,j,k}) \right](x) \right) \right| \\ &\leq \sup_{x \in [0,1]} \left| \left[\prod_{i=1}^k (x - \tfrac{j}{N})^k \right] - \left[R_\varrho(\Pi^k_{\varepsilon/8e}) \right] (x - \tfrac{j}{N}, x - \tfrac{j}{N}, \ldots, x - \tfrac{j}{N}) \right| \\ &\leq \sup_{x \in [-1,1]^k} \left| \left[\prod_{i=1}^k x_i \right] - \left[R_\varrho(\Pi^k_{\varepsilon/8e}) \right](x) \right| \leq \tfrac{\varepsilon}{8e} \end{aligned} \tag{A.66}$$

and

$$\sup_{x\in[0,1]} \left|(x-\tfrac{j}{N}) - \left[R_\varrho(\xi^1_{\varepsilon,N,j})\right](x)\right| = 0. \tag{A.67}$$

Moreover, Lemma 5.3, Lemma 5.4, (A.58), (A.59), (A.60), and (A.61) demonstrate that for every $f \in B^n_1$, $N \in \mathbb{N}$, $\varepsilon \in (0,\infty)$, $j \in \{0,1,\dots,N\}$, $x \in [0,1]$ it holds

$$\left[R_\varrho(\tau_{f,\varepsilon,N,j})\right](x) = \sum_{k=1}^{n-1}\left[\frac{f^{(k)}(\frac{j}{N})}{k!}\left[R_\varrho(\xi^k_{\varepsilon,N,j})\right](x)\right] + f(\tfrac{j}{N}). \tag{A.68}$$

Combining this with (A.52), (A.61), (A.66) and (A.66) establishes that for every $f \in B^n_1$, $N \in \mathbb{N}$, $\varepsilon \in (0,\infty)$, $j \in \{0,1,\dots,N\}$, $x \in [0,1]$ it holds

$$\begin{aligned}
&|T_{f,N,j}(x) - \left[R_\varrho(\tau_{f,\varepsilon,N,j})\right](x)| \\
&= \left|\left(\sum_{k=0}^{n-1}\frac{f^{(k)}(\frac{j}{N})}{k!}(x-\tfrac{j}{N})^k\right) - \left(\sum_{k=1}^{n-1}\left[\frac{f^{(k)}(\frac{j}{N})}{k!}\left[R_\varrho(\xi^k_{\varepsilon,N,j})\right](x)\right] + f(\tfrac{j}{N})\right)\right| \\
&\leq \sum_{k=1}^{n-1}\left(\frac{f^{(k)}(\frac{j}{N})}{k!}\left|(x-\tfrac{j}{N})^k - \left[R_\varrho(\xi^k_{\varepsilon,N,j})\right](x)\right|\right) \\
&\leq \frac{\varepsilon}{8e}\sum_{k=1}^{n-1}\frac{f^{(k)}(\frac{j}{N})}{k!} \leq \frac{\varepsilon}{8e}\left(\sum_{k=1}^{\infty}\frac{1}{k!}\right) \leq \frac{\varepsilon}{8}.
\end{aligned} \tag{A.69}$$

Next, (A.62) ensures for every $N \in \mathbb{N}$, $j \in \{0,1,\dots,N\}$, $x \in [0,1]$

$$[R_\varrho(\chi_{N,j})](x) = \varrho(x-\tfrac{j-1}{N}) - 2\varrho(x-\tfrac{j}{N}) + \varrho(x-\tfrac{j+1}{N}) = h_{N,j}(x). \tag{A.70}$$

Now (A.69) and Taylor's Theorem imply for every $f \in B^n_1$, $N \in \mathbb{N}$, $\varepsilon \in (0,1)$, $j \in \{0,1,\dots,N\}$, $x \in [0,1]$ that

$$\begin{aligned}
|[R_\varrho(\tau_{f,\varepsilon,N,j})](x)| &\leq |[R_\varrho(\tau_{f,\varepsilon,N,j})](x) - T_{f,N,j}(x)| + |T_{f,N,j}(x) - f(x)| + |f(x)| \\
&\leq \frac{\varepsilon}{4(N+1)} + \tfrac{1}{n!}x^n \sup_{t\in[0,1]}|f^{(n)}(t)| + \sup_{t\in[0,1]}|f(t)| \leq 3.
\end{aligned} \tag{A.71}$$

Combining this with Lemma 5.3, Lemma 5.4, (A.51), (C), (A.69), and (A.70) establishes for every $f \in B^n_1$, $N \in \mathbb{N}$, $\varepsilon \in (0,1)$, $j \in \{0,1,\dots,N\}$, $x \in [0,1]$ the bound

$$\begin{aligned}
&|h_{N,j}(x)T_{f,N,j}(x) - [R_\varrho(\psi_{f,\varepsilon,N,j})](x,x)| \\
&\leq |h_{N,j}(x)T_{f,N,j}(x) - [R_\varrho(\chi_{N,j})](x)[R_\varrho(\tau_{N,j})](x)| \\
&\quad + \left|[R_\varrho(\chi_{N,j})](x)[R_\varrho(\tau_{N,j})](x) - [R_\varrho(\Pi^2_{\varepsilon/8} \circ \mathcal{P}(\chi_{N,j}, \tau_{f,\varepsilon,N,j}))](x,x)\right| \\
&\leq |h_{N,j}(x)T_{f,N,j}(x) - [R_\varrho(\tau_{N,j})](x)| \\
&\quad + \left|[R_\varrho(\chi_{N,j})](x)[R_\varrho(\tau_{N,j})](x) - [R_\varrho(\Pi^2_{\varepsilon/8})]([R_\varrho(\chi_{N,j}](x), [R_\varrho(\tau_{f,\varepsilon,N,j})](x))\right| \\
&\leq \tfrac{\varepsilon}{8} + \tfrac{\varepsilon}{8} = \tfrac{\varepsilon}{4}.
\end{aligned} \tag{A.72}$$

Furthermore, note that for every $N \in \mathbb{N}$, $j \in \{0,1,\dots,N\}$, $x \notin [\frac{j-1}{N}, \frac{j+1}{N}]$ it holds that $h_{N,j}(x) = \chi_{N,j}(x) = 0$. Thus (D) ensures that for every $f \in B^n_1$, $N \in \mathbb{N}$, $\varepsilon \in (0,1)$, $j \in \{0,1,\dots,N\}$, $x \in [0,1]$ it holds

$$|h_{N,j}(x)T_{f,N,j}(x) - [R_\varrho(\psi_{f,\varepsilon,N,j})](x,x)| = 0. \tag{A.73}$$

This, Lemma 5.3, Lemma 5.4, (A.53), (A.65), and (A.72) imply that for every $f \in B^n_1$, $N \in \mathbb{N}$, $\varepsilon \in (0,1)$, $x \in [0,1]$ it holds

$$\begin{aligned}
|f_N(x) - [R_\varrho(\varphi_{f,\varepsilon,N})](x)| &= \left|\sum_{j=0}^{N} h_{N,j}(x)T_{f,N,j}(x) - \sum_{j=0}^{N}[R_\varrho(\psi_{f,\varepsilon,N,j})](x,x)\right| \\
&\leq 2\max_{j\in\{0,1,\dots,N\}}|h_{N,j}(x)T_{f,N,j}(x) - [R_\varrho(\psi_{f,\varepsilon,N,j})](x,x)| \\
&\leq \tfrac{\varepsilon}{2}.
\end{aligned} \tag{A.74}$$

Combining this with (A.57) establishes that for every $f \in B_1^n$, $N \in \mathbb{N}$, $\varepsilon \in (0,1)$, $x \in [0,1]$ it holds

$$|f(x) - [R_\varrho(\varphi_{f,\varepsilon,N})](x)| \leq |f(x) - f_N(x)| + |f_N(x) - [R_\varrho(\varphi_{f,\varepsilon,N})]| \leq \tfrac{1}{n!}N^{-n} + \tfrac{\varepsilon}{2}. \tag{A.75}$$

Let $N_\varepsilon \in \mathbb{N}$ satisfy for every $\varepsilon \in (0,\infty)$

$$N_\varepsilon = \left\lceil \left[\tfrac{2}{n!\varepsilon}\right]^{1/n} \right\rceil, \tag{A.76}$$

let $\theta \in \mathcal{N}_1^{1,1}$ be given by $\theta = (0,0)$, and let $(\Phi_{f,\varepsilon})_{f\in B_1^n, \varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ be the neural networks given by

$$\Phi_{f,\varepsilon} = \begin{cases} \varphi_{f,\varepsilon,N_\varepsilon} & : \varepsilon < 1 \\ \theta & : \varepsilon \geq 1 \end{cases}. \tag{A.77}$$

Oberve that (A.75) implies that for every $f \in B_1^n$, $\varepsilon \in (0,1)$, $x \in [0,1]$

$$|f(x) - [R_\varrho(\Phi_{f,\varepsilon})](x)| = |f(x) - [R_\varrho(\varphi_{f,\varepsilon,N_\varepsilon})](x)| \leq \tfrac{1}{n!}N_\varepsilon^{-n} + \tfrac{\varepsilon}{2} \leq \tfrac{1}{n!}\left[\tfrac{n!\varepsilon}{2}\right] + \tfrac{\varepsilon}{2} = \varepsilon. \tag{A.78}$$

Moreover that for every $f \in B_1^n$, $\varepsilon \in [1,\infty)$, $x \in [0,1]$ it holds

$$|f(x) - [R_\varrho(\Phi_{f,\varepsilon})](x)| = |f(x) - [R_\varrho(\theta)](x)| = |f(x)| \leq 1 \leq \varepsilon. \tag{A.79}$$

This and (A.78) establish that the neural networks $(\Phi_{f,\varepsilon})_{f\in B_1^n,\varepsilon\in(0,\infty)}$ satisfy (iii).

Next, Lemma 5.3, Lemma 5.4, (A), (A.58), and (A.59) imply for every $N \in \mathbb{N}$, $\varepsilon \in (0,\infty)$, $j \in \{0,1,\dots,N\}$, $k \in \{1,2,\dots,n-1\}$

$$\mathcal{L}(\xi^k_{\varepsilon,N,j}) \leq \max\{1, \mathcal{L}(\Pi^k_{\varepsilon/8e}) + \mathcal{L}(\nabla_{N,j,k})\} \leq C\ln(k)\left(|\ln(\tfrac{\varepsilon}{8e})| + k\ln(3) + \ln(k)\right) + 1. \tag{A.80}$$

Combining this with Lemma 5.3, Lemma 5.4, (A.58), (A.60), (A.61) shows for every $f \in B_1^n$, $N \in \mathbb{N}$, $\varepsilon \in (0,\infty)$, $j \in \{0,1,\dots,N\}$ the bound

$$\begin{aligned} \mathcal{L}(\tau_{f,\varepsilon,N,j}) &\leq \mathcal{L}(\Sigma_{f,N,j}) + \left[\max_{k\in\{1,2,\dots,n-1\}} \mathcal{L}(\xi^k_{\varepsilon,N,j})\right] + \mathcal{L}(\nabla_{1,0,n-1}) \\ &\leq 3 + C\ln(n)\left(|\ln(\tfrac{\varepsilon}{8e})| + n\ln(3) + \ln(n)\right). \end{aligned} \tag{A.81}$$

This, Lemma 5.3, Lemma 5.4, (A), (A.62), (A.63), (A.65), and (A.58) ensure for every $f \in B_1^n$, $N \in \mathbb{N}$, $\varepsilon \in (0,\infty)$ it holds

$$\begin{aligned} \mathcal{L}(\varphi_{f,\varepsilon,N}) &\leq \mathcal{L}(\lambda_N) + \left[\max_{j\in\{0,1,\dots,N\}} \mathcal{L}(\psi_{f,\varepsilon,N,j})\right] + \mathcal{L}(\nabla_{1,0,2N+2}) \\ &\leq 2 + \left[\max_{j\in\{0,1,\dots,N\}} \mathcal{L}(\Pi^2_{\varepsilon/8} \odot \mathcal{P}(\chi_{N,j}, \tau_{f,\varepsilon,N,j}))\right] \\ &\leq 2 + \left[C\ln(2)\left(|\ln(\tfrac{\varepsilon}{8})| + 2\ln(3) + \ln(2)\right) + \max\{3, \mathcal{L}(\tau_{f,\varepsilon,N,j})\}\right] \\ &\leq 5 + C\ln(2)\left(|\ln(\tfrac{\varepsilon}{8})| + \ln(18)\right) + C\ln(n)\left(|\ln(\tfrac{\varepsilon}{8e})| + n\ln(3) + \ln(n)\right) \\ &\leq 5 + C\ln(2)\left(|\ln(\varepsilon)| + |\ln(8)| + \ln(18)\right) \\ &\quad + C\ln(n)\left(|\ln(\varepsilon)| + |\ln(8e)| + n\ln(3) + \ln(n)\right) \\ &= C\ln(2n)\,|\ln(\varepsilon)| + C(\ln(2)\ln(144) + \ln(n)(\ln(3)n + \ln(n) + |\ln(8e)|)) + 5. \end{aligned} \tag{A.82}$$

With the constant C from (A.82), define the term T_1 by

$$T_1 = C(\ln(2)\ln(144) + \ln(n)(\ln(3)n + \ln(n) + |\ln(8e)|)) + 5. \tag{A.83}$$

Observe that (A.82) implies for every $f \in B_1^n$, $\varepsilon \in (0,1)$

$$\mathcal{L}(\Phi_{f,\varepsilon}) = \mathcal{L}(\varphi_{f,\varepsilon,N_\varepsilon}) = C\ln(2n)\,|\ln(\varepsilon)| + T_1. \tag{A.84}$$

Hence we obtain

$$\sup_{f\in B_1^n,\varepsilon\in(0,e^{-r}]}\left[\tfrac{\mathcal{L}(\Phi_{f,\varepsilon})}{\max\{r,|\ln(\varepsilon)|\}}\right]\le\sup_{f\in B_1^n,\varepsilon\in(0,e^{-r}]}\left[\tfrac{C\ln(2n)|\ln(\varepsilon)|+T_1}{|\ln(\varepsilon)|}\right]\le C\ln(2n)+\tfrac{T_1}{r}<\infty. \tag{A.85}$$

In addition, note that (A.84) ensures that

$$\sup_{f\in B_1^n,\varepsilon\in(e^{-r},1)}\left[\tfrac{\mathcal{L}(\Phi_{f,\varepsilon})}{\max\{r,|\ln(\varepsilon)|\}}\right]\le\sup_{f\in B_1^n,\varepsilon\in(e^{-r},1)}\left[\tfrac{C\ln(2n)|\ln(\varepsilon)|+T_1}{r}\right]\le C\ln(2n)+\tfrac{T_1}{r}<\infty. \tag{A.86}$$

Furthermore

$$\sup_{f\in B_1^n,\varepsilon\in[1,\infty)}\left[\tfrac{\mathcal{L}(\Phi_{f,\varepsilon})}{\max\{r,|\ln(\varepsilon)|\}}\right]=\sup_{f\in B_1^n,\varepsilon\in[1,\infty)}\left[\tfrac{1}{\max\{r,|\ln(\varepsilon)|\}}\right]<\infty. \tag{A.87}$$

This, (A.85), and (A.86) establish that the neural networks $(\Phi_{f,\varepsilon})_{\varepsilon\in(0,\infty)}$ satisfy (i). Next, Lemma 5.3, (B), (A.58), and (A.59) imply for every $N\in\mathbb{N}$, $\varepsilon\in(0,\infty)$, $j\in\{0,1,\dots,N\}$, $k\in\{1,2,\dots,n-1\}$

$$\mathcal{M}(\xi^k_{\varepsilon,N,j})\le\max\{1,2(\mathcal{M}(\Pi^k_{\varepsilon/8e})+\mathcal{M}(\nabla_{N,j,k}))\}\le 2(Ck\left(\left|\ln(\tfrac{\varepsilon}{8e})\right|+k\ln(3)+\ln(k)\right)+1) \tag{A.88}$$

Combining this with Lemma 5.3, Lemma 5.4, (A.58), (A.60), and (A.61) shows for every $f\in B_1^n$, $N\in\mathbb{N}$, $\varepsilon\in(0,\infty)$, $j\in\{0,1,\dots,N\}$ it holds

$$\begin{aligned}\mathcal{M}(\tau_{f,\varepsilon,N,j})&\le 2\left(\mathcal{M}(\Sigma_{f,N,j})+2\left(\mathcal{M}(\mathcal{P}(\xi^{n-1}_{\varepsilon,N,j},\dots,\xi^1_{\varepsilon,N,j}))+\mathcal{L}(\nabla_{1,0,n-1})\right)\right)\\&\le 2n+4\left(2\left[\sum_{k=1}^{n-1}\mathcal{M}(\xi^k_{\varepsilon,N,j})\right]+4(n-1)\max_{k\in\{1,2,\dots,n-1\}}\mathcal{L}(\xi^k_{\varepsilon,N,j})\right)+8(n-1)\\&\le 10n+8(n-1)(2Cn\left(\ln(\tfrac{\varepsilon}{(8e)})|+n\ln(3)+\ln(n)\right)+2)\\&\quad+16(n-1)(C\ln(n)\left(|\ln(\tfrac{\varepsilon}{8e})|+n\ln(3)+\ln(n)\right)+1)\\&\le 32n^2C\left(|\ln(\tfrac{\varepsilon}{8e})|+n\ln(3)+\ln(n)\right)+42n.\end{aligned} \tag{A.89}$$

Let the term T_2 be given by

$$T_2=128\left(C+32n^2C+C\ln(n)\right), \tag{A.90}$$

and let the term T_3 be given by

$$T_3=1556+128(C\ln(144)+64n^2C(n\ln(3)+\ln(n))+42n. \tag{A.91}$$

This, Lemma 5.3, Lemma 5.4, (B), (A.58), (A.62), (A.63), (A.65), and the fact that for every $\psi\in\mathfrak{N}$ with $\min_{l\in\{1,2,\dots,\mathcal{L}(\psi)\}}\mathcal{M}_l(\psi)>0$ it holds that $\mathcal{L}(\psi)\le\mathcal{M}(\psi)$ ensure that for every $f\in B_1^n$, $N\in\mathbb{N}$, $\varepsilon\in(0,\infty)$ it holds

$$\begin{aligned}
&\mathcal{M}(\varphi_{f,\varepsilon,N}) \\
&\leq 2\left(\mathcal{M}(\lambda_N) + 2\left[\mathcal{M}(\mathcal{P}(\psi_{f,\varepsilon,N,1}, \psi_{f,\varepsilon,N,2}, \dots, \psi_{f,\varepsilon,N,N})) + \mathcal{M}(\nabla_{1,0,2N+2})\right]\right) \\
&\leq 2(N+1) + 8\left[\sum_{j=0}^{N} \mathcal{M}(\psi_{f,\varepsilon,N,j})\right] + 16(N+1)\left[\max_{j\in\{0,1,\dots,N\}} \mathcal{L}(\psi_{f,\varepsilon,N,j})\right] + 8(N+1) \\
&\leq 20N + 32(N+1) \max_{j\in\{1,2,\dots,N\}} \mathcal{M}(\psi_{f,\varepsilon,N,j}) \\
&\leq 20N + 64N\Big(\mathcal{M}(\Pi^2_{\varepsilon/8}) + \mathcal{M}(\mathcal{P}(\chi_{N,N}, \tau_{f,\varepsilon,N,N}))\Big) \\
&\leq 20N + 128NC\left(\left|\ln(\tfrac{\varepsilon}{8})\right| + 2\ln(3) + \ln(2)\right) \\
&\quad + 64N\left(2\mathcal{M}(\chi_{N,N}) + 2\mathcal{M}(\tau_{f,\varepsilon,N,N}) + 4\max\{\mathcal{L}(\chi_{N,N}), \mathcal{L}(\tau_{f,\varepsilon,N,N})\}\right) \\
&\leq 20N + 128NC\left(\left|\ln(\tfrac{\varepsilon}{8})\right| + \ln(18)\right) + 1152N \\
&\quad + 128N\left(32n^2C\left(|\ln(\tfrac{\varepsilon}{8e})| + n\ln(3) + \ln(n)\right) + 42n\right) \\
&\quad + 128N\left(3 + C\ln(n)\left(|\ln(\tfrac{\varepsilon}{8e})| + n\ln(3) + \ln(n)\right)\right) \\
&= 128\left(C + 32n^2C + C\ln(n)\right)N|\ln(\varepsilon)| \\
&\quad + \left(1556 + 128(C\ln(144) + 64n^2C(n\ln(3) + \ln(n)) + 42n\right)N \\
&= T_2N|\ln(\varepsilon)| + T_3N.
\end{aligned} \tag{A.92}$$

Combining this with Lemma A.1 demonstrates that for every $f \in B_1^n$, $\varepsilon \in (0, \exp(-2n^2)]$ it holds

$$\begin{aligned}
\mathcal{M}(\Phi_{f,\varepsilon}) = \mathcal{M}(\varphi_{f,\varepsilon,N_\varepsilon}) &\leq T_2N_\varepsilon|\ln(\varepsilon)| + T_3N_\varepsilon \\
&= T_2\left\lceil\left[\tfrac{2}{n!\varepsilon}\right]^{1/n}\right\rceil|\ln(\varepsilon)| + T_3\left\lceil\left[\tfrac{2}{n!\varepsilon}\right]^{1/n}\right\rceil \\
&\leq 3T_2\varepsilon^{-\frac{1}{n}}|\ln(\varepsilon)| + 3T_3\varepsilon^{-\frac{1}{n}} \\
&\leq 3T_2\varepsilon^{-\frac{1}{n}}\max\{r, |\ln(\varepsilon)|\} + 3T_3\varepsilon^{-\frac{1}{n}}.
\end{aligned} \tag{A.93}$$

Hence we obtain

$$\sup_{f\in B_1^n, \varepsilon\in(0,\exp(-2n^2))}\left[\frac{\mathcal{M}(\Phi_{f,\varepsilon})}{\varepsilon^{-\frac{1}{n}}\max\{r, |\ln(\varepsilon)|\}}\right] \leq 3T_2 + 3T_3\frac{1}{\max\{r, 2n^2\}} < \infty. \tag{A.94}$$

Combining (A.93) with the fact that continuous function are bounded on compact sets ensures

$$\begin{aligned}
&\sup_{f\in B_1^n, \varepsilon\in[\exp(-2n^2),1]}\left[\frac{\mathcal{M}(\Phi_{f,\varepsilon})}{\varepsilon^{-\frac{1}{n}}\max\{r, |\ln(\varepsilon)|\}}\right] \\
\leq &\sup_{f\in B_1^n, \varepsilon\in[\exp(-2n^2),1]}\left[\frac{T_2N(|\ln(\varepsilon)| + |\ln(N)|) + T_3N}{\varepsilon^{-\frac{1}{n}}\max\{r, |\ln(\varepsilon)|\}}\right] < \infty.
\end{aligned} \tag{A.95}$$

In addition note

$$\sup_{f\in B_1^n, \varepsilon\in(1,\infty)}\left[\frac{\mathcal{M}(\Phi_{f,\varepsilon})}{\varepsilon^{-\frac{1}{n}}\max\{r, |\ln(\varepsilon)|\}}\right] = \sup_{f\in B_1^n, \varepsilon\in(1,\infty)}\left[\frac{\mathcal{M}(\theta)}{\varepsilon^{-\frac{1}{n}}\max\{r, |\ln(\varepsilon)|\}}\right] \tag{A.96}$$

$$= \sup_{f\in B_1^n, \varepsilon\in(1,\infty)}\left[\frac{0}{\varepsilon^{-\frac{1}{n}}\max\{r, |\ln(\varepsilon)|\}}\right] = 0 < \infty. \tag{A.97}$$

This, (A.94), and (A.95) establish that the neural networks $(\Phi_{f,\varepsilon})_{f\in B_1^n, \varepsilon\in(0,\infty)}$ satisfy (ii). The proof of Theorem 6.5 is completed. □

A.5 Proof of Corollary 6.6

Proof of Corollary 6.6. Throughout this proof assume Setting 5.2, let $c_{a,b} \in \mathbb{R}$, $[a,b] \subseteq \mathbb{R}_+$, be the real numbers given by $c_{a,b} = \min\{1, (b-a)^{-n}\}$, let $\lambda_{a,b} \in \mathcal{N}_1^{1,1}$, $[a,b] \subseteq \mathbb{R}_+$, be the neural networks given by $\lambda_{a,b} = (\frac{1}{b-a}, -\frac{a}{b-a})$, let $\alpha_f \in \mathcal{N}_1^{1,1}$, $f \in \mathcal{C}^n$ be the neural networks given by $\alpha_f = (\frac{1}{c} \|f\|_{n,\infty}, 0)$, let $L_{a,b} \colon [0,1] \to [a,b]$, $[a,b] \subseteq \mathbb{R}_+$ be the functions which satisfy for every $[a,b] \subseteq \mathbb{R}_+$, $t \in [0,1]$

$$L_{a,b}(t) = (b-a)t + a, \tag{A.98}$$

and for every $f \in \mathcal{C}^n$ let $f_* \in C^n([0,1], \mathbb{R})$ be the function which satisfies for every $t \in [0,1]$

$$f_*(t) = \|f\|_{n,\infty}^{-1} c_{a,b}(f(L_{a,b}(t))). \tag{A.99}$$

We claim that for every $[a,b] \subseteq \mathbb{R}_+$, $f \in C^n([a,b], \mathbb{R})$, $m \in \{1, 2, \ldots, n\}$, $t \in [0,1]$ it holds

$$f_*^{(m)}(t) = \|f\|_{n,\infty}^{-1} c_{a,b}(b-a)^m [f^{(m)}(L_{a,b}(t))]. \tag{A.100}$$

We now prove (A.100) by induction on $m \in \{1, 2, \ldots, n\}$. For the base case $m = 1$, the chain rule implies for every $[a,b] \subseteq \mathbb{R}_+$, $f \in C^n([a,b], \mathbb{R})$, $t \in [0,1]$

$$\begin{aligned} f_*'(t) &= \tfrac{\mathrm{d}}{\mathrm{d}t} \Big[\|f\|_{n,\infty}^{-1} c_{a,b} f(L_{a,b}(t)) \Big] = \|f\|_{n,\infty}^{-1} c_{a,b} \left[f'(L_{a,b}(t)) L_{a,b}'(t) \right] \\ &= \|f\|_{n,\infty}^{-1} c_{a,b} \left[f'(L_{a,b}(t))(b-a) \right] = \|f\|_{n,\infty}^{-1} c_{a,b}(b-a)[f'(L_{a,b}(t))]. \end{aligned} \tag{A.101}$$

This establishes (A.100) in the base case $m = 1$.

For the induction step $\{1, 2, \ldots, n-1\} \ni m \to m+1 \in \{2, 3, \ldots, n\}$ observe that the chain rule ensures for every $[a,b] \subseteq \mathbb{R}_+$, $f \in C^n([a,b], \mathbb{R})$, $m \in \mathbb{N}$, $t \in [0,1]$

$$\begin{aligned} \tfrac{\mathrm{d}}{\mathrm{d}t} \Big[\|f\|_{n,\infty}^{-1} c_{a,b}(b-a)^m [f^{(m)}(L_{a,b}(t))] \Big] &= \|f\|_{n,\infty}^{-1} c_{a,b}(b-a)^m [f^{(m+1)}(L_{a,b}(t)) L_{a,b}'(t)] \\ &= \|f\|_{n,\infty}^{-1} c_{a,b}(b-a)^{m+1} [f^{(m+1)}(L_{a,b}(t))]. \end{aligned} \tag{A.102}$$

Induction thus establishes (A.100).

In addition, for every $[a,b] \subseteq \mathbb{R}_+$, $k \in \{0, 1, \ldots, n\}$

$$c_{a,b}(b-a)^k = \min\{1, (b-a)^{-n}\}(b-a)^k = \min\{(b-a)^k, (b-a)^{-n+k}\} \le 1. \tag{A.103}$$

Combining this with (6.30), (A.98), and (A.100) ensures for every $[a,b] \subseteq \mathbb{R}_+$, $f \in C^n([a,b], \mathbb{R})$

$$\begin{aligned} \max_{k \in \{0,1,\ldots,n\}} \left[\sup_{t \in [0,1]} \left| f_*^{(k)}(t) \right| \right] &= \max_{k \in \{0,1,\ldots,n\}} \left[\sup_{t \in [a,b]} \left| \|f\|_{n,\infty}^{-1} c_{a,b}(b-a)^k [f^{(k)}(t)] \right| \right] \\ &\le \|f\|_{n,\infty}^{-1} \max_{k \in \{0,1,\ldots,n\}} \left[\sup_{t \in [a,b]} \left| f^{(k)}(t) \right| \right] = 1. \end{aligned} \tag{A.104}$$

Theorem 6.5 therefore establishes that there exist neural networks $(\Phi_{g,\eta})_{g \in B_1^n, \eta \in (0,\infty)} \subseteq \mathfrak{N}$ which satisfy

(a) $\displaystyle \sup_{g \in B_1^n, \eta \in (0,\infty)} \left[\frac{\mathcal{L}(\Phi_{g,\eta})}{\max\{r, |\ln(\eta)|\}} \right] < \infty,$

(b) $\displaystyle \sup_{g \in B_1^n, \eta \in (0,\infty)} \left[\frac{\mathcal{M}(\Phi_{g,\eta})}{\eta^{-\frac{1}{n}} \max\{r, |\ln(\eta)|\}} \right] < \infty,$ and

(c) for every $g \in B_1^n$, $\eta \in (0,\infty)$ that

$$\sup_{t \in [0,1]} |g(t) - [R_\varrho(\Phi_{g,\eta})](t)| \le \eta. \tag{A.105}$$

Let $(\Phi_{f,\varepsilon})_{f\in\mathcal{C}^n,\varepsilon\in(0,\infty)} \subseteq \mathfrak{N}$ denote neural networks which satisfy for every $[a,b]\subseteq\mathbb{R}_+$, $f\in C^n([a,b],\mathbb{R})$, $\varepsilon\in(0,\infty)$

$$\Phi_{f,\varepsilon} = \alpha_f \odot \varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}} \odot \lambda_{a,b}. \tag{A.106}$$

Observe that for every $[a,b]\subseteq\mathbb{R}_+$, $f\in C^n([a,b],\mathbb{R})$, $t\in[0,1]$ it holds

$$[R_\varrho(\lambda_{a,b})](t) = \left[\tfrac{1}{(b-a)}\right] t - \tfrac{a}{(b-a)} = L_{a,b}^{-1}(t) \qquad \text{and} \qquad [R_\varrho(\alpha_f)](t) = \tfrac{\|f\|_{n,\infty}}{c_{a,b}} t. \tag{A.107}$$

Lemma 5.3 therefore demonstrates for every $[a,b]\subseteq\mathbb{R}_+$, $f\in C^n([a,b],\mathbb{R})$, $\varepsilon\in(0,\infty)$, $t\in[0,1]$ it holds

$$\begin{aligned}
[R_\varrho(\Phi_{f,\varepsilon})](t) &= [R_\varrho(\alpha_f \odot \varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}} \odot \lambda_{a,b})](t) \\
&= [R_\varrho(\alpha_f)\circ R_\varrho(\varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}})\circ R_\varrho(\lambda_{a,b})](t) \\
&= \tfrac{\|f\|_{n,\infty}}{c_{a,b}} [R_\varrho(\varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}})](L_{a,b}^{-1}(t)).
\end{aligned} \tag{A.108}$$

Moreover, note (A.99) ensures that for every $[a,b]\subseteq\mathbb{R}_+$, $f\in C^n([a,b],\mathbb{R})$, $t\in[a,b]$ it holds

$$f(t) = \tfrac{\|f\|_{n,\infty}}{c_{a,b}} f_*(L_{a,b}^{-1}(t)). \tag{A.109}$$

Combining (c), (A.106), and (A.108) implies for every $[a,b]\subseteq\mathbb{R}_+$, $f\in C^n([a,b],\mathbb{R})$, $\varepsilon\in(0,\infty)$

$$\begin{aligned}
\sup_{t\in[a,b]} |f(t) - [R_\varrho(\Phi_{f,\varepsilon})](t)| &= \sup_{t\in[a,b]} \left| \tfrac{\|f\|_{n,\infty}}{c_{a,b}} f_*(L_{a,b}^{-1}(t)) - \tfrac{\|f\|_{n,\infty}}{c_{a,b}} [R_\varrho(\varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}})](L_{a,b}^{-1}(t)) \right| \\
&= \tfrac{\|f\|_{n,\infty}}{c_{a,b}} \left[\sup_{t\in[0,1]} \left| f_*(t) - [R_\varrho(\varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}})](t) \right| \right] \leq \tfrac{\|f\|_{n,\infty}}{c_{a,b}} \tfrac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}} = \varepsilon.
\end{aligned} \tag{A.110}$$

This establishes that the neural networks $(\Phi_{f,\varepsilon})_{f\in\mathcal{C}^n,\varepsilon\in(0,\infty)}$ satisfy (iii). Furthermore, Lemma 5.3 ensures for every $[a,b]\subseteq\mathbb{R}_+$, $f\in C^n([a,b],\mathbb{R})$, $\varepsilon\in(0,\infty)$ holds

$$\mathcal{L}(\Phi_{f,\varepsilon}) = \mathcal{L}(\alpha_f \odot \varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}} \odot \lambda_{a,b}) = \mathcal{L}(\alpha_f) + \mathcal{L}(\varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}}) + \mathcal{L}(\lambda_{a,b}) = \mathcal{L}(\varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}}) + 2. \tag{A.111}$$

In addition, for every $[a,b]\subseteq\mathbb{R}_+$, $f\in C^n([a,b],\mathbb{R})$, $\varepsilon\in(0,\infty)$ holds

$$\begin{aligned}
\max\{r, |\ln(\tfrac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}})|\} = \max\{r, |\ln(\tfrac{\min\{1,(b-a)^{-n}\}\varepsilon}{\|f\|_{n,\infty}})|\} &= \max\{r, |\ln(\tfrac{\varepsilon}{(\max\{1,(b-a)\})^n\|f\|_{n,\infty}})|\} \\
\leq n \max\{r, |\ln(\tfrac{\varepsilon}{(\max\{1,(b-a)\})\|f\|_{n,\infty}})|\}&.
\end{aligned} \tag{A.112}$$

Combining this with (a) and (A.111) implies that

$$\begin{aligned}
\sup_{f\in\mathcal{C}^n,\varepsilon\in(0,\infty)} \left[\frac{\mathcal{L}(\Phi_{f,\varepsilon})}{\max\{r, |\ln(\frac{\varepsilon}{\max\{1,b-a\}\|f\|_{n,\infty}})|\}} \right] &\leq n \sup_{f\in\mathcal{C}^n,\varepsilon\in(0,\infty)} \left[\frac{\mathcal{L}(\varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}}) + 2}{\max\{r, |\ln(\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}})|\}} \right] \\
&= n \sup_{g\in B_1^n,\eta\in(0,\infty)} \left[\frac{\mathcal{L}(\Phi_{g,\eta}) + 2}{\max\{r, |\ln(\eta)|\}} \right] < \infty.
\end{aligned} \tag{A.113}$$

This establishes that the neural networks $(\Phi_{f,\varepsilon})_{f\in\mathcal{C}^n,\varepsilon\in(0,\infty)}$ satisfy (i). Next, Lemma 5.3 implies that for every $[a,b]\subseteq\mathbb{R}_+$, $f\in C^n([a,b],\mathbb{R})$, $\varepsilon\in(0,\infty)$

$$\mathcal{M}(\Phi_{f,\varepsilon}) = \mathcal{M}(\alpha_f \odot \varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}} \odot \lambda_{a,b}) = \mathcal{M}(\alpha_f) + \mathcal{M}(\varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}}) + \mathcal{M}(\lambda_{a,b}) = \mathcal{M}(\varphi_{f_*,\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}}) + 3. \tag{A.114}$$

In addition, note that (A.112) shows for every $[a,b] \subseteq \mathbb{R}_+$, $f \in C^n([a,b],\mathbb{R})$, $\varepsilon \in (0,\infty)$

$$\left[\tfrac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}\right]^{-\frac{1}{n}} \max\{r, |\ln(\tfrac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}})|\}n \leq \max\{1, b-a\} \, \|f\|_{n,\infty}^{\frac{1}{n}} \, \varepsilon^{-\frac{1}{n}} \max\{r, |\ln(\tfrac{\varepsilon}{\max\{1,b-a\}\|f\|_{n,\infty}})|\}. \tag{A.115}$$

Combining this with (b) and (A.106) therefore ensures

$$\begin{aligned} &\sup_{f \in \mathcal{C}^n, \varepsilon \in (0,\infty)} \left[\frac{\mathcal{M}(\Phi_{f,\varepsilon})}{\max\{1, b-a\} \, \|f\|_{n,\infty}^{\frac{1}{n}} \, \varepsilon^{-\frac{1}{n}} \max\{r, |\ln(\frac{\varepsilon}{\max\{1,b-a\}\|f\|_{n,\infty}})|\}}\right] \\ &\leq n \sup_{f \in \mathcal{C}^n, \varepsilon \in (0,\infty)} \left[\frac{\mathcal{M}(\varphi_{f_*, \frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}}) + 3}{\left[\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}}\right]^{-\frac{1}{n}} \max\{r, |\ln(\frac{c_{a,b}\varepsilon}{\|f\|_{n,\infty}})|\}}\right] \\ &\leq n \sup_{g \in B_1^n, \eta \in (0,\infty)} \left[\frac{\mathcal{M}(\Phi_{g,\eta}) + 3}{\eta^{-\frac{1}{n}} \max\{r, |\ln(\eta)|\}}\right] < \infty. \end{aligned} \tag{A.116}$$

This establishes that the neural networks $(\Phi_{f,\varepsilon})_{f \in \mathcal{C}^n, \varepsilon \in (0,\infty)}$ satisfy (ii) and completes the proof. □

II. Deep Neural Network Approximation Theory

Authors: Dennis Elbrächter, Dmytro Perekrestenko, Philipp Grohs, and Helmut Bölcskei

Status: Accepted for publication in IEEE Transactions on Information Theory (10.01.2021).

Arxiv: https://arxiv.org/abs/1901.02220

Contribution: Major contribution to the development of the results and writing.

Deep Neural Network Approximation Theory

Dennis Elbrächter, Dmytro Perekrestenko, Philipp Grohs, and Helmut Bölcskei

Abstract

This paper develops fundamental limits of deep neural network learning by characterizing what is possible if no constraints are imposed on the learning algorithm and on the amount of training data. Concretely, we consider Kolmogorov-optimal approximation through deep neural networks with the guiding theme being a relation between the complexity of the function (class) to be approximated and the complexity of the approximating network in terms of connectivity and memory requirements for storing the network topology and the associated quantized weights. The theory we develop establishes that deep networks are Kolmogorov-optimal approximants for markedly different function classes, such as unit balls in Besov spaces and modulation spaces. In addition, deep networks provide exponential approximation accuracy—i.e., the approximation error decays exponentially in the number of nonzero weights in the network—of the multiplication operation, polynomials, sinusoidal functions, and certain smooth functions. Moreover, this holds true even for one-dimensional oscillatory textures and the Weierstrass function—a fractal function, neither of which has previously known methods achieving exponential approximation accuracy. We also show that in the approximation of sufficiently smooth functions finite-width deep networks require strictly smaller connectivity than finite-depth wide networks.

I. Introduction

Triggered by the availability of vast amounts of training data and drastic improvements in computing power, deep neural networks have become state-of-the-art technology for a wide range of practical machine learning tasks such as image classification [1], handwritten digit recognition [2], speech recognition [3], or game intelligence [4]. For an in-depth overview, we refer to the survey paper [5] and the recent book [6].

A neural network effectively implements a mapping approximating a function that is learned based on a given set of input-output value pairs, typically through the backpropagation algorithm [7]. Characterizing the fundamental limits of approximation through neural networks shows what is possible if no constraints are imposed on the learning algorithm and on the amount of training data [8].

D. Elbrächter is with the Department of Mathematics, University of Vienna, Austria (e-mail: dennis.elbraechter@univie.ac.at).

D. Perekrestenko and H. Bölcskei are with the Chair for Mathematical Information Science, ETH Zurich, Switzerland (e-mail: pdmytro@mins.ee.ethz.ch, hboelcskei@ethz.ch).

P. Grohs is with the Department of Mathematics and the Research Platform DataScience@UniVienna, University of Vienna, Austria (e-mail: philipp.grohs@univie.ac.at).

D. Elbrächter was supported through the FWF projects P 30148 and I 3403 as well as the WWTF project ICT19-041.

The theory of function approximation through neural networks has a long history dating back to the work by McCulloch and Pitts [9] and the seminal paper by Kolmogorov [10], who showed, when interpreted in neural network parlance, that any continuous function of n variables can be represented exactly through a 2-layer neural network of width $2n + 1$. However, the nonlinearities in Kolmogorov's neural network are highly nonsmooth and the outer nonlinearities, i.e., those in the output layer, depend on the function to be represented. In modern neural network theory, one is usually interested in networks with nonlinearities that are independent of the function to be realized and exhibit, in addition, certain smoothness properties. Significant progress in understanding the approximation capabilities of such networks has been made in [11], [12], where it was shown that single-hidden-layer neural networks can approximate continuous functions on bounded domains arbitrarily well, provided that the activation function satisfies certain (mild) conditions and the number of nodes is allowed to grow arbitrarily large. In practice one is, however, often interested in approximating functions from a given function class $\mathcal{C}$ determined by the application at hand. It is therefore natural to ask how the complexity of a neural network approximating every function in $\mathcal{C}$ to within a prescribed accuracy depends on the complexity of $\mathcal{C}$ (and on the desired approximation accuracy). The recently developed Kolmogorov-Donoho rate-distortion theory for neural networks [13] formalizes this question by relating the complexity of $\mathcal{C}$—in terms of the number of bits needed to describe any element in $\mathcal{C}$ to within prescribed accuracy—to network complexity in terms of connectivity and memory requirements for storing the network topology and the associated quantized weights. The theory is based on a framework for quantifying the fundamental limits of nonlinear approximation through dictionaries as introduced by Donoho [14], [15].

The purpose of this paper is to provide a comprehensive, principled, and self-contained introduction to Kolmogorov-Donoho rate-distortion optimal approximation through deep neural networks. The idea is to equip the reader with a working knowledge of the mathematical tools underlying the theory at a level that is sufficiently deep to enable further research in the field. Part of this paper is based on [13], but extends the theory therein to the rectified linear unit (ReLU) activation function and to networks with depth scaling in the approximation error.

The theory we develop educes remarkable universality properties of finite-width deep networks. Specifically, deep networks are Kolmogorov-Donoho optimal approximants for vastly different function classes such as unit balls in Besov spaces [16] and modulation spaces [17]. This universality is afforded by a concurrent invariance property of deep networks to time-shifts, scalings, and frequency-shifts. In addition, deep networks provide exponential approximation accuracy—i.e., the approximation error decays exponentially in the number of parameters employed in the approximant, namely the number of nonzero weights in the network—for vastly different functions such as the squaring operation, multiplication, polynomials, sinusoidal functions, general smooth functions, and even one-dimensional oscillatory textures [18] and the Weierstrass function—a fractal function, neither of which has known methods achieving exponential approximation accuracy.

While we consider networks based on the ReLU[1] activation function throughout, certain parts of our theory carry over to strongly sigmoidal activation functions of order $k \geq 2$ as defined in [13]. For the sake of conciseness, we refrain from providing these extensions.

Outline of the paper. In Section II, we introduce notation, formally define neural networks, and record basic elements needed in the neural network constructions throughout the paper. Section III presents an algebra of function approximation by neural networks. In Section IV, we develop the Kolmogorov-Donoho rate-distortion framework that will allow us to characterize the fundamental limits of deep neural network learning of function classes. This theory is based on the concept of metric entropy, which is introduced and reviewed starting from first principles. Section V then puts the Kolmogorov-Donoho framework to work in the context of nonlinear function approximation with dictionaries. This discussion serves as a basis for the development of the concept of best M-weight approximation in neural networks presented in Section VI. We proceed, in Section VII, with the development of a method—termed the transference principle—for transferring results on function approximation through dictionaries to results on approximation by neural networks. The purpose of Section VIII is to demonstrate that function classes that are optimally approximated by affine dictionaries (e.g., wavelets), are optimally approximated by neural networks as well. In Section IX, we show that this optimality transfer extends to function classes that are optimally approximated by Weyl-Heisenberg dictionaries. Section X demonstrates that neural networks can improve the best-known approximation rates for two example functions, namely oscillatory textures and the Weierstrass function, from polynomial to exponential. The final Section XI makes a formal case for depth in neural network approximation by establishing a provable benefit of deep networks over shallow networks in the approximation of sufficiently smooth functions. The Appendices collect ancillary technical results.

Notation. For a function $f(x)\colon \mathbb{R}^d \to \mathbb{R}$ and a set $\Omega \subseteq \mathbb{R}^d$, we define $\|f\|_{L^\infty(\Omega)} := \sup\{|f(x)| : x \in \Omega\}$. $L^p(\mathbb{R}^d)$ and $L^p(\mathbb{R}^d, \mathbb{C})$ denote the space of real-valued, respectively complex-valued, L^p-functions. When dealing with the approximation error for simple functions such as, e.g., $(x, y) \mapsto xy$, we will for brevity of exposition and with slight abuse of notation, make the arguments inside the norm explicit according to $\|f(x, y) - xy\|_{L^p(\Omega)}$. For a vector $b \in \mathbb{R}^d$, we let $\|b\|_\infty := \max_{i=1,\dots,d} |b_i|$, similarly we write $\|A\|_\infty := \max_{i,j} |A_{i,j}|$ for the matrix $A \in \mathbb{R}^{m\times n}$. We denote the identity matrix of size $n \times n$ by $\mathbb{I}_n$. log stands for the logarithm to base 2. For a set $X \in \mathbb{R}^d$, we write $|X|$ for its Lebesgue measure. Constants like C are understood to be allowed to take on different values in different uses.

[1] ReLU stands for the Rectified Linear Unit nonlinearity defined as $x \mapsto \max\{0, x\}$.

II. Setup and basic ReLU calculus

This section defines neural networks, introduces the basic setup as well as further notation, and lists basic elements needed in the neural network constructions considered throughout, namely compositions and linear combinations of neural networks. There is a plethora of neural network architectures and activation functions in the literature. Here, we restrict ourselves to the ReLU activation function and consider the following general network architecture.

Definition II.1. *Let $L \in \mathbb{N}$ and $N_0, N_1, \ldots, N_L \in \mathbb{N}$. A* ReLU neural network Φ *is a map $\Phi : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ given by*

$$\Phi = \begin{cases} W_1, & L = 1 \\ W_2 \circ \rho \circ W_1, & L = 2 \\ W_L \circ \rho \circ W_{L-1} \circ \rho \circ \cdots \circ \rho \circ W_1, & L \geq 3 \end{cases}, \tag{1}$$

where, for $\ell \in \{1, 2, \ldots, L\}$, $W_\ell \colon \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}, W_\ell(x) := A_\ell x + b_\ell$ are the associated affine transformations with matrices $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and (bias) vectors $b_\ell \in \mathbb{R}^{N_\ell}$, and the ReLU activation function $\rho \colon \mathbb{R} \to \mathbb{R}$, $\rho(x) := \max(0, x)$ acts component-wise, i.e., $\rho(x_1, \ldots, x_N) := (\rho(x_1), \ldots, \rho(x_N))$. We denote by $\mathcal{N}_{d,d'}$ the set of all ReLU networks with input dimension $N_0 = d$ and output dimension $N_L = d'$. Moreover, we define the following quantities related to the notion of size of the ReLU network Φ:

- *the* connectivity $\mathcal{M}(\Phi)$ *is the total number of nonzero entries in the matrices A_ℓ, $\ell \in \{1, 2, \ldots, L\}$, and the vectors b_ℓ, $\ell \in \{1, 2, \ldots, L\}$,*
- depth $\mathcal{L}(\Phi) := L$,
- width $\mathcal{W}(\Phi) := \max_{\ell=0,\ldots,L} N_\ell$,
- weight magnitude $\mathcal{B}(\Phi) := \max_{\ell=1,\ldots,L} \max\{\|A_\ell\|_\infty, \|b_\ell\|_\infty\}$.

Remark II.2. *Note that for a given function $f : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$, which can be expressed according to* (1)*, the underlying affine transformations W_ℓ are highly nonunique in general [19], [20]. The question of uniqueness in this context is of independent interest and was addressed recently in [21], [22]. Whenever we talk about a given ReLU network Φ, we will either explicitly or implicitly associate Φ with a given set of affine transformations W_ℓ.*

N_0 *is the* dimension of the input layer *indexed as the* 0*-th layer,* $N_1, \ldots, N_{L-1}$ *are the* dimensions of the $L-1$ hidden layers*, and* N_L *is the* dimension of the output layer. *Our definition of depth $\mathcal{L}(\Phi)$ counts the number of affine transformations involved in the representation (1). Single-hidden-layer neural networks hence have depth* 2 *in this terminology. Finally, we consider standard affine transformations as neural networks of depth* 1 *for technical purposes.*

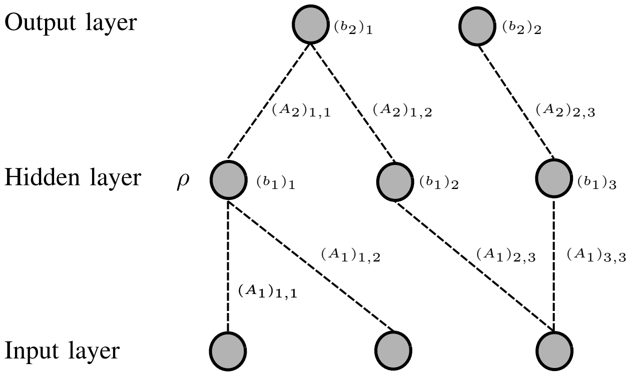
The matrix entry $(A_\ell)_{i,j}$ represents the weight associated with the edge between the j-th node in the $(\ell-1)$-th layer and the i-th node in the ℓ-th layer, $(b_\ell)_i$ *is* the weight associated with the i-th node in the ℓ-th layer. *These* *assignments are schematized in Figure 1. The real numbers $(A_\ell)_{i,j}$ and $(b_\ell)_i$ are referred to as the network's edge weights and node weights, respectively.*

Throughout the paper, we assume that every node in the input layer and in layers $1, \ldots, L-1$ has at least one outgoing edge and every node in the output layer L has at least one incoming edge. These nondegeneracy assumptions are basic as nodes that do not satisfy them can be removed without changing the functional relationship realized by the network.

Finally, we note that the connectivity satisfies

$$\mathcal{M}(\Phi) \leq \mathcal{L}(\Phi)\mathcal{W}(\Phi)(\mathcal{W}(\Phi)+1).$$

The term "network" stems from the interpretation of the mapping Φ as a weighted acyclic directed graph with nodes arranged in hierarchical layers and edges only between adjacent layers.



$$A_2 = \begin{pmatrix} (A_2)_{1,1} & (A_2)_{1,2} & 0 \\ 0 & 0 & (A_2)_{2,3} \end{pmatrix}$$

$$A_1 = \begin{pmatrix} (A_1)_{1,1} & (A_1)_{1,2} & 0 \\ 0 & 0 & (A_1)_{2,3} \\ 0 & 0 & (A_1)_{3,3} \end{pmatrix}$$

Fig. 1: Assignment of the weights $(A_\ell)_{i,j}$ and $(b_\ell)_i$ of a two-layer network to the edges and nodes, respectively.

We mostly consider the case $\Phi : \mathbb{R}^d \to \mathbb{R}$, i.e., $N_L = 1$, but emphasize that our results readily generalize to $N_L > 1$.

The neural network constructions provided in the paper frequently make use of basic elements introduced next, namely compositions and linear combinations of networks [23].

Lemma II.3. *Let $d_1, d_2, d_3 \in \mathbb{N}$, $\Phi_1 \in \mathcal{N}_{d_1,d_2}$, and $\Phi_2 \in \mathcal{N}_{d_2,d_3}$. Then, there exists a network $\Psi \in \mathcal{N}_{d_1,d_3}$ with $\mathcal{L}(\Psi) = \mathcal{L}(\Phi_1) + \mathcal{L}(\Phi_2)$, $\mathcal{M}(\Psi) \leq 2\mathcal{M}(\Phi_1) + 2\mathcal{M}(\Phi_2)$, $\mathcal{W}(\Psi) \leq \max\{2d_2, \mathcal{W}(\Phi_1), \mathcal{W}(\Phi_2)\}$, $\mathcal{B}(\Psi) = \max\{\mathcal{B}(\Phi_1), \mathcal{B}(\Phi_2)\}$, and satisfying*

$$\Psi(x) = (\Phi_2 \circ \Phi_1)(x) = \Phi_2(\Phi_1(x)), \quad \textit{for all } x \in \mathbb{R}^{d_1}.$$

Proof. The proof is based on the identity $x = \rho(x) - \rho(-x)$. First, note that by Definition II.1, we can write

$$\Phi_1 = W^1_{L_1} \circ \rho \circ W^1_{L_1-1} \circ \cdots \circ \rho \circ W^1_1 \quad \text{and} \quad \Phi_2 = W^2_{L_2} \circ \rho \circ \cdots \circ W^2_2 \circ \rho \circ W^2_1.$$

Next, let $N^1_{L_1-1}$ denote the width of layer $L_1 - 1$ in Φ_1 and let N^2_1 denote the width of layer 1 in Φ_2. We define the affine transformations $\widetilde{W}^1_{L_1} : \mathbb{R}^{N^1_{L_1-1}} \mapsto \mathbb{R}^{2d_2}$ and $\widetilde{W}^2_1 : \mathbb{R}^{2d_2} \mapsto \mathbb{R}^{N^2_1}$ according to

$$\widetilde{W}^1_{L_1}(x) := \begin{pmatrix} \mathbb{I}_{d_2} \\ -\mathbb{I}_{d_2} \end{pmatrix} W^1_{L_1}(x) \quad \text{and } \widetilde{W}^2_1(y) := W^2_1 \left(\begin{pmatrix} \mathbb{I}_{d_2} & -\mathbb{I}_{d_2} \end{pmatrix} y \right).$$

The proof is finalized by noting that the network

$$\Psi := W^2_{L_2} \circ \rho \circ \cdots \circ W^2_2 \circ \rho \circ \widetilde{W}^2_1 \circ \rho \circ \widetilde{W}^1_{L_1} \circ \rho \circ W^1_{L_1-1} \circ \cdots \circ \rho \circ W^1_1$$

satisfies the claimed properties. □

Unless explicitly stated otherwise, the composition of two neural networks will be understood in the sense of Lemma II.3.

In order to formalize the concept of a linear combination of networks with possibly different depths, we need the following two technical lemmas which show how to augment network depth while retaining the network's input-output relation and how to parallelize networks.

Lemma II.4. *Let $d_1, d_2, K \in \mathbb{N}$, and $\Phi \in \mathcal{N}_{d_1,d_2}$ with $\mathcal{L}(\Phi) < K$. Then, there exists a network $\Psi \in \mathcal{N}_{d_1,d_2}$ with $\mathcal{L}(\Psi) = K$, $\mathcal{M}(\Psi) \leq \mathcal{M}(\Phi) + d_2\mathcal{W}(\Phi) + 2d_2(K - \mathcal{L}(\Phi))$, $\mathcal{W}(\Psi) = \max\{2d_2, \mathcal{W}(\Phi)\}$, $\mathcal{B}(\Psi) = \max\{1, \mathcal{B}(\Phi)\}$, and satisfying $\Psi(x) = \Phi(x)$ for all $x \in \mathbb{R}^{d_1}$.*

Proof. Let $\widetilde{W}_j(x) := \operatorname{diag}(\mathbb{I}_{d_2}, \mathbb{I}_{d_2})\, x$, for $j \in \{\mathcal{L}(\Phi) + 1, \dots, K - 1\}$, $\widetilde{W}_K(x) := \begin{pmatrix} \mathbb{I}_{d_2} & -\mathbb{I}_{d_2} \end{pmatrix} x$, and note that with

$$\Phi = W_{\mathcal{L}(\Phi)} \circ \rho \circ W_{\mathcal{L}(\Phi)-1} \circ \rho \circ \cdots \circ \rho \circ W_1,$$

the network

$$\Psi := \widetilde{W}_K \circ \rho \circ \widetilde{W}_{K-1} \circ \rho \circ \cdots \circ \rho \circ \widetilde{W}_{\mathcal{L}(\Phi)+1} \circ \rho \circ \begin{pmatrix} W_{\mathcal{L}(\Phi)} \\ -W_{\mathcal{L}(\Phi)} \end{pmatrix} \circ \rho \circ W_{\mathcal{L}(\Phi)-1} \circ \rho \circ \cdots \circ \rho \circ W_1$$

satisfies the claimed properties. □

For the sake of simplicity of exposition, we state the following two lemmas only for networks of the same depth, the extension to the general case follows by straightforward application of Lemma II.4. The first of these two lemmas formalizes the notion of neural network parallelization, concretely of combining neural networks implementing the functions f and g into a neural network realizing the mapping $x \mapsto (f(x), g(x))$.

Lemma II.5. *Let $n, L \in \mathbb{N}$ and, for $i \in \{1, 2, \dots, n\}$, let $d_i, d_i' \in \mathbb{N}$ and $\Phi_i \in \mathcal{N}_{d_i, d_i'}$ with $\mathcal{L}(\Phi_i) = L$. Then, there exists a network $\Psi \in \mathcal{N}_{\sum_{i=1}^n d_i, \sum_{i=1}^n d_i'}$ with $\mathcal{L}(\Psi) = L$, $\mathcal{M}(\Psi) = \sum_{i=1}^n \mathcal{M}(\Phi_i)$, $\mathcal{W}(\Psi) = \sum_{i=1}^n \mathcal{W}(\Phi_i)$, $\mathcal{B}(\Psi) = \max_i \mathcal{B}(\Phi_i)$, and satisfying*

$$\Psi(x) = (\Phi_1(x_1), \Phi_2(x_2), \dots, \Phi_n(x_n)) \in \mathbb{R}^{\sum_{i=1}^n d_i'},$$

for $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^{\sum_{i=1}^n d_i}$ with $x_i \in \mathbb{R}^{d_i}$, $i \in \mathbb{N}$.

Proof. We write the networks Φ_i as

$$\Phi_i = W_L^i \circ \rho \circ W_{L-1}^i \circ \rho \circ \cdots \circ \rho \circ W_1^i,$$

with $W_\ell^i(x) = A_\ell^i x + b_\ell^i$. Furthermore, we denote the layer dimensions of Φ_i by $N_0^i, \dots, N_L^i$ and set $N_\ell := \sum_{i=1}^n N_\ell^i$, for $\ell \in \{0, 1, \dots, L\}$. Next, define, for $\ell \in \{1, 2, \dots, L\}$, the block-diagonal matrices $A_\ell := \operatorname{diag}(A_\ell^1, A_\ell^2, \dots, A_\ell^n)$, the vectors $b_\ell = (b_\ell^1, b_\ell^2, \dots, b_\ell^n)$, and the affine transformations $W_\ell(x) := A_\ell x + b_\ell$. The proof is concluded by noting that

$$\Psi := W_L \circ \rho \circ W_{L-1} \circ \rho \circ \cdots \circ \rho \circ W_1$$

satisfies the claimed properties. □

We are now ready to formalize the concept of a linear combination of neural networks.

Lemma II.6. *Let $n, L, d' \in \mathbb{N}$ and, for $i \in \{1, 2, \dots, n\}$, let $d_i \in \mathbb{N}$, $a_i \in \mathbb{R}$, and $\Phi_i \in \mathcal{N}_{d_i, d'}$ with $\mathcal{L}(\Phi_i) = L$. Then, there exists a network $\Psi \in \mathcal{N}_{\sum_{i=1}^n d_i, d'}$ with $\mathcal{L}(\Psi) = L$, $\mathcal{M}(\Psi) \le \sum_{i=1}^n \mathcal{M}(\Phi_i)$, $\mathcal{W}(\Psi) \le \sum_{i=1}^n \mathcal{W}(\Phi_i)$, $\mathcal{B}(\Psi) = \max_i\{|a_i| \mathcal{B}(\Phi_i)\}$, and satisfying*

$$\Psi(x) = \sum_{i=1}^n a_i \Phi_i(x_i) \in \mathbb{R}^{d'},$$

for $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^{\sum_{i=1}^n d_i}$ with $x_i \in \mathbb{R}^{d_i}$, $i \in \{1, 2, \dots, n\}$.

Proof. The proof follows by taking the construction in Lemma II.5, replacing A_L by $(a_1 A_L^1, a_2 A_L^2, \dots, a_n A_L^n)$, b_L by $\sum_{i=1}^n a_i b_L^i$, and noting that the resulting network satisfies the claimed properties. □

III. Approximation of multiplication, polynomials, smooth functions, and sinusoidals

This section constitutes the first part of the paper dealing with the approximation of basic function “templates” through neural networks. Specifically, we shall develop an algebra of neural network approximation by starting with the squaring function, building thereon to approximate the multiplication function, proceeding to polynomials and general smooth functions, and ending with sinusoidal functions.

The basic element of the neural network algebra we develop is based on an approach by Yarotsky [24] and by Schmidt-Hieber [25], both of whom, in turn, employed the “sawtooth” construction from [26].

We start by reviewing the sawtooth construction underlying our program. Consider the hat function $g : \mathbb{R} \to [0,1]$,

$$g(x) = 2\rho(x) - 4\rho(x - \tfrac{1}{2}) + 2\rho(x-1) = \begin{cases} 2x, & \text{if } 0 \le x < \frac{1}{2} \\ 2(1-x), & \text{if } \frac{1}{2} \le x \le 1 \\ 0, & \text{else} \end{cases},$$

let $g_0(x) = x, g_1(x) = g(x)$, and define the s-th order sawtooth function g_s as the s-fold composition of g with itself, i.e.,

$$g_s := \underbrace{g \circ g \circ \cdots \circ g}_{s}, \quad s \ge 2. \tag{2}$$

We note that g can be realized by a 2-layer network $\Phi_g \in \mathcal{N}_{1,1}$ according to $\Phi_g := W_2 \circ \rho \circ W_1 = g$ with

$$W_1(x) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} x \ - \begin{pmatrix} 0 \\ 1/2 \\ 1 \end{pmatrix}, \qquad W_2(x) = \begin{pmatrix} 2 & -4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

The s-th order sawtooth function g_s can hence be realized by a network $\Phi_g^s \in \mathcal{N}_{1,1}$ according to

$$\Phi_g^s := W_2 \circ \rho \circ \underbrace{W_g \circ \rho \circ \cdots \circ W_g \circ \rho}_{s-1} \circ W_1 = g_s \tag{3}$$

with

$$W_g(x) = \begin{pmatrix} 2 & -4 & 2 \\ 2 & -4 & 2 \\ 2 & -4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - \begin{pmatrix} 0 \\ 1/2 \\ 1 \end{pmatrix}.$$

The following restatement of [26, Lemma 2.4] summarizes the self-similarity and symmetry properties of $g_s(x)$ we will frequently make use of.

Lemma III.1. *For $s \in \mathbb{N}$, $k \in \{0, 1, \dots, 2^{s-1} - 1\}$, it holds that $g(2^{s-1} \cdot -k)$ is supported in $\left[\frac{k}{2^{s-1}}, \frac{k+1}{2^{s-1}}\right]$,*

$$g_s(x) = \sum_{k=0}^{2^{s-1}-1} g(2^{s-1}x - k), \quad \text{for } x \in [0,1],$$

and

$$g_s\left(\tfrac{k}{2^{s-1}} + x\right) = g_s\left(\tfrac{k+1}{2^{s-1}} - x\right), \quad \text{for } x \in \left[0, \tfrac{1}{2^{s-1}}\right].$$

We are now ready to proceed with the statement of the basic building block of our neural network algebra, namely the approximation of the squaring function through deep ReLU networks.

Proposition III.2. *There exists a constant $C > 0$ such that for all $\varepsilon \in (0, 1/2)$, there is a network $\Phi_\varepsilon \in \mathcal{N}_{1,1}$ with $\mathcal{L}(\Phi_\varepsilon) \le C \log(\varepsilon^{-1})$, $\mathcal{W}(\Phi_\varepsilon) = 3$, $\mathcal{B}(\Phi_\varepsilon) \le 1$, $\Phi_\varepsilon(0) = 0$, satisfying*

$$\|\Phi_\varepsilon(x) - x^2\|_{L^\infty([0,1])} \le \varepsilon.$$

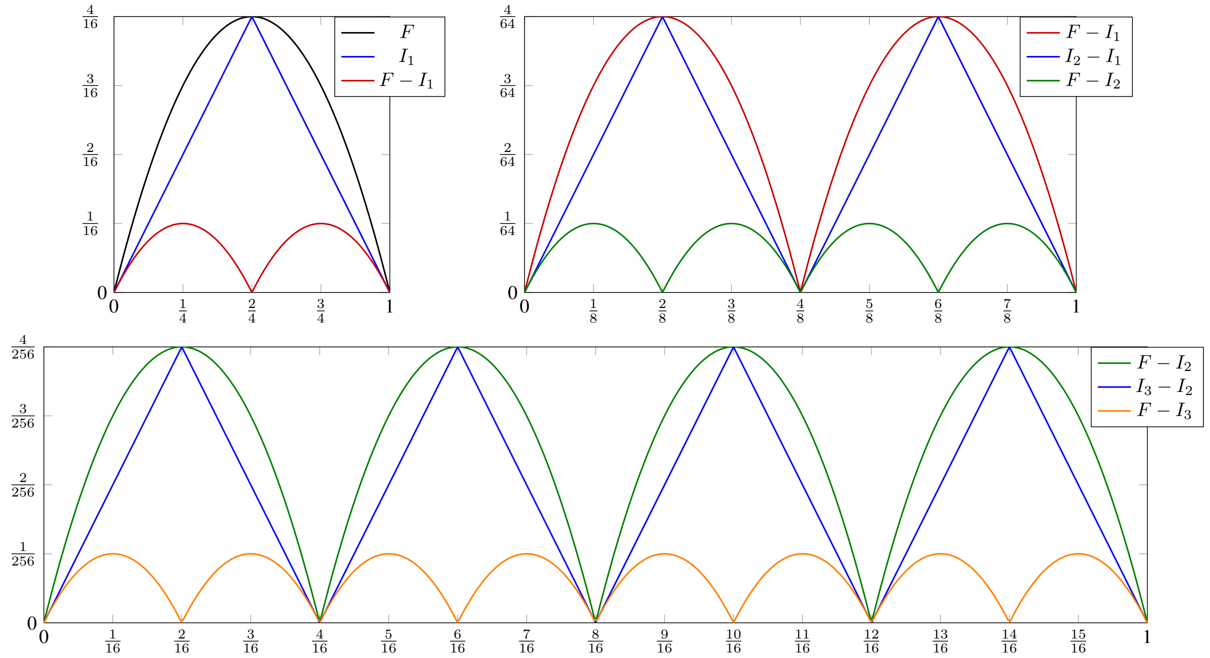


Fig. 2: First three steps of approximating $F(x) = x - x^2$ by an equispaced linear interpolation I_m at $2^m + 1$ points.

Proof. The proof builds on two rather elementary observations. The first one concerns the linear interpolation $I_m\colon [0,1] \to \mathbb{R}$, $m \in \mathbb{N}$, of the function $F(x) := x - x^2$ at the points $\frac{j}{2^m}$, $j \in \{0, 1, \dots, 2^m\}$, and in particular the self-similarity of the refinement step $I_m \to I_{m+1}$. For every $m \in \mathbb{N}$, the residual $F - I_m$ is identical on each interval between two points of interpolation. Concretely, let $f_m\colon [0, 2^{-m}] \to [0, 2^{-2m-2}]$ be defined as $f_m(x) = 2^{-m}x - x^2$ and consider its linear interpolation $h_m\colon [0, 2^{-m}] \to [0, 2^{-2m-2}]$ at the midpoint and the endpoints of the interval $[0, 2^{-m}]$ given by

$$h_m(x) := \begin{cases} 2^{-m-1}x, & x \in [0, 2^{-m-1}] \\ -2^{-m-1}x + 2^{-2m-1}, & x \in [2^{-m-1}, 2^{-m}] \end{cases}.$$

Direct calculation shows that

$$f_m(x) - h_m(x) = \begin{cases} f_{m+1}(x), & x \in [0, 2^{-m-1}] \\ f_{m+1}(x - 2^{-m-1}), & x \in [2^{-m-1}, 2^{-m}] \end{cases}.$$

As $F = f_0$ and $I_1 = h_0$ this implies that, for all $m \in \mathbb{N}$,

$$F(x) - I_m(x) = f_m(x - \tfrac{j}{2^m}), \ \text{ for } \ x \in [\tfrac{j}{2^m}, \tfrac{j+1}{2^m}], \ \ j \in \{0, 1, \dots, 2^m - 1\}$$

and $I_m = \sum_{k=0}^{m-1} H_k$, where $H_k\colon [0,1] \to \mathbb{R}$ is given by

$$H_k(x) = h_k(x - \tfrac{j}{2^k}), \ \text{ for } \ x \in [\tfrac{j}{2^k}, \tfrac{j+1}{2^k}], \ \ j \in \{0, 1, \dots, 2^k - 1\}.$$

Thus, we have

$$\sup_{x\in[0,1]} |x^2 - (x - I_m(x))| = \sup_{x\in[0,1]} |F(x) - I_m(x)| = \sup_{x\in[0,2^{-m}]} |f_m(x)| = 2^{-2m-2}. \tag{4}$$

The second observation we build on is a manifestation of the sawtooth construction described above and leads to economic realizations of the H_k through k-layer networks with two neurons in each layer; a third neuron is used to realize the approximation $x - I_m(x)$ to x^2. Concretely, let $s_k(x) := 2^{-1}\rho(x) - \rho(x - 2^{-2k-1})$, and note that, for $x \in [0,1]$, $H_0 = s_0$, we get $H_k = s_k \circ H_{k-1}$. We can thus construct a network realizing $x - I_m(x)$, for $x \in [0,1]$, as follows. Let $A_1 := (1,1,1)^T \in \mathbb{R}^{3\times 1}$, $b_1 := (0,-2^{-1},0)^T \in \mathbb{R}^3$,

$$A_\ell := \begin{pmatrix} 2^{-1} & -1 & 0 \\ 2^{-1} & -1 & 0 \\ -2^{-1} & 1 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 3}, \quad b_\ell := \begin{pmatrix} 0 \\ -2^{-2\ell+1} \\ 0 \end{pmatrix} \in \mathbb{R}^3, \quad \text{for } \ell \in \{2,\dots,m\},$$

and $A_{m+1} := (-2^{-1},1,1) \in \mathbb{R}^{1\times 3}$, $b_{m+1} = 0$. Setting $W_\ell(x) := A_\ell x + b_\ell$, $\ell \in \{1,2,\dots,m+1\}$, and

$$\tilde{\Phi}_m := W_{m+1} \circ \rho \circ W_m \circ \rho \circ \cdots \circ \rho \circ W_1,$$

a direct calculation yields $\tilde{\Phi}_m(x) = x - \sum_{k=0}^{m-1} H_k(x)$, for $x \in [0,1]$. The proof is completed upon noting that the networks $\Phi_\varepsilon := \tilde{\Phi}_{\lceil \log(\varepsilon^{-1})/2 \rceil}$ satisfy the claimed properties. □

The symmetry properties of $g_s(x)$ according to Lemma III.1 lead to the interpolation error in the proof of Proposition III.2 being identical in each interval, with the maximum error taken on at the centers of the respective intervals. More importantly, however, the approximating neural networks realize linear interpolation at a number of points that grows exponentially in network depth. This is a manifestation of the fact that the number of linear regions in the sawtooth construction (3) grows exponentially with depth, which, owing to Lemma XI.1, is optimal. We emphasize that the theory developed in this paper hinges critically on this optimality property, which, however, is brittle in the sense that networks with weights obtained through training will, as observed in [27], in general, not exhibit exponential growth of the number of linear regions with network depth. An interesting approach to neural network training which manages to partially circumvent this problem was proposed recently in [28]. Understanding how the number of linear regions grows in general trained networks and quantifying the impact of this—possibly subexponential—growth behavior on the approximation-theoretic fundamental limits of neural networks constitutes a major open problem.

We proceed to the construction of networks that approximate the multiplication function over the interval $[-D,D]$. This will be effected by using the result on the approximation of x^2 just established combined with the polarization identity $xy = \frac{1}{4}((x+y)^2 - (x-y)^2)$, the fact that $\rho(x) + \rho(-x) = |x|$, and a scaling argument exploiting that the ReLU function is positive homogeneous, i.e., $\rho(\lambda x) = \lambda\rho(x)$, for all $\lambda \geq 0$, $x \in \mathbb{R}$.

Proposition III.3. *There exists a constant $C > 0$ such that, for all $D \in \mathbb{R}_+$ and $\varepsilon \in (0, 1/2)$, there is a network $\Phi_{D,\varepsilon} \in \mathcal{N}_{2,1}$ with $\mathcal{L}(\Phi_{D,\varepsilon}) \leq C(\log(\lceil D \rceil) + \log(\varepsilon^{-1}))$, $\mathcal{W}(\Phi_{D,\varepsilon}) \leq 5$, $\mathcal{B}(\Phi_{D,\varepsilon}) \leq 1$, satisfying $\Phi_{D,\varepsilon}(0, x) = \Phi_{D,\varepsilon}(x, 0) = 0$, for all $x \in \mathbb{R}$, and*

$$\|\Phi_{D,\varepsilon}(x, y) - xy\|_{L^\infty([-D,D]^2)} \leq \varepsilon. \tag{5}$$

Proof. We first note that, w.l.o.g., we can assume $D \geq 1$ in the following, as for $D < 1$, we can simply employ the network constructed for $D = 1$ to guarantee the claimed properties. The proof builds on the polarization identity and essentially constructs two squaring networks according to Proposition III.2 which share the neuron responsible for summing up the H_k, preceded by a layer mapping (x, y) to $(|x + y|/2D, |x - y|/2D|)$ and followed by layers realizing the multiplication by D^2 through weights bounded by 1. Specifically, consider the network $\tilde{\Psi}_m$ with associated matrices A_ℓ and vectors b_ℓ given by

$$A_1 := \frac{1}{2D}\begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} \in \mathbb{R}^{4\times 2}, \quad b_1 := 0 \in \mathbb{R}^4, \quad A_2 := \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & -1 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \in \mathbb{R}^{5\times 4}, \quad b_2 := \begin{pmatrix} 0 \\ -2^{-1} \\ 0 \\ 0 \\ -2^{-1} \end{pmatrix}$$

$$A_\ell := \begin{pmatrix} 2^{-1} & -1 & 0 & 0 & 0 \\ 2^{-1} & -1 & 0 & 0 & 0 \\ -2^{-1} & 1 & 1 & 2^{-1} & -1 \\ 0 & 0 & 0 & 2^{-1} & -1 \\ 0 & 0 & 0 & 2^{-1} & -1 \end{pmatrix} \in \mathbb{R}^{5\times 5}, \quad b_\ell := \begin{pmatrix} 0 \\ -2^{-2\ell+3} \\ 0 \\ 0 \\ -2^{-2\ell+3} \end{pmatrix}, \quad \text{for } \ell \in \{3, \dots, m+1\},$$

and $A_{m+2} := (-2^{-1}, 1, 1, 2^{-1}, -1) \in \mathbb{R}^{1\times 5}$, $b_{m+2} := 0$. A direct calculation yields

$$\begin{aligned} \tilde{\Psi}_m(x, y) &= \left(\tfrac{|x+y|}{2D} - \sum_{k=0}^{m-1} H_k\big(\tfrac{|x+y|}{2D}\big) \right) - \left(\tfrac{|x-y|}{2D} - \sum_{k=0}^{m-1} H_k\big(\tfrac{|x-y|}{2D}\big) \right) \\ &= \tilde{\Phi}_m\left(\tfrac{|x+y|}{2D}\right) - \tilde{\Phi}_m\left(\tfrac{|x-y|}{2D}\right), \end{aligned} \tag{6}$$

with H_k and $\tilde{\Phi}_m$ as defined in the proof of Proposition III.2. With (4) this implies

$$\begin{aligned} \sup_{(x,y)\in[-D,D]^2} \left| \tilde{\Psi}_m(x, y) - \tfrac{xy}{D^2} \right| &= \sup_{(x,y)\in[-D,D]^2} \left| \left(\tilde{\Phi}_m\left(\tfrac{|x+y|}{2D}\right) - \tilde{\Phi}_m\left(\tfrac{|x-y|}{2D}\right) \right) - \left(\left(\tfrac{|x+y|}{2D}\right)^2 - \left(\tfrac{|x-y|}{2D}\right)^2 \right) \right| \\ &\leq 2 \sup_{z\in[0,1]} |\tilde{\Phi}_m(z) - z^2| \leq 2^{-2m-1}. \end{aligned} \tag{7}$$

Next, let $\Psi_D(x) = D^2 x$ be the scalar multiplication network according to Lemma A.1 and take $\Phi_{D,\varepsilon} := \Psi_D \circ \tilde{\Psi}_{m(D,\varepsilon)}$, where $m(D, \varepsilon) := \lceil 2^{-1}(1 + \log(D^2\varepsilon^{-1})) \rceil$. Then, the error estimate (5) follows directly from (7) and Lemma II.3 establishes the desired bounds on depth, width, and weight magnitude. Finally, $\Phi_{D,\varepsilon}(0, x) = \Phi_{D,\varepsilon}(x, 0) = 0$, for all $x \in \mathbb{R}$, follows directly from (6). □

Remark III.4. *Note that the multiplication network just constructed has weights bounded by* 1 *irrespectively of the size* D *of the domain. This is accomplished by trading network depth for weight magnitude according to Lemma A.1.*

We proceed to the approximation of polynomials, effected by networks that realize linear combinations of monomials, which, in turn, are built by composing multiplication networks. Before presenting the specifics of this construction, we hasten to add that a similar approach was considered previously in [24] and [25]. While there are slight differences in formulation, the main distinction between our construction and those in [24] and [25] resides in their purpose. Specifically, the goal in [24] and [25] is to establish, by way of local Taylor-series approximation, that d-variate, k-times (weakly) differentiable functions can be approximated in L^∞-norm to within error ε with networks of connectivity scaling according to $\varepsilon^{-d/k}\log(\varepsilon^{-1})$. Here, on the other hand, we will be interested in functions that allow approximation with networks of connectivity scaling polylogarithmically in ε^{-1} (i.e., as a polynomial in $\log(\varepsilon^{-1})$). Moreover, for ease of exposition, we will employ finite-width networks. Polylogarithmic connectivity scaling will turn out to be crucial (see Sections VI-IX) in establishing Kolmogorov-Donoho rate-distortion optimality of neural networks in the approximation of a variety of prominent function classes. Finally, we would like to mention related recent work [29], [30], [31] on the approximation of Sobolev-class functions in certain Sobolev norms enabled by neural network approximations of the multiplication operation and of polynomials.

Proposition III.5. *There exists a constant* $C > 0$ *such that for all* $m \in \mathbb{N}$, $a = (a_i)_{i=0}^m \in \mathbb{R}^{m+1}$, $D \in \mathbb{R}_+$, *and* $\varepsilon \in (0, 1/2)$*, there is a network* $\Phi_{a,D,\varepsilon} \in \mathcal{N}_{1,1}$ *with* $\mathcal{L}(\Phi_{a,D,\varepsilon}) \leq Cm(\log(\varepsilon^{-1}) + m\log(\lceil D\rceil) + \log(m) + \log(\lceil \|a\|_\infty \rceil))$, $\mathcal{W}(\Phi_{a,D,\varepsilon}) \leq 9$, $\mathcal{B}(\Phi_{a,D,\varepsilon}) \leq 1$*, and satisfying*

$$\|\Phi_{a,D,\varepsilon}(x) - \sum_{i=0}^m a_i x^i\|_{L^\infty([-D,D])} \leq \varepsilon.$$

Proof. As in the proof of Proposition III.3 and for the same reason, it suffices to consider the case $D \geq 1$. For $m = 1$, we simply have an affine transformation and the statement follows directly from Corollary A.2. The proof for $m \geq 2$ will be effected by realizing the monomials $x^k, k \geq 2$, through iterative composition of multiplication networks and combining this with a construction that uses the network realizing x^k not only as a building block in the network implementing x^{k+1} but also to approximate the partial sum $\sum_{i=0}^k a_i x^i$ in parallel.

We start by setting $B_k = B_k(D, \eta) := \lceil D\rceil^k + \eta \sum_{s=0}^{k-2} \lceil D\rceil^s$, $k \in \mathbb{N}, \eta \in \mathbb{R}_+$ and take $\Phi_{B_k,\eta}$ to be the multiplication network from Proposition III.3. Next, we recursively define the functions

$$f_{k,D,\eta}(x) = \Phi_{B_{k-1},\eta}(x, f_{k-1,D,\eta}(x)), \quad k \geq 2,$$

with $f_{0,D,\eta}(x) = 1$ and $f_{1,D,\eta}(x) = x$. For notational simplicity, we use the abbreviation $f_k = f_{k,D,\eta}$ in the following. First, we verify that the $f_{k,D,\eta}$ approximate monomials sufficiently well. Specifically, we prove by induction that

$$\|f_k(x) - x^k\|_{L^\infty([-D,D])} \leq \eta \sum_{s=0}^{k-2} \lceil D\rceil^s, \tag{8}$$

for all $k \geq 2$. The base case $k = 2$, i.e.,

$$\|f_2(x) - x^2\|_{L^\infty([-D,D])} = \|\Phi_{B_1,\eta}(x,x) - x^2\|_{L^\infty([-D,D])} \leq \eta,$$

follows directly from Proposition III.3 upon noting that $D \leq B_1 = \lceil D \rceil$ (we take the sum in the definition of B_k to equal zero when the upper limit of summation is negative). We proceed to establishing the induction step $(k-1) \to k$ with the induction assumption given by

$$\|f_{k-1}(x) - x^{k-1}\|_{L^\infty([-D,D])} \leq \eta \sum_{s=0}^{k-3} \lceil D \rceil^s.$$

As

$$\|f_{k-1}\|_{L^\infty([-D,D])} \leq \|x^{k-1}\|_{L^\infty([-D,D])} + \|f_{k-1}(x) - x^{k-1}\|_{L^\infty([-D,D])} \leq B_{k-1},$$

application of Proposition III.3 yields

$$\begin{aligned}\|f_k(x) - x^k\|_{L^\infty([-D,D])} &\leq \|f_k(x) - xf_{k-1}(x)\|_{L^\infty([-D,D])} + \|xf_{k-1}(x) - x^k\|_{L^\infty([-D,D])} \\ &\leq \|\Phi_{B_{k-1},\eta}(x, f_{k-1}(x)) - xf_{k-1}(x)\|_{L^\infty([-D,D])} + D\|f_{k-1}(x) - x^{k-1}\|_{L^\infty([-D,D])} \\ &\leq \eta + \lceil D \rceil \eta \sum_{s=0}^{k-3} \lceil D \rceil^s = \eta \sum_{s=0}^{k-2} \lceil D \rceil^s,\end{aligned}$$

which completes the induction.

We now construct the network $\Phi_{a,D,\varepsilon}$ approximating the polynomial $\sum_{i=0}^m a_i x^i$. To this end, note that there exists a constant C' such that for all $m \geq 2$, $a = (a_i)_{i=0}^m \in \mathbb{R}^{m+1}$, and $i \in \{1, \dots, m-1\}$, there is a network $\Psi^i_{a,D,\eta} \in \mathcal{N}_{3,3}$ with $\mathcal{L}(\Psi^i_{a,D,\eta}) \leq C'(\log(\eta^{-1}) + \log(\lceil B_i \rceil) + \log(\|a\|_\infty))$, $\mathcal{W}(\Psi^i_{a,D,\eta}) \leq 9$, $\mathcal{B}(\Psi^i_{a,D,\eta}) \leq 1$, and satisfying

$$\Psi^i_{a,D,\eta}(x, s, y) = (x, s + a_i y, \Phi_{B_i,\eta}(x, y)).$$

To see that this is, indeed, the case, consider the following chain of mappings

$$(x, s, y) \xrightarrow{(I)} (x, s, y, y) \xrightarrow{(II)} (x, s + a_i y, y) \xrightarrow{(III)} (x, s + a_i y, x, y) \xrightarrow{(IV)} (x, s + a_i y, \Phi_{B_i,\eta}(x, y)).$$

Observe that the mapping (I) is an affine transformation with coefficients in $\{0, 1\}$, which we can simply consider to be a depth-1 network. The mapping (II) is obtained by using Corollary A.2 in order to implement the affine transformation $(s, y) \mapsto s + a_i y$ with weights bounded by 1, followed by application of Lemmas II.4 and II.5 to put this network in parallel with two networks realizing the identity mapping according to $x = \rho(x) - \rho(-x)$. Mapping (III) is obtained along the same lines by putting the result of mapping (II) in parallel with another network realizing the identity mapping. Finally, mapping (IV) is realized by putting the network $\Phi_{B_i,\eta}$ in parallel with two identity networks. Composing these four networks according to Lemma II.3 yields, for $i \in \{1, \dots, m-1\}$, a network $\Psi^i_{a,D,\eta}$ with the claimed properties. Next, we employ Corollary A.2 to get networks $\Psi^0_{a,D,\eta}$ which implement $x \mapsto (x, a_0, x)$

as well as networks $\Psi^m_{a,D,\eta}$ realizing $(x,s,y) \mapsto s + a_m y$. Let now $\eta = \eta(a,D,\varepsilon) := (\|a\|_\infty (m-1)^2 \lceil D \rceil^{m-2})^{-1}\varepsilon$ and define

$$\Phi_{a,D,\varepsilon} := \Psi^m_{a,D,\eta} \circ \Psi^{m-1}_{a,D,\eta} \circ \cdots \circ \Psi^1_{a,D,\eta} \circ \Psi^0_{a,D,\eta}.$$

A direct calculation yields

$$\Phi_{a,D,\varepsilon} = \sum_{i=0}^{m} a_i f_{i,D,\eta}.$$

Hence (8) implies

$$\begin{aligned}\Big\|\Phi_{a,D,\varepsilon}(x) - \sum_{i=0}^{m} a_i x^i\Big\|_{L^\infty([-D,D])} &\le \sum_{i=0}^{m} |a_i| \|f_{i,D,\eta}(x) - x^i\|_{L^\infty([-D,D])} \le \sum_{i=2}^{m} |a_i| \Big(\eta \sum_{s=0}^{i-2} \lceil D \rceil^s\Big) \\ &\le \|a\|_\infty \eta \sum_{k=0}^{m-2} (m-1-k) \lceil D \rceil^k \le \|a\|_\infty (m-1)^2 \lceil D \rceil^{m-2} \eta = \varepsilon.\end{aligned}$$

Lemma II.3 now establishes that $\mathcal{W}(\Phi_{a,D,\varepsilon}) \le 9$, $\mathcal{B}(\Phi_{a,D,\varepsilon}) \le 1$, and

$$\begin{aligned}\mathcal{L}(\Phi_{a,D,\varepsilon}) &\le \sum_{i=0}^{m} \mathcal{L}(\Psi^i_{a,D,\eta}) \\ &\le 2(\log(\lceil \|a\|_\infty \rceil) + 5) + \sum_{i=1}^{m-1} C'(\log(\eta^{-1}) + \log(\lceil B_{i-1} \rceil) + \log(\lceil \|a\|_\infty \rceil)) \\ &\le Cm(\log(\varepsilon^{-1}) + m\log(\lceil D \rceil) + \log(m) + \log(\lceil \|a\|_\infty \rceil))\end{aligned}$$

for a suitably chosen absolute constant C. This completes the proof. □

Next, we recall that the Weierstrass approximation theorem states that every continuous function on a closed interval can be approximated to within arbitrary accuracy by a polynomial.

Theorem III.6 ([32])**.** *Let $[a,b] \subseteq \mathbb{R}$ and $f \in C([a,b])$. Then, for every $\varepsilon > 0$, there exists a polynomial π such that*

$$\|f - \pi\|_{L^\infty([a,b])} \le \varepsilon.$$

Proposition III.5 hence allows us to conclude that every continuous function on a closed interval can be approximated to within arbitrary accuracy by a deep ReLU network of width no more than 9. This amounts to a variant of the universal approximation theorem [11], [12] for finite-width deep ReLU networks. A quantitative statement in terms of making the approximating network's width, depth, and weight bounds explicit can be obtained for (very) smooth functions by applying Proposition III.5 to Lagrangian interpolation with Chebyshev points.

Lemma III.7. *Consider the set*

$$\mathcal{S}_{[-1,1]} := \left\{ f \in C^\infty([-1,1],\mathbb{R})\colon \|f^{(n)}(x)\|_{L^\infty([-1,1])} \le n!, \;\text{ for all } n \in \mathbb{N}_0 \right\}.$$

There exists a constant $C > 0$ such that for all $f \in \mathcal{S}_{[-1,1]}$ and $\varepsilon \in (0,1/2)$, there is a network $\Psi_{f,\varepsilon} \in \mathcal{N}_{1,1}$ with $\mathcal{L}(\Psi_{f,\varepsilon}) \le C(\log(\varepsilon^{-1}))^2$, $\mathcal{W}(\Psi_{f,\varepsilon}) \le 9$, $\mathcal{B}(\Psi_{f,\varepsilon}) \le 1$, and satisfying

$$\|\Psi_{f,\varepsilon} - f\|_{L^\infty([-1,1])} \le \varepsilon.$$

Proof. A fundamental result on Lagrangian interpolation with Chebyshev points (see e.g. [33, Lemma 3]) guarantees, for all $f \in \mathcal{S}_{[-1,1]}$, $m \in \mathbb{N}$, the existence of a polynomial $P_{f,m}$ of degree m such that

$$\|f - P_{f,m}\|_{L^\infty([-1,1])} \le \tfrac{1}{2^m(m+1)!}\|f^{(m+1)}\|_{L^\infty([-1,1])} \le \tfrac{1}{2^m}.$$

Note that $P_{f,m}$ can be expressed in the Chebyshev basis (see e.g. [34, Section 3.4.1]) according to $P_{f,m} = \sum_{j=0}^m c_{f,m,j}T_j(x)$ with $|c_{f,m,j}| \le 2$ and the Chebyshev polynomials defined through the two-term recursion $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, $k \ge 2$, with $T_0(x) = 1$ and $T_1(x) = x$. We can moreover use this recursion to conclude that the coefficients of the T_k in the monomial basis are upper-bounded by 3^k. Consequently, we can express $P_{f,m}$ according to $P_{f,m} = \sum_{j=0}^m a_{f,m,j}x^j$ with

$$A_{f,m} := \max_{j=0,\dots,m} |a_{f,m,j}| \le 2(m+1)3^m.$$

Application of Proposition III.5 to $P_{f,m}$ in the monomial basis, with $m = \lceil \log(2/\varepsilon) \rceil$ and approximation error $\varepsilon/2$, completes the proof upon noting that

$$C'm(\log(2/\varepsilon) + \log(m) + \log(|A_{f,m}|)) \le C(\log(\varepsilon^{-1}))^2$$

for some absolute constant C. □

An extension of Lemma III.7 to approximation over general intervals is provided in Lemma A.6.

While Lemma III.7 shows that a specific class of C^∞-functions, namely those whose derivatives are suitably bounded, can be approximated by neural networks with connectivity growing polylogarithmically in ε^{-1}, it turns out that this is not possible for general (Sobolev-class) k-times differentiable functions [24, Thm.4].

We are now ready to proceed to the approximation of sinusoidal functions. Before stating the corresponding result, we comment on the basic idea enabling the approximation of oscillatory functions through deep neural networks. In essence, we exploit the optimality of the sawtooth construction (3) in terms of achieving exponential—in network depth—growth in the number of linear regions. As indicated in Figure 3, the composition of the cosine function (realized according to Lemma III.7) with the sawtooth function, combined with the symmetry properties of the cosine function and the sawtooth function, yields oscillatory behavior that increases exponentially with network depth.

Theorem III.8. *There exists a constant $C > 0$ such that for every $a, D \in \mathbb{R}_+$, $\varepsilon \in (0, 1/2)$, there is a network $\Psi_{a,D,\varepsilon} \in \mathcal{N}_{1,1}$ with $\mathcal{L}(\Psi_{a,D,\varepsilon}) \leq C((\log(\varepsilon^{-1}))^2 + \log(\lceil aD \rceil))$, $\mathcal{W}(\Psi_{a,D,\varepsilon}) \leq 9$, $\mathcal{B}(\Psi_{a,D,\varepsilon}) \leq 1$, and satisfying*

$$\|\Psi_{a,D,\varepsilon}(x) - \cos(ax)\|_{L^\infty([-D,D])} \leq \varepsilon.$$

Proof. Note that $f(x) := (6/\pi^3)\cos(\pi x)$ is in $\mathcal{S}_{[-1,1]}$. Thus, by Lemma III.7, there exists a constant $C > 0$ such that for every $\varepsilon \in (0, 1/2)$, there is a network $\Phi_\varepsilon \in \mathcal{N}_{1,1}$ with $\mathcal{L}(\Phi_\varepsilon) \leq C(\log(\varepsilon^{-1}))^2$, $\mathcal{W}(\Phi_\varepsilon) \leq 9$, $\mathcal{B}(\Phi_\varepsilon) \leq 1$, and satisfying

$$\|\Phi_\varepsilon - f\|_{L^\infty([-1,1])} \leq \tfrac{6}{\pi^3}\varepsilon. \tag{9}$$

We now extend this result to the approximation of $x \mapsto \cos(ax)$ on the interval $[-1, 1]$ for arbitrary $a \in \mathbb{R}_+$. This will be accomplished by exploiting that $x \mapsto \cos(\pi x)$ is 2-periodic and even. Let $g_s \colon [0, 1] \to [0, 1]$, $s \in \mathbb{N}$, be the s-th order sawtooth functions as defined in (2) and note that, due to the periodicity and the symmetry of the cosine function (see Figure 3 for illustration), we have for all $s \in \mathbb{N}_0$, $x \in [-1, 1]$,

$$\cos(\pi 2^s x) = \cos(\pi g_s(|x|)).$$

For $a > \pi$, we define $s = s(a) := \lceil \log(a) - \log(\pi) \rceil$ and $\alpha = \alpha(a) := (\pi 2^s)^{-1} a \in (1/2, 1]$, and note that

$$\cos(ax) = \cos(\pi 2^s \alpha x) = \cos(\pi g_s(\alpha|x|)), \quad x \in [-1, 1].$$

As $g_s(\alpha|x|) \in [0, 1]$, it follows from (9) that

$$\|\tfrac{\pi^3}{6}\Phi_\varepsilon(g_s(\alpha|x|)) - \cos(ax)\|_{L^\infty([-1,1])} = \tfrac{\pi^3}{6}\|\Phi_\varepsilon(g_s(\alpha|x|)) - f(g_s(\alpha|x|))\|_{L^\infty([-1,1])} \leq \varepsilon. \tag{10}$$

In order to realize $\Phi_\varepsilon(g_s(\alpha|x|))$ as a neural network, we start from the networks Φ_g^s defined in (3) and apply Proposition A.3 to convert them into networks $\Psi_g^s(x) = g_s(x)$, for $x \in [0, 1]$, with $\mathcal{B}(\Psi_g^s) \leq 1$, $\mathcal{L}(\Psi_g^s) = 7(s + 1)$, and $\mathcal{W}(\Psi_g^s) = 3$. Furthermore, let $\Psi(x) := \alpha\rho(x) - \alpha\rho(-x) = \alpha|x|$ and take $\Phi^{\text{mult}}_{\pi^3/6}$ to be the scalar multiplication network from Lemma A.1. Noting that $\Psi_{a,\varepsilon} := \Phi^{\text{mult}}_{\pi^3/6} \circ \Phi_\varepsilon \circ \Psi_g^s \circ \Psi = \Phi_\varepsilon(g_s(\alpha|x|))$ and concluding from Lemma II.3 that $\mathcal{L}(\Psi_{a,\varepsilon}) \leq C((\log(\varepsilon^{-1}))^2 + \log(\lceil a \rceil))$, $\mathcal{W}(\Psi_{a,\varepsilon}) \leq 9$, and $\mathcal{B}(\Psi_{a,\varepsilon}) \leq 1$, together with (10), establishes the desired result for $a > \pi$ and for approximation over the interval $[-1, 1]$. For $a \in (0, \pi)$, we can simply take $\Psi_{a,\varepsilon} := \Phi^{\text{mult}}_{\pi^3/6} \circ \Phi_\varepsilon$ as $x \mapsto (6/\pi^3)\cos(ax)$ is in $\mathcal{S}_{[-1,1]}$ in this case.

Finally, we consider the approximation of $x \mapsto \cos(ax)$ on intervals $[-D, D]$, for arbitrary $D \geq 1$. To this end, we define the networks $\Psi_{a,D,\varepsilon}(x) := \Psi_{aD,\varepsilon}(\frac{x}{D})$ and observe that

$$\begin{aligned} \sup_{x \in [-D,D]} |\Psi_{a,D,\varepsilon}(x) - \cos(ax)| &= \sup_{y \in [-1,1]} |\Psi_{a,D,\varepsilon}(Dy) - \cos(aDy)| \\ &= \sup_{y \in [-1,1]} |\Psi_{aD,\varepsilon}(y) - \cos(aDy)| \leq \varepsilon. \end{aligned} \tag{11}$$

This concludes the proof. □

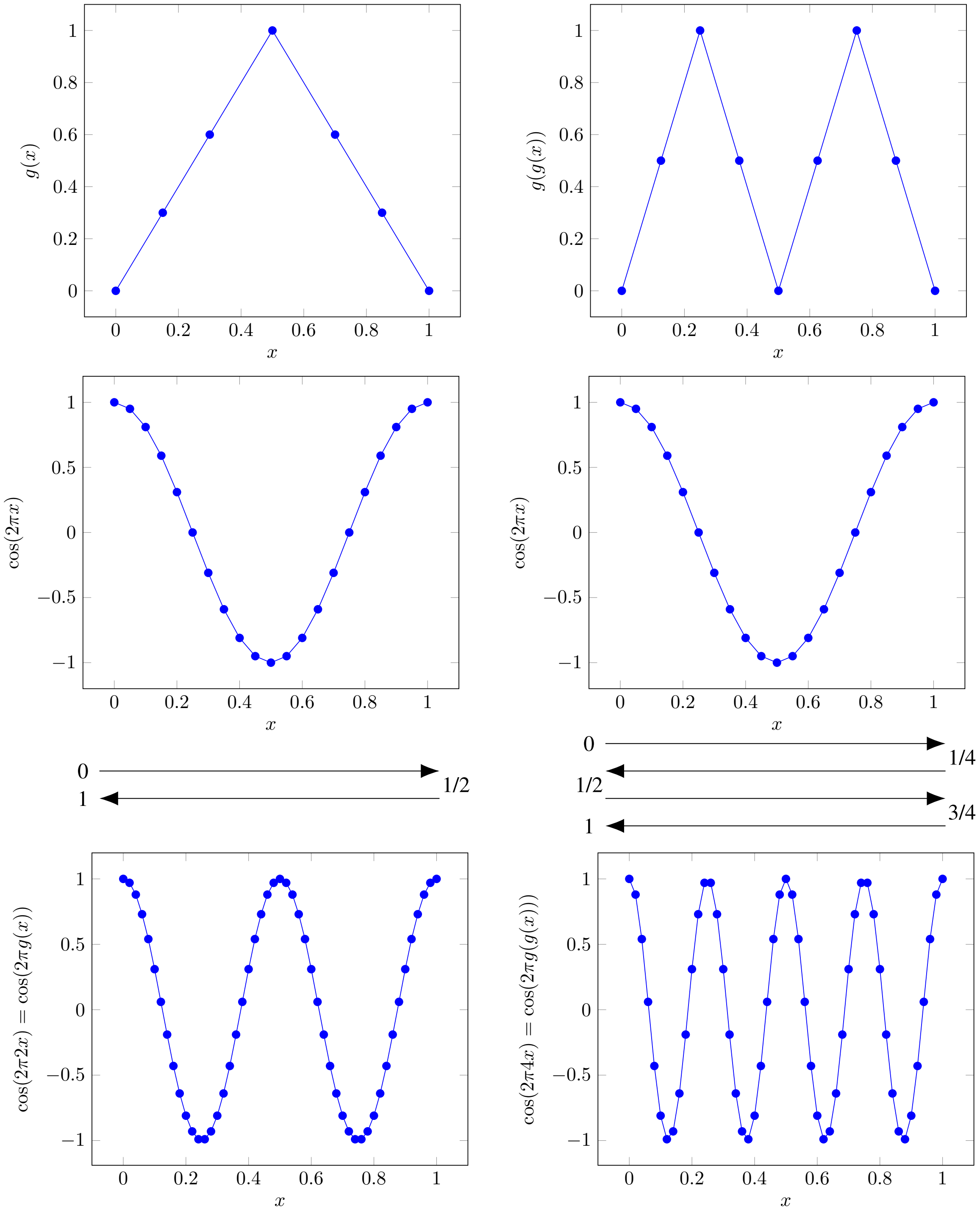


Fig. 3: Approximation of the function $\cos(2\pi ax)$ according to Theorem III.8 using "sawtooth" functions $g_s(x)$ as per (2), left $a = 2$, right $a = 4$.

The result just obtained extends to the approximation of $x \mapsto \sin(ax)$, formalized next, simply by noting that $\sin(x) = \cos(x - \pi/2)$.

Corollary III.9. *There exists a constant $C > 0$ such that for every $a, D \in \mathbb{R}_+$, $b \in \mathbb{R}$, $\varepsilon \in (0, 1/2)$, there is a network $\Psi_{a,b,D,\varepsilon} \in \mathcal{N}_{1,1}$ with $\mathcal{L}(\Psi_{a,b,D,\varepsilon}) \leq C((\log(\varepsilon^{-1}))^2 + \log(\lceil aD + |b| \rceil))$, $\mathcal{W}(\Psi_{a,b,D,\varepsilon}) \leq 9$, $\mathcal{B}(\Psi_{a,b,D,\varepsilon}) \leq 1$, and satisfying*

$$\|\Psi_{a,b,D,\varepsilon}(x) - \cos(ax - b)\|_{L^\infty([-D,D])} \leq \varepsilon.$$

Proof. For given $a, D \in \mathbb{R}_+$, $b \in \mathbb{R}$, $\varepsilon \in (0, 1/2)$, consider the network $\Psi_{a,b,D,\varepsilon}(x) := \Psi_{a,D+\frac{|b|}{a},\varepsilon}\left(x - \frac{b}{a}\right)$ with $\Psi_{a,D,\varepsilon}$ as defined in the proof of Theorem III.8, and observe that, owing to (11),

$$\sup_{x \in [-D,D]} |\Psi_{a,b,D,\varepsilon}(x) - \cos(ax - b)| \leq \sup_{y \in \left[-(D+\frac{|b|}{a}), D+\frac{|b|}{a}\right]} |\Psi_{a,D+\frac{|b|}{a},\varepsilon}(y) - \cos(ay)| \leq \varepsilon.$$

□

Remark III.10. *The results in this section all have approximating networks of finite width and depth scaling polylogarithmically in ε^{-1}. Owing to*

$$\mathcal{M}(\Phi) \leq \mathcal{L}(\Phi)\mathcal{W}(\Phi)(\mathcal{W}(\Phi) + 1)$$

this implies that the connectivity scales no faster than polylogarithmic in ε^{-1}. It therefore follows that the approximation error ε decays (at least) exponentially fast in the connectivity or equivalently in the number of parameters the approximant (i.e., the neural network) employs. We say that the network provides exponential approximation accuracy.

IV. Approximation of Function Classes and Metric Entropy

So far we considered the explicit construction of deep neural networks for the approximation of a wide range of functions, namely polynomials, smooth functions, and sinusoidal functions, in all cases with exponential accuracy, i.e., with an approximation error that decays exponentially in network connectivity. We now proceed to lay the foundation for the development of a framework that allows us to characterize the fundamental limits of deep neural network approximation of entire function classes. But first, we provide a review of relevant literature.

The best-known results on approximation by neural networks are the universal approximation theorems of Hornik [12] and Cybenko [11], stating that continuous functions on bounded domains can be approximated arbitrarily well by a single-hidden-layer ($L = 2$ in our terminology) neural network with sigmoidal activation function. The literature on approximation-theoretic properties of networks with a single hidden layer continuing this line of work is abundant. Without any claim to completeness, we mention work on approximation error bounds in terms of the number of neurons for functions with Fourier transforms of bounded first moments [35], [36], the nonexistence of localized approximations [37], a fundamental lower bound on approximation rates [38], [39], and the approximation of smooth or analytic functions [40], [41].

Approximation-theoretic results for networks with multiple hidden layers were obtained in [42], [43] for general functions, in [44] for continuous functions, and for functions together with their derivatives in [45]. In [37] it was shown that for certain approximation tasks deep networks can perform fundamentally better than single-hidden-layer networks. We also highlight two recent papers, which investigate the benefit—from an approximation-theoretic perspective—of multiple hidden layers. Specifically, in [46] it was shown that there exists a function which, although expressible through a small three-layer network, can only be represented through a very large two-layer network; here size is measured in terms of the total number of neurons in the network.

In the setting of deep convolutional neural networks first results of a nature similar to those in [46] were reported in [47]. Linking the expressivity properties of neural networks to tensor decompositions, [48], [49] established the existence of functions that can be realized by relatively small deep convolutional networks but require exponentially larger shallow convolutional networks.

We conclude by mentioning recent results bearing witness to the approximation power of deep ReLU networks in the context of PDEs. Specifically, it was shown in [29] that deep ReLU networks can approximate very effectively certain solution families of parametric PDEs depending on a large (possibly infinite) number of parameters. The series of papers [50], [51], [52], [53] constructs and analyzes a deep-learning-based numerical solver for Black-Scholes PDEs.

For survey articles on approximation-theoretic aspects of neural networks, we refer the interested reader to [54] and [55] as well as the very recent [56]. Most closely related to the framework we develop here is the paper by Shaham, Cloninger, and Coifman [57], which shows that for functions that are sparse in specific wavelet frames, the best M-weight approximation rate (see Definition VI.1 below) of three-layer neural networks is at least as large as the best M-term approximation rate in piecewise linear wavelet frames.

We begin the development of our framework with a review of a widely used theoretical foundation for deterministic lossy data compression [58], [59]. Our presentation essentially follows [14], [60].

A. Kolmogorov-Donoho Rate Distortion Theory

Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and consider a set of functions $\mathcal{C} \subseteq L^2(\Omega)$, which we will frequently refer to as *function class*. Then, for each $\ell \in \mathbb{N}$, we denote by

$$\mathfrak{E}^\ell := \left\{E : \mathcal{C} \to \{0,1\}^\ell\right\}$$

the set of *binary encoders of $\mathcal{C}$ of length ℓ*, and we let

$$\mathfrak{D}^\ell := \left\{D : \{0,1\}^\ell \to L^2(\Omega)\right\}$$

be the set of *binary decoders of length ℓ*. An encoder-decoder pair $(E, D) \in \mathfrak{E}^\ell \times \mathfrak{D}^\ell$ is said to *achieve uniform error ε over the function class $\mathcal{C}$*, if

$$\sup_{f\in\mathcal{C}} \|D(E(f)) - f\|_{L^2(\Omega)} \leq \varepsilon.$$

Note that here we quantified the approximation error in $L^2(\Omega)$-norm, whereas in the previous section we used the $L^\infty(\Omega)$-norm. While results in terms of $L^\infty(\Omega)$-norm are stronger, we shall employ the $L^2(\Omega)$-norm in order to parallel the Kolmogorov-Donoho framework for nonlinear approximation through dictionaries [14], [15]. We furthermore note that for sets Ω of finite Lebesgue measure $|\Omega|$, the two norms are related through $\|f\|_{L^2(\Omega)} \leq |\Omega|^{1/2}\|f\|_{L^\infty(\Omega)}$. Finally, whenever we talk about compactness and related topological notions, we shall always mean w.r.t. the topology induced by the $L^2(\Omega)$-norm.

A quantity of central interest is the minimal length $\ell \in \mathbb{N}$ for which there exists an encoder-decoder pair $(E, D) \in \mathfrak{E}^\ell \times \mathfrak{D}^\ell$ that achieves uniform error ε over the function class $\mathcal{C}$, along with its asymptotic behavior as made precise in the following definition.

Definition IV.1. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and let $\mathcal{C} \subseteq L^2(\Omega)$ be compact. Then, for $\varepsilon > 0$, the* minimax code length *$L(\varepsilon, \mathcal{C})$ is*

$$L(\varepsilon, \mathcal{C}) := \min\left\{\ell \in \mathbb{N} : \exists(E, D) \in \mathfrak{E}^\ell \times \mathfrak{D}^\ell : \sup_{f\in\mathcal{C}} \|D(E(f)) - f\|_{L^2(\Omega)} \leq \varepsilon\right\}. \tag{12}$$

Moreover, the optimal exponent *$\gamma^*(\mathcal{C})$ is defined as*

$$\gamma^*(\mathcal{C}) := \sup\left\{\gamma \in \mathbb{R} : L(\varepsilon, \mathcal{C}) \in \mathcal{O}\left(\varepsilon^{-1/\gamma}\right), \varepsilon \to 0\right\}.$$

The optimal exponent $\gamma^*(\mathcal{C})$ determines the minimum growth rate of $L(\varepsilon, \mathcal{C})$ as the error ε tends to zero and can hence be seen as quantifying the "description complexity" of the function class $\mathcal{C}$. Larger $\gamma^*(\mathcal{C})$ results in smaller growth rate and hence smaller memory requirements for storing functions $f \in \mathcal{C}$ such that reconstruction with uniformly bounded error is possible.

Remark IV.2. *The optimal exponent $\gamma^*(\mathcal{C})$ can equivalently be thought of as quantifying the asymptotic behavior of the minimal achievable error for the function class $\mathcal{C}$ with a given code length. Specifically, we have*

$$\gamma^*(\mathcal{C}) = \sup\big\{\gamma \in \mathbb{R} : L(\varepsilon, \mathcal{C}) \in \mathcal{O}\big(\varepsilon^{-1/\gamma}\big), \varepsilon \to 0\big\} = \sup\big\{\gamma \in \mathbb{R} : \varepsilon(L) \in \mathcal{O}\big(L^{-\gamma}\big), L \to \infty\big\}, \tag{13}$$

where

$$\varepsilon(L) := \inf_{(E,D)\in\mathfrak{E}^L\times\mathfrak{D}^L} \sup_{f\in\mathcal{C}} \|D(E(f)) - f\|_{L^2(\Omega)}.$$

The quantity $\gamma^*(\mathcal{C})$ is closely related to the concept of Kolmogorov-Tikhomirov epsilon entropy a.k.a. metric entropy [61]. We next make this connection explicit.

B. Metric entropy

Most of the discussion in this subsection, which is almost exclusively of review nature, follows very closely [62, Chapter 5]. Consider the metric space $(\mathcal{X}, \rho)$ with $\mathcal{X}$ a nonempty set and $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a distance function. A natural measure for the size of a compact subset $\mathcal{C}$ of $\mathcal{X}$ is given by the number of balls of a fixed radius ε required to cover $\mathcal{C}$, a quantity known as the covering number (for covering radius ε).

Definition IV.3. *[62] Let $(\mathcal{X}, \rho)$ be a metric space. An ε-covering of a compact set $\mathcal{C} \subseteq \mathcal{X}$ with respect to the metric ρ is a set $\{x_1, \dots, x_N\} \subseteq \mathcal{C}$ such that for each $x \in \mathcal{C}$, there exists an $i \in \{1, \dots, N\}$ so that $\rho(x, x_i) \leq \varepsilon$. The ε-covering number $N(\varepsilon; \mathcal{C}, \rho)$ is the cardinality of the smallest ε-covering.*

An ε-covering is a collection of balls of radius ε that cover the set $\mathcal{C}$, i.e.,

$$\mathcal{C} \subseteq \bigcup_{i=1}^{N} B(x_i, \varepsilon),$$

where $B(x_i, \varepsilon)$ is a ball—in the metric ρ—of radius ε centered at x_i. The covering number is nonincreasing in ε, i.e., $N(\varepsilon) \geq N(\varepsilon')$, for all $\varepsilon \leq \varepsilon'$. When the set $\mathcal{C}$ is not finite, the covering number goes to infinity as ε goes to zero. We shall be interested in the corresponding rate of growth, more specifically in the quantity $\log N(\varepsilon; \mathcal{C}, \rho)$ known as the metric entropy of $\mathcal{C}$ with respect to ρ. Recall that $\log$ is to the base 2, hence the unit of metric entropy is “bits”. The operational significance of metric entropy follows from the question: What is the minimum number of bits needed to represent any element $x \in \mathcal{C}$ with error—quantified in terms of the distance measure ρ—of at most ε? By what was just developed, the answer to this question is $\lceil \log N(\varepsilon; \mathcal{C}, \rho) \rceil$. Specifically, for a given $x \in \mathcal{X}$, the corresponding encoder $E(x)$ simply identifies the closest ball center x_i and encodes the index i using $\lceil \log N(\varepsilon; \mathcal{C}, \rho) \rceil$ bits. The corresponding decoder D delivers the ball center x_i, which guarantees that the resulting error satisfies $\|D(E(x)) - x\| \leq \varepsilon$.

We proceed with a simple example ([62, Example 5.2]) computing an upper bound on the metric entropy of the interval $\mathcal{C} = [-1, 1]$ in $\mathbb{R}$ with respect to the metric $\rho(x, x') = |x - x'|$. To this end, we divide $\mathcal{C}$ into intervals of length 2ε by setting $x_i = -1 + 2(i-1)\varepsilon$, for $i \in [1, L]$, where $L = \lfloor \frac{1}{\varepsilon} \rfloor + 1$. This guarantees that, for every point $x \in [-1, 1]$, there is an $i \in [1, L]$ such that $|x - x_i| \leq \varepsilon$, which, in turn, establishes

$$N(\varepsilon; \mathcal{C}, \rho) \leq \left\lfloor \frac{1}{\varepsilon} \right\rfloor + 1 \leq \frac{1}{\varepsilon} + 1$$

and hence yields an upper bound on metric entropy according to[2]

$$\log N(\varepsilon; \mathcal{C}, \rho) \leq \log \left(\frac{1}{\varepsilon} + 1 \right) \asymp \log(\varepsilon^{-1}), \quad \text{as } \varepsilon \to 0. \tag{14}$$

[2] The notation $f(\varepsilon) \asymp g(\varepsilon)$, as $\varepsilon \to 0$, means that there are constants $c, C, \varepsilon_0 > 0$ such that $cf(\varepsilon) \leq g(\varepsilon) \leq Cf(\varepsilon)$, for all $\varepsilon \leq \varepsilon_0$. For ease of exposition, we shall usually omit the qualifier $\varepsilon \to 0$.

This result can be generalized to the d-dimensional unit cube to yield $\log(N(\varepsilon;\mathcal{C},\rho)) \leq d\log(1/\varepsilon{+}1) \asymp d\log(\varepsilon^{-1})$. In order to show that the upper bound (14) correctly reflects metric entropy scaling for $\mathcal{C} = [-1,1]$ with respect to $\rho(x,x') = |x - x'|$, we would need a lower bound on $N(\varepsilon;\mathcal{C},\rho)$ that exhibits the same scaling (in ε) behavior. A systematic approach to establishing lower bounds on metric entropy is through the concept of packing, which will be introduced next.

We start with the definition of the packing number of a compact set $\mathcal{C}$ in a metric space $(\mathcal{X},\rho)$.

Definition IV.4. *[62, Definition 5.4] Let $(\mathcal{X},\rho)$ be a metric space. An ε-packing of a compact set $\mathcal{C} \subseteq \mathcal{X}$ with respect to the metric ρ is a set $\{x_1,\dots,x_N\} \subseteq \mathcal{C}$ such that $\rho(x_i,x_j) > \varepsilon$, for all distinct i,j. The ε-packing number $M(\varepsilon;\mathcal{X},\rho)$ is the cardinality of the largest ε-packing.*

An ε-packing is a collection of nonintersecting balls of radius $\varepsilon/2$ and centered at elements in $\mathcal{X}$. Although different, the covering number and the packing number provide essentially the same measure of size of a set as formalized next.

Lemma IV.5. *[62, Lemma 5.5] Let $(\mathcal{X},\rho)$ be a metric space and $\mathcal{C}$ a compact set in $\mathcal{X}$. For all $\varepsilon > 0$, the packing and the covering number are related according to*

$$M(2\varepsilon;\mathcal{C},\rho) \leq N(\varepsilon;\mathcal{C},\rho) \leq M(\varepsilon;\mathcal{C},\rho).$$

Proof. [62], [63] First, choose a minimal ε-covering and a maximal 2ε-packing of $\mathcal{C}$. Since no two centers of the 2ε-packing can lie in the same ball of the ε-covering, it follows that $M(2\varepsilon;\mathcal{C},\rho) \leq N(\varepsilon;\mathcal{C},\rho)$. To establish $N(\varepsilon;\mathcal{C},\rho) \leq M(\varepsilon;\mathcal{C},\rho)$, we note that, given a maximal packing $M(\varepsilon;\mathcal{C},\rho)$, for any $x \in \mathcal{C}$, we have the center of at least one of the balls in the packing within distance less than ε. If this were not the case, we could add another ball to the packing thereby violating its maximality. This maximal packing hence also provides an ε-covering and since $N(\varepsilon;\mathcal{C},\rho)$ is a minimal covering, we must have $N(\varepsilon;\mathcal{C},\rho) \leq M(\varepsilon;\mathcal{C},\rho)$. □

We now return to the example in which we computed an upper bound on the metric entropy of $\mathcal{C} = [-1,1]$ with respect to $\rho(x,x') = |x - x'|$ and show how Lemma IV.5 can be employed to establish the scaling behavior of metric entropy. To this end, we simply note that the points $x_i = -1 + 2(i-1)\varepsilon$, $i \in [1,L]$, are separated according to $|x_i - x_j| = 2\varepsilon > \varepsilon$, for all $i \neq j$, which implies that $M(\varepsilon;\mathcal{C},|\cdot|) \geq L = \lfloor 1/\varepsilon \rfloor + 1 \geq \frac{1}{\varepsilon}$. Combining this with the upper bound (14) and Lemma IV.5, we obtain $\log N(\varepsilon;\mathcal{C},|\cdot|) \asymp \log(\varepsilon^{-1})$. Likewise, it can be established that $\log N(\varepsilon;\mathcal{C},\|\cdot\|) \asymp d\log(\varepsilon^{-1})$ for the d-dimensional unit cube. This illustrates how an explicit construction of a packing set can be used to determine the scaling behavior of metric entropy.

We next formalize the notion that metric entropy is determined by the volume of the corresponding covering balls. Specifically, the following result establishes a relationship between a certain volume ratio and metric entropy.

Lemma IV.6. *[62, Lemma 5.7] Consider a pair of norms $\|\cdot\|$ and $\|\cdot\|'$ on $\mathbb{R}^d$, and let $\mathcal{B}$ and $\mathcal{B}'$ be their corresponding unit balls, i.e., $\mathcal{B} = \{x \in \mathbb{R}^d \| \|x\| \leq 1\}$ and $\mathcal{B}' = \{x \in \mathbb{R}^d \| \|x\|' \leq 1\}$. Then, the ε-covering number of $\mathcal{B}$ in the $\|\cdot\|'$-norm satisfies*

$$\left(\frac{1}{\varepsilon}\right)^d \frac{vol(\mathcal{B})}{vol(\mathcal{B}')} \leq N(\varepsilon; \mathcal{B}, \|\cdot\|') \leq \frac{vol(\frac{2}{\varepsilon}\mathcal{B} + \mathcal{B}')}{vol(\mathcal{B}')}. \tag{15}$$

Proof. [62] Let $\{x_1, \ldots, x_{N(\varepsilon;\mathcal{B},\|\cdot\|')}\}$ be an ε-covering of $\mathcal{B}$ in $\|\cdot\|'$-norm. Then, we have

$$\mathcal{B} \subseteq \bigcup_{j=1}^{N(\varepsilon;\mathcal{B},\|\cdot\|')} \{x_j + \varepsilon\mathcal{B}'\},$$

which implies $vol(\mathcal{B}) \leq N(\varepsilon; \mathcal{B}, \|\cdot\|')\, \varepsilon^d\, vol(\mathcal{B}')$, thus establishing the lower bound in (15). The upper bound is obtained by starting with a maximal ε-packing $\{x_1, \ldots, x_{M(\varepsilon;\mathcal{B},\|\cdot\|')}\}$ of $\mathcal{B}$ in the $\|\cdot\|'$-norm. The balls $\{x_j + \frac{\varepsilon}{2}\mathcal{B}', j = 1, \ldots, M(\varepsilon; \mathcal{B}, \|\cdot\|')\}$ are all disjoint and contained within $\mathcal{B} + \frac{\varepsilon}{2}\mathcal{B}'$. We can therefore conclude that

$$\sum_{j=1}^{M(\varepsilon;\mathcal{B},\|\cdot\|')} vol\Big(x_j + \frac{\varepsilon}{2}\mathcal{B}'\Big) \leq vol\Big(\mathcal{B} + \frac{\varepsilon}{2}\mathcal{B}'\Big),$$

and hence

$$M(\varepsilon; \mathcal{B}, \|\cdot\|')\, vol\Big(\frac{\varepsilon}{2}\mathcal{B}'\Big) \leq vol\Big(\mathcal{B} + \frac{\varepsilon}{2}\mathcal{B}'\Big).$$

Finally, we have $vol(\frac{\varepsilon}{2}\mathcal{B}') = (\frac{\varepsilon}{2})^d vol(\mathcal{B}')$ and $vol(\mathcal{B}+\frac{\varepsilon}{2}\mathcal{B}') = (\frac{\varepsilon}{2})^d vol(\frac{2}{\varepsilon}\mathcal{B}+\mathcal{B}')$, which, together with $M(\varepsilon; \mathcal{B}, \|\cdot\|')$ $\geq N(\varepsilon; \mathcal{B}, \|\cdot\|')$ due to Lemma IV.5, yields the upper bound in (15). □

This result now allows us to establish the scaling of the metric entropy of unit balls in terms of their own norm, thus yielding a measure of the massiveness of unit balls in d-dimensional spaces. Specifically, we set $\mathcal{B}' = \mathcal{B}$ in Lemma IV.6 and get

$$vol\left(\frac{2}{\varepsilon}\mathcal{B} + \mathcal{B}'\right) = vol\left(\left(\frac{2}{\varepsilon} + 1\right)\mathcal{B}\right) = \left(\frac{2}{\varepsilon} + 1\right)^d vol(\mathcal{B}),$$

which when used in (15) yields $N(\varepsilon; \mathcal{B}, \|\cdot\|) \asymp \varepsilon^{-d}$ and hence results in metric entropy scaling according to $\log(N(\varepsilon; \mathcal{B}, \|\cdot\|)) \asymp d\log(\varepsilon^{-1})$. Particularizing this result to the unit ball $\mathcal{B}^d_\infty = [-1,1]^d$ and the metric $\|\cdot\|_\infty$, we recover the result of our direct analysis in the example above.

So far we have been concerned with the metric entropy of subsets of $\mathbb{R}^d$. We now proceed to analyzing the metric entropy of function classes, which will eventually allow us to establish the desired connection between the optimal exponent $\gamma^*(\mathcal{C})$ and metric entropy. We begin with the simple one-parameter function class considered in [62, Example 5.9] and follow closely the exposition in [62]. For a fixed θ, define the real-valued function $f_\theta(x) = 1 - e^{-\theta x}$, and consider the class

$$\mathcal{P} = \{f_\theta : [0,1] \to \mathbb{R} \,|\, \theta \in [0,1]\}.$$

The set $\mathcal{P}$ constitutes a metric space under the sup-norm given by $\|f-g\|_{L^\infty([0,1])} = \sup_{x\in[0,1]} |f(x)-g(x)|$. We show that the covering number of $\mathcal{P}$ satisfies

$$1 + \left\lfloor \frac{1-1/e}{2\varepsilon} \right\rfloor \leq N(\varepsilon; \mathcal{P}, \|\cdot\|_{L^\infty([0,1])}) \leq \frac{1}{2\varepsilon} + 2,$$

which leads to the scaling behavior $N(\varepsilon; \mathcal{P}, \|\cdot\|_{L^\infty([0,1])}) \asymp \varepsilon^{-1}$ and hence to metric entropy scaling according to $\log(N(\varepsilon; \mathcal{P}, \|\cdot\|_{L^\infty([0,1])})) = \log(\varepsilon^{-1})$. We start by establishing the upper bound. For given $\varepsilon \in [0,1]$, set $T = \lfloor \frac{1}{2\varepsilon} \rfloor$, and define the points $\theta_i = 2\varepsilon i$, for $i = 0, 1, \ldots, T$. By also adding the point $\theta_{T+1} = 1$, we obtain a collection of $T+2$ points $\{\theta_0, \theta_1, \ldots, \theta_{T+1}\}$ in $[0,1]$. We show that the associated functions $\{f_{\theta_0}, f_{\theta_1}, \ldots, f_{\theta_{T+1}}\}$ form an ε-covering for $\mathcal{P}$. Indeed, for any $f_\theta \in \mathcal{P}$, we can find some θ_i in the covering such that $|\theta - \theta_i| \leq \varepsilon$. We then have

$$\|f_\theta - f_{\theta_i}\|_{L^\infty([0,1])} = \max_{x\in[0,1]} |e^{-\theta x} - e^{-\theta_i x}| \leq |\theta - \theta_i|,$$

where we used, for $\theta < \theta_i$,

$$\begin{aligned} \max_{x\in[0,1]} |e^{-\theta x} - e^{-\theta_i x}| &= \max_{x\in[0,1]} (e^{-\theta x} - e^{-\theta_i x}) = \max_{x\in[0,1]} e^{-\theta x}(1 - e^{-(\theta_i - \theta)x}) \leq \max_{x\in[0,1]} (1 - e^{-(\theta_i-\theta)x}) \\ &\leq \max_{x\in[0,1]} (\theta_i - \theta)x \leq \theta_i - \theta = |\theta - \theta_i|, \end{aligned}$$

as a consequence of $1-e^{-x} \leq x$, for $x \in [0,1]$, which is easily verified by noting that the function $g(x) = 1-e^{-x}-x$ satisfies $g(0) = 0$ and $g'(x) \leq 0$, for $x \in [0,1]$. The case $\theta > \theta_i$ follows similarly. In summary, we have shown that $N(\varepsilon; \mathcal{P}, \|\cdot\|_{L^\infty([0,1])}) \leq T + 2 \leq \frac{1}{2\varepsilon} + 2$.

In order to derive the lower bound, we first bound the packing number from below and then use Lemma IV.5. We start by constructing an explicit packing as follows. Set $\theta_0 = 0$ and define $\theta_i = -\log(1 - \varepsilon i)$, for all i such that $\theta_i \leq 1$. The largest index T such that this holds is given by $T = \lfloor \frac{1-1/e}{\varepsilon} \rfloor$. Moreover, note that for all i, j with $i \neq j$, we have $\|f_{\theta_i} - f_{\theta_j}\|_{L^\infty([0,1])} \geq |f_{\theta_i}(1) - f_{\theta_j}(1)| = |\varepsilon(i-j)| \geq \varepsilon$. We can therefore conclude that $M(\varepsilon; \mathcal{P}, \|\cdot\|_{L^\infty([0,1])}) \geq \lfloor \frac{1-1/e}{\varepsilon} \rfloor + 1$, and hence, due to the lower bound in Lemma IV.5,

$$N(\varepsilon; \mathcal{P}, \|\cdot\|_{L^\infty([0,1])}) \geq M(2\varepsilon; \mathcal{P}, \|\cdot\|_{L^\infty([0,1])}) \geq \left\lfloor \frac{1-1/e}{2\varepsilon} \right\rfloor + 1,$$

as claimed. We have thus established that the function class $\mathcal{P}$ has metric entropy scaling according to

$$\log(N(\varepsilon; \mathcal{P}, \|\cdot\|_{L^\infty([0,1])})) \asymp \log(1/\varepsilon), \text{ as } \varepsilon \to 0.$$

This rate is typical for one-parameter function classes.

We now turn our attention to richer function classes and start by considering Lipschitz functions on the d-dimensional unit cube, meaning real-valued functions on $[0,1]^d$ such that

$$|f(x) - f(y)| \leq L\|x - y\|_\infty, \qquad \text{for all} \quad x, y \in [0,1]^d.$$

This class, denoted as $\mathcal{F}_L([0,1]^d)$, has metric entropy scaling [64], [62]

$$\log N(\varepsilon; \mathcal{F}_L, \|\cdot\|_{L^\infty([0,1]^d)}) \asymp (L/\varepsilon)^d. \tag{16}$$

Contrasting the exponential dependence of metric entropy in (16) on the ambient dimension d to the linear dependence we identified earlier for simpler sets such as unit balls in $\mathbb{R}^d$, where we had

$$\log N(\varepsilon; \mathcal{B}, \|\cdot\|_\infty) \asymp d \log(\varepsilon^{-1}),$$

shows that $\mathcal{F}_L([0,1]^d)$ is significantly more massive.

We are now ready to relate the optimal exponent $\gamma^*(\mathcal{C})$ in Definition IV.1 to metric entropy scaling. All the examples of metric entropy scaling we have seen exhibit a behavior that fits the law $\log(N(\varepsilon; \mathcal{C}, \|\cdot\|)) \asymp \varepsilon^{-1/\gamma}$ or $\log(N(\varepsilon; \mathcal{C}, \|\cdot\|)) \asymp \varepsilon^{-1/\gamma} \log(\varepsilon^{-1})^\beta$. The optimal exponent is hence a crude measure of growth insensitive to log-factors or similar factors that are dominated by the growth of $\varepsilon^{-1/\gamma}$.

While we restrict ourselves to the approximation of functions on Euclidean domains, the framework described in this section can be extended to functions on manifolds (see e.g. [65]). As such, an interesting direction for future research would be the extension of the deep neural network approximation theory developed in this paper to functions on manifolds. First results on the neural network approximation of functions on manifolds have been reported in [57], [13], [66]. For further reading on the general subject of function approximation on manifolds, we recommend [67] and references therein.

V. Approximation with Dictionaries

We now show how Kolmogorov-Donoho rate-distortion theory can be put to work in the context of optimal approximation with dictionaries. Again, this subsection is of review nature. We start with a brief discussion of basics on optimal approximation in Hilbert spaces. Specifically, we shall consider two types of approximation, namely linear and nonlinear.

Let $\mathcal{H}$ be a Hilbert space equipped with inner product $\langle\cdot,\cdot\rangle$ and induced norm $\|\cdot\|_\mathcal{H}$ and let e_k, $k = 1, 2, \dots$ be an orthonormal basis for $\mathcal{H}$. For linear approximation, we use the linear space $\mathcal{H}_M := \operatorname{span}\{e_k : 1 \leq k \leq M\}$ to approximate a given element $f \in \mathcal{H}$. We measure the approximation error by

$$E_M(f) := \inf_{g \in \mathcal{H}_M} \|f - g\|_\mathcal{H}.$$

In nonlinear approximation, we consider best M-term approximation, which replaces $\mathcal{H}_M$ by the set Σ_M consisting of all elements $g \in \mathcal{H}$ that can be expressed as

$$g = \sum_{k \in \Lambda} c_k e_k,$$

where $\Lambda \subseteq \mathbb{N}$ is a set of indices with $|\Lambda| \leq M$. Note that, in contrast to $\mathcal{H}_M$, the set Σ_M is not a linear space as a linear combination of two elements in Σ_M will, in general, need $2M$ terms in its representation by the e_k. Analogous to E_M, we define the error of best M-term approximation

$$\Gamma_M(f) := \inf_{g \in \Sigma_M} \|f - g\|_\mathcal{H}.$$

The key difference between linear and nonlinear approximation resides in the fact that in nonlinear approximation, we can choose the M elements e_k participating in the approximation of f freely from the entire orthonormal basis whereas in linear approximation we are constrained to the first M elements. A classical example for linear approximation is the approximation of periodic functions by the Fourier series elements corresponding to the M lowest frequencies (assuming natural ordering of the dictionary). This approach clearly leads to poor approximation if the function under consideration consists of high-frequency components. In contrast, in nonlinear approximation we would seek the M frequencies that yield the smallest approximation error. In summary, it is clear that (nonlinear) best M-term approximation can achieve smaller approximation error than linear M-term approximation.

We shall consider nonlinear approximation in arbitrary, possibly redundant, dictionaries, i.e., in frames [68], and will exclusively be interested in the case $\mathcal{H} = L^2(\Omega)$, in particular the approximation error will be measured in terms of $L^2(\Omega)$-norm. Specifically, let $\mathcal{C}$ be a set of functions in $L^2(\Omega)$ and consider a countable family of functions $\mathcal{D} := (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$, termed *dictionary*.

We consider the *best M-term approximation error* of $f \in \mathcal{C}$ in $\mathcal{D}$ defined as follows.

Definition V.1. *[58] Given $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, a function class $\mathcal{C} \subseteq L^2(\Omega)$, and a dictionary $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$, we define, for $f \in \mathcal{C}$ and $M \in \mathbb{N}$,*

$$\Gamma_M^{\mathcal{D}}(f) := \inf_{\substack{I_{f,M} \subseteq \mathbb{N},\\ |I_{f,M}|=M,(c_i)_{i\in I_{f,M}}}} \left\| f - \sum_{i\in I_{f,M}} c_i\varphi_i \right\|_{L^2(\Omega)}. \tag{17}$$

We call $\Gamma_M^{\mathcal{D}}(f)$ the best M-term approximation error of f in $\mathcal{D}$*. Every $f_M = \sum_{i\in I_{f,M}} c_i\varphi_i$ attaining the infimum in* (17) *is referred to as a* best M-term approximation *of f in $\mathcal{D}$. The supremal $\gamma > 0$ such that*

$$\sup_{f\in\mathcal{C}} \Gamma_M^{\mathcal{D}}(f) \in \mathcal{O}(M^{-\gamma}),\ M \to \infty,$$

will be denoted by $\gamma^*(\mathcal{C},\mathcal{D})$. We say that the best M-term approximation rate of $\mathcal{C}$ in the dictionary $\mathcal{D}$ *is $\gamma^*(\mathcal{C},\mathcal{D})$.*

Function classes $\mathcal{C}$ widely studied in the approximation theory literature include unit balls in Lebesgue, Sobolev, or Besov spaces [59], as well as α-cartoon-like functions [69]. A wealth of structured dictionaries $\mathcal{D}$ is provided by the area of applied harmonic analysis, starting with wavelets [70], followed by ridgelets [39], curvelets [71], shearlets [72], parabolic molecules [73], and most generally α-molecules [69], which include all previously named dictionaries as special cases. Further examples are Gabor frames [17], Wilson bases [74], and wave atoms [18].

The best M-term approximation rate $\gamma^*(\mathcal{C},\mathcal{D})$ according to Definition V.1 quantifies how difficult it is to approximate a given function class $\mathcal{C}$ in a fixed dictionary $\mathcal{D}$. It is sensible to ask whether for given $\mathcal{C}$, there is a fundamental limit on $\gamma^*(\mathcal{C},\mathcal{D})$ when one is allowed to vary over $\mathcal{D}$. To answer this question, we first note that for every dense (and countable) $\mathcal{D}$, for any given $f \in \mathcal{C}$, by density of $\mathcal{D}$, there exists a single dictionary element that approximates f to within arbitrary accuracy thereby effectively realizing a 1-term approximation for arbitrary approximation error ε. Formally, this can be expressed through $\gamma^*(\mathcal{C},\mathcal{D}) = \infty$. Identifying this single dictionary element or, more generally, the M elements participating in the best M-term approximation is in general, however, practically infeasible as it entails searching through the infinite set $\mathcal{D}$ and requires an infinite number of bits to describe the indices of the participating elements. This insight leads to the concept of "best M-term approximation subject to polynomial-depth search" as introduced by Donoho in [15]. Here, the basic idea is to restrict the search for the elements in $\mathcal{D}$ participating in the best M-term approximation to the first $\pi(M)$ elements of $\mathcal{D}$, with π a polynomial. We formalize this under the name of effective best M-term approximation as follows.

Definition V.2. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, $\mathcal{C} \subseteq L^2(\Omega)$ be compact, and $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$. We define for $M \in \mathbb{N}$ and π a polynomial*

$$\varepsilon^{\pi}_{\mathcal{C},\mathcal{D}}(M) := \sup_{f\in\mathcal{C}} \inf_{\substack{I_{f,M}\subseteq\{1,2,\dots,\pi(M)\},\\ |I_{f,M}|=M,\, |c_i|\le\pi(M)}} \left\| f - \sum_{i\in I_{f,M}} c_i\varphi_i \right\|_{L^2(\Omega)} \tag{18}$$

and

$$\gamma^{*,\mathit{eff}}(\mathcal{C},\mathcal{D}) := \sup\{\gamma \ge 0 \colon \exists \text{ polynomial } \pi \text{ s.t. } \varepsilon^{\pi}_{\mathcal{C},\mathcal{D}}(M) \in \mathcal{O}(M^{-\gamma}),\ M \to \infty\}. \tag{19}$$

We refer to $\gamma^{*,\mathit{eff}}(\mathcal{C},\mathcal{D})$ as the effective best M-term approximation rate of $\mathcal{C}$ in the dictionary $\mathcal{D}$.

Note that we required the coefficients c_i in the approximant in Definition V.2 to be polynomially bounded in M. This condition, not present in [14], [60] and easily met for generic $\mathcal{C}$ and $\mathcal{D}$, is imposed for technical reasons underlying the transference results in Section VII. Strictly speaking—relative to [14], [60]—we hence get a subtly different notion of approximation rate. Exploring the implications of this difference is certainly worthwhile, but deemed beyond the scope of this paper.

We next present a central result in best M-term approximation theory stating that for compact $\mathcal{C} \subseteq L^2(\Omega)$, the effective best M-term approximation rate in any dictionary $\mathcal{D}$ is upper-bounded by $\gamma^*(\mathcal{C})$ and hence limited by the "description complexity" of $\mathcal{C}$. This endows $\gamma^*(\mathcal{C})$ with operational meaning.

Theorem V.3. *[14], [60] Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and let $\mathcal{C} \subseteq L^2(\Omega)$ be compact. The effective best M-term approximation rate of the function class $\mathcal{C} \subseteq L^2(\Omega)$ in the dictionary $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$ satisfies*

$$\gamma^{*,\mathit{eff}}(\mathcal{C},\mathcal{D}) \le \gamma^*(\mathcal{C}).$$

In light of this result the following definition is natural (see also [60]).

Definition V.4. *(Kolmogorov-Donoho optimality) Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and let $\mathcal{C} \subseteq L^2(\Omega)$ be compact. If the effective best M-term approximation rate of the function class $\mathcal{C} \subseteq L^2(\Omega)$ in the dictionary $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$ satisfies*

$$\gamma^{*,\mathit{eff}}(\mathcal{C},\mathcal{D}) = \gamma^*(\mathcal{C}),$$

we say that the function class $\mathcal{C}$ is optimally representable *by $\mathcal{D}$.*

As the ideas underlying the proof of Theorem V.3 are essential ingredients in the development of a kindred theory of best M-weight approximation rates for neural networks, we present a detailed proof, which is similar to that in [60]. We perform, however, some minor technical modifications with an eye towards rendering the proof a suitable genesis for the new theory of best M-weight approximation with neural networks, developed in the next section. The spirit of the proof is to construct, for every given $M \in \mathbb{N}$ an encoder that, for each $f \in \mathcal{C}$, maps the indices of the dictionary elements participating in the effective best M-term approximation[3] of f, along with the corresponding coefficients c_i, to a bitstring. This bitstring needs to be of sufficient length for the decoder to be able to reconstruct an approximation to f with an error which is of the same order as that of the best M-term approximation we started from. As elucidated in the proof, this can be accomplished while ensuring that the length of the bitstring is proportional to $M\log(M)$, which upon noting that $\varepsilon = M^{-\gamma}$ implies $M = \varepsilon^{-1/\gamma}$, establishes optimality.

Proof of Theorem V.3. The proof will be based on showing that for every $\gamma \in \mathbb{R}_+$ the following Implication (I) holds: Assume that there exist a constant $C > 0$ and a polynomial π such that for every $M \in \mathbb{N}$, the following holds: For every $f \in \mathcal{C}$, there are an index set $I_{f,M} \subseteq \{1, 2, \ldots, \pi(M)\}$ and coefficients $(c_i)_{i \in I_{f,M}} \subseteq \mathbb{R}$ with $|c_i| \leq \pi(M)$ so that

$$\|f - \sum_{i \in I_{f,M}} c_i \varphi_i\|_{L^2(\Omega)} \leq CM^{-\gamma}. \tag{20}$$

This implies the existence of a constant $C' > 0$ such that for every $M \in \mathbb{N}$, there is an encoder-decoder pair $(E_M, D_M) \in \mathfrak{E}^{\ell(M)} \times \mathfrak{D}^{\ell(M)}$ with $\ell(M) \leq C'M\log(M)$ and

$$\|f - D_M(E_M(f))\|_{L^2(\Omega)} \leq C'M^{-\gamma}. \tag{21}$$

The implication will be proven by explicit construction. For a given $f \in \mathcal{C}$, we pick an M-term approximation according to (20) and encode the associated index set $I_{f,M}$ and weights c_i as follows. First, note that owing to $|I_{f,M}| \leq \pi(M)$, each index in $I_{f,M}$ can be represented by at most $C_\pi \log(M)$ bits; this results in a total of $C_\pi M \log(M)$ bits needed to encode the indices of all dictionary elements participating in the M-term approximation. The encoder and the decoder are assumed to know C_π, which allows stacking of the binary representations of the indices such that the decoder can read them off uniquely from the sequence of their binary representations.

We proceed to the encoding of the coefficients c_i. First, note that even though the c_i are bounded (namely, polynomially in M) by assumption, we did not impose bounds on the norms of the dictionary elements $\{\varphi_i\}_{i \in I_{f,M}}$ participating in the M-term approximation under consideration. Hence, we can not, in general, expect to be able to control the approximation error incurred by reconstructing f from quantized c_i. We can get around this by performing a Gram-Schmidt orthogonalization on the dictionary elements $\{\varphi_i\}_{i \in I_{f,M}}$ and, as will be seen later, using the fact

[3] Note that as we have an infimum in (18) an effective best M-term approximation need not exist, but we can pick an M-term approximation that yields an error arbitrarily close to the infimum.

that the function class $\mathcal{C}$ was assumed to be compact. Specifically, this Gram-Schmidt orthogonalization yields a set of functions $\{\tilde{\varphi}_i\}_{i \in \tilde{I}_{f,\widetilde{M}}}$, with $\widetilde{M} \leq M$, that has the same span as $\{\varphi_i\}_{i \in I_{f,M}}$. Next, we define (implicitly) the coefficients $\tilde{c}_i$ according to

$$\sum_{i \in \tilde{I}_{f,\widetilde{M}}} \tilde{c}_i \tilde{\varphi}_i = \sum_{i \in I_{f,M}} c_i \varphi_i. \tag{22}$$

Now, note that

$$\left\| \sum_{i \in \tilde{I}_{f,\widetilde{M}}} \tilde{c}_i \tilde{\varphi}_i \right\|_{L^2(\Omega)}^2 = \left\| f - (f - \sum_{i \in \tilde{I}_{f,\widetilde{M}}} \tilde{c}_i \tilde{\varphi}_i) \right\|_{L^2(\Omega)}^2 \leq \|f\|_{L^2(\Omega)}^2 + \left\| f - \sum_{i \in I_{f,M}} c_i \varphi_i \right\|_{L^2(\Omega)}^2 .$$

Making use of the orthonormality of the $\tilde{\varphi}_i$, we can conclude that

$$\sum_{i \in \tilde{I}_{f,\widetilde{M}}} |\tilde{c}_i|^2 \leq \sup_{f \in \mathcal{C}} \|f\|_{L^2(\Omega)}^2 + C^2 M^{-2\gamma}.$$

As $\mathcal{C}$ is compact by assumption, we have $\sup_{f \in \mathcal{C}} \|f\|_{L^2(\Omega)}^2 < \infty$, which establishes that the coefficients $\tilde{c}_i$ are uniformly bounded. This, in turn, allows us to quantize them, specifically, we shall round the $\tilde{c}_i$ to integer multiples of $M^{-(\gamma+1/2)}$, and denote the resulting rounded coefficients by $\hat{c}_i$. As the $\tilde{c}_i$ are uniformly bounded, this results in a number of quantization levels that is proportional to $M^{(\gamma+1/2)}$. The number of bits needed to store the binary representations of the quantized coefficients is therefore proportional to $M \log(M)$. Again, the proportionality constant is assumed known to encoder and decoder, which allows us to stack the binary representations of the quantized coefficients in a uniquely decodable manner. The resulting bitstring is then appended to the bitstring encoding the indices of the participating dictionary elements. We finally note that the specific choice of the exponent $\gamma + 1/2$ is informed by the upper bound on the reconstruction error we are allowed, this will be made explicit below in the description of the decoder.

In summary, we have mapped the function f to a bitstring of length $\mathcal{O}(M \log(M))$. The decoder is presented with this bitstring and reconstructs an approximation to f as follows. It first reads out the indices of the set $I_{f,M}$ and the quantized coefficients $\hat{c}_i$. Recall that this is uniquely possible. Next, the decoder performs a Gram-Schmidt orthonormalization on the set of dictionary elements indexed by $I_{f,M}$. The error resulting from reconstructing the function f from the quantized coefficients $\hat{c}_i$ rather than the exact coefficients $\tilde{c}_i$ can be bounded according to

$$\begin{aligned} \left\| f - \sum_{i \in \tilde{I}_{f,\widetilde{M}}} \hat{c}_i \tilde{\varphi}_i \right\|_{L^2(\Omega)} &= \left\| f - \sum_{i \in \tilde{I}_{f,\widetilde{M}}} \tilde{c}_i \tilde{\varphi}_i + \sum_{i \in \tilde{I}_{f,\widetilde{M}}} \tilde{c}_i \tilde{\varphi}_i - \sum_{i \in \tilde{I}_{f,\widetilde{M}}} \hat{c}_i \tilde{\varphi}_i \right\|_{L^2(\Omega)} \\ &\leq \left\| f - \sum_{i \in \tilde{I}_{f,\widetilde{M}}} \tilde{c}_i \tilde{\varphi}_i \right\|_{L^2(\Omega)} + \left\| \sum_{i \in \tilde{I}_{f,\widetilde{M}}} (\tilde{c}_i - \hat{c}_i) \tilde{\varphi}_i \right\|_{L^2(\Omega)} \\ &= \left\| f - \sum_{i \in \tilde{I}_{f,\widetilde{M}}} \tilde{c}_i \tilde{\varphi}_i \right\|_{L^2(\Omega)} + \left(\sum_{i \in \tilde{I}_{f,\widetilde{M}}} |\tilde{c}_i - \hat{c}_i|^2 \right)^{1/2} , \end{aligned} \tag{23}$$

where in the last step we again exploited the orthonormality of the $\tilde{\varphi}_i$. Next, note that due to the choice of the quantizer resolution, we have $|\tilde{c}_i - \hat{c}_i|^2 \leq C'' M^{-2\gamma-1}$ for some constant C''. With $\widetilde{M} \leq M$ this yields

$$\sum_{i \in \tilde{I}_{f,\widetilde{M}}} |\tilde{c}_i - \hat{c}_i|^2 \leq C'' M^{-2\gamma}.$$

Combining (20), (22), and (23), we obtain

$$\left\| f - \sum_{i \in \tilde{I}_{f,\widetilde{M}}} \hat{c}_i \tilde{\varphi}_i \right\|_{L^2(\Omega)} \leq C' M^{-\gamma},$$

for some constant C'. As the length of the bitstring used in this construction is proportional to $M \log(M)$, the claim (21) is established.

Now, we note that the antecedent of Implication (I) holds for all $\gamma < \gamma^{*,\mathrm{eff}}(\mathcal{C}, \mathcal{D})$. Assume next, towards a contradiction, that the antecedent holds for a $\gamma > \gamma^*(\mathcal{C})$. This would imply that for any $\gamma' < \gamma$,

$$\inf_{(E,D) \in \mathfrak{E}^L \times \mathfrak{D}^L} \sup_{f \in \mathcal{C}} \|D(E(f)) - f\|_{L^2(\Omega)} \in \mathcal{O}\big(L^{-\gamma'}\big),\ L \to \infty. \tag{24}$$

In particular, (24) would hold for some $\gamma' > \gamma^*(\mathcal{C})$ which, owing to (13) stands in contradiction to the definition of $\gamma^*(\mathcal{C})$. This completes the proof. □

Space		$\mathcal{C}$	Optimal dictionary	$\gamma^*(\mathcal{C})$	
L^2-Sobolev	$W_2^m([0,1])$	$\mathcal{U}(W_2^m([0,1]))$	Fourier/Wavelet basis	m	[75, Sec. 14.2]
Hölder	$C^\alpha([0,1])$	$\mathcal{U}(C^\alpha([0,1]))$	Wavelet basis	α	[75, Sec. 14.2]
Bump Algebra	$B^1_{1,1}([0,1])$	$\mathcal{U}(B^1_{1,1}([0,1]))$	Wavelet basis	1	[75, Sec. 14.2]
Bounded Variation	$BV([0,1])$	$\mathcal{U}(BV([0,1]))$	Haar basis	1	[75, Sec. 14.2]
L^p-Sobolev[4]	$W_p^m(\Omega)$	$\mathcal{U}(W_p^m(\Omega))$	Wavelet frame	$\frac{m}{d}$	[76, Thm. 1.3]
Besov[5]	$B^m_{p,q}(\Omega)$	$\mathcal{U}(B^m_{p,q}(\Omega))$	Wavelet frame	$\frac{m}{d}$	[76, Thm. 1.3]
Modulation[6]	$M^s_{p,p}(\mathbb{R}^d)$	$\mathcal{U}(M^s_{p,p}(\mathbb{R}^d))$	Wilson basis	$(\frac{1}{p} - \frac{1}{2} + \frac{2s}{d})^{-1}$	[77, Thm. 4.4]
Cartoon functions[7]		$\mathcal{E}^\beta([-\frac{1}{2}, \frac{1}{2}]^d)$	α-Curvelet frame[8]	$\frac{\beta(d-1)}{2}$	[23]

Table 1: Optimal exponents and corresponding optimal dictionaries. $\mathcal{U}(X) = \{f \in X : \|f\|_X \leq 1\}$ denotes the unit ball in the space X and $\Omega \subseteq \mathbb{R}^d$ is a Lipschitz domain. Recall that compactness of these unit balls is w.r.t. L^2-norm.

[4] $p \in [1, \infty]$, $m > d(1/p - 1/2)_+$

[5] $p, q \in (0, \infty]$, $m > d(1/p - 1/2)_+$

[6] $1 < p < 2$, $s \in \mathbb{R}_+$

[7] This is actually a set of functions and not a (unit) ball in a Banach space.

[8] For $d = 2$, see [78].

The optimal exponent $\gamma^*(\mathcal{C})$ is known for various function classes such as unit balls in Besov spaces $B^m_{p,q}(\mathbb{R}^d)$ with $p, q \in (0, \infty]$ and $m > d(1/p - 1/2)_+$, where $\gamma^*(\mathcal{C}) = m/d$ (see [76]), and unit balls in (polynomially) weighted modulation spaces $M^s_{p,p}(\mathbb{R}^d)$ with $p \in (1, 2)$ and $s \in \mathbb{R}_+$, where $\gamma^*(\mathcal{C}) = (\frac{1}{p} - \frac{1}{2} + \frac{2s}{d})^{-1}$ (see [77]). A further example is the set of β-cartoon-like functions, which are β-smooth on some bounded d-dimensional domain with sufficiently smooth boundary and zero otherwise. Here, we have $\gamma^*(\mathcal{C}) = \beta(d-1)/2$ (see [79], [78], [23]). These examples along with additional ones are summarized in Table 1. For an extensive summary of metric entropy results and techniques for their derivation, we also refer to [64].

We conclude this section with general remarks on certain formal aspects of the Kolmogorov-Donoho rate-distortion framework. First, we note that for the set $\mathcal{C} \subseteq L^2(\Omega)$ to have a well-defined optimal exponent it must be relatively compact[9]. This follows from the fact that the set over which the minimum in the definition (12) of $L(\varepsilon, \mathcal{C})$ is taken must be nonempty for every $\varepsilon \in (0, \infty)$. To see this, note that every length-$L(\varepsilon, \mathcal{C})$ encoder-decoder pair induces an ε-covering of $\mathcal{C}$ with at most $2^{L(\varepsilon,\mathcal{C})}$ balls (and ball centers $\{D(E(f))\}_{f\in\mathcal{C}}$). It hence follows that $\mathcal{C}$ must be totally bounded and thus relatively compact as a consequence of $L^2(\Omega)$ being a complete metric space [80, Thm. 45.1].

As shown in the proof of Theorem V.3, effective best M-term approximations construct encoder-decoder pairs and thereby induce ε-coverings. By the arguments just made, this implies that also $\gamma^{*,\mathrm{eff}}(\mathcal{C}, \mathcal{D})$ is well-defined only for compact function classes $\mathcal{C}$.

A consequence of the compactness requirement on $\mathcal{C}$ is that the spaces in Table 1 either consist of functions on bounded domains or, in the case of modulation spaces, are equipped with a weighted norm. In order to provide intuition on why this must be so, let us consider a function space $(X, \|\cdot\|_X)$ with $X \subseteq L^2(\mathbb{R}^d)$ and $\|\cdot\|_X$ translation invariant. Take $\varepsilon > 0$ and $f \in X$ with $\|f\|_X = 1$ and choose $C > 0$ such that $\|f\|_{L^2([-C,C]^d)} > \frac{4}{5}\|f\|_{L^2(\mathbb{R}^d)}$. Now, consider the family of translates of f given by $f_i(x) := f(x - 2Ci)$, $i \in \mathbb{Z}^d$, and note that $\|f_i\|_X = 1$ for all $i \in \mathbb{Z}^d$ by translation invariance of $\|\cdot\|_X$. Furthermore, we have

$$\|f_i\|_{L^2([-C,C]^d)} = \Big(\|f_i\|^2_{L^2(\mathbb{R}^d)} - \|f_i\|^2_{L^2(\mathbb{R}^d\setminus[-C,C]^d)}\Big)^{\frac{1}{2}} \leq \Big(\|f\|^2_{L^2(\mathbb{R}^d)} - \|f\|^2_{L^2([-C,C]^d)}\Big)^{\frac{1}{2}} < \tfrac{3}{5}\|f\|_{L^2(\mathbb{R}^d)}$$

for all $i \in \mathbb{Z}^d\backslash\{0\}$ by construction. This, in turn, implies

$$\|f_i - f_j\|_{L^2(\mathbb{R}^d)} = \|f_{i-j} - f\|_{L^2(\mathbb{R}^d)} \geq \|f_{i-j} - f\|_{L^2([-C,C]^d)} > \tfrac{1}{5}\|f\|_{L^2(\mathbb{R}^d)} \tag{25}$$

for all $i, j \in \mathbb{Z}^d$, with $i \neq j$, by the reverse triangle inequality. As such no ε-ball (w.r.t. $L^2(\mathbb{R}^d)$-norm) with $\varepsilon \leq \frac{1}{10}\|f\|_{L^2(\mathbb{R}^d)}$ can contain more than one of the infinitely many $(f_i)_{i\in\mathbb{Z}^d}$ which are, however, all contained in the unit ball $\mathcal{U}(X)$ of the space $(X, \|\cdot\|_X)$. This implies that $\mathcal{U}(X)$ cannot be totally bounded and thereby not relatively compact (w.r.t. $L^2(\mathbb{R}^d)$-norm). Somewhat nonchalantly speaking, for spaces equipped with translation-invariant norms this issue can be avoided by considering functions that live on a bounded domain, which ensures that

[9] For the sake of simplicity, we assume, however, compactness throughout even though relative compactness (i.e. having a compact closure) would be sufficient.

(25) pertains only to a finite number of translates. Alternatively, for spaces of functions living on unbounded domains once can consider weighted norms that are not translation invariant. Here, the weighting effectively constrains the functions to a bounded domain.

The less restrictive concept of best M-term approximation rate $\gamma^*(\mathcal{C},\mathcal{D})$ (see Definition V.1) is, in apparent contrast, often studied for noncompact function classes $\mathcal{C}$.

In [75, Sec. 15.2] a condition for $\gamma^{*,\text{eff}}(\mathcal{C},\mathcal{D})$ and $\gamma^*(\mathcal{C},\mathcal{D})$ to coincide is presented. Specifically, this condition, referred to as tail compactness, is expressed as follows. Let $\mathcal{C} \subseteq L^2(\Omega)$ be bounded and let $\mathcal{D} = \{\varphi_i\}_{i\in\mathbb{N}}$ be an ordered orthonormal basis for $\mathcal{C}$. We say that tail compactness holds if there exist $C, \beta > 0$ such that for all $N \in \mathbb{N}$,

$$\sup_{f\in\mathcal{C}} \left\| f - \sum_{i=1}^{N} \langle f, \varphi_i\rangle \varphi_i \right\|_{L^2(\Omega)} \leq C N^{-\beta}. \tag{26}$$

In order to see that (26) implies $\gamma^{*,\text{eff}}(\mathcal{C},\mathcal{D}) = \gamma^*(\mathcal{C},\mathcal{D})$, we consider, for fixed $f \in \mathcal{C}$, the (unconstrained) best M-term approximation $f_M = \sum_{i\in I}\langle f,\varphi_i\rangle\varphi_i$ with $I \subseteq \mathbb{N}$, $|I| = M$. We now modify this M-term approximation by letting $\alpha := \lceil \gamma^*(\mathcal{C},\mathcal{D})/\beta \rceil \in \mathbb{N}$ and removing, in the expansion $f_M = \sum_{i\in I}\langle f,\varphi_i\rangle\varphi_i$, all terms corresponding to indices that are larger than M^α. Recalling that in Definition V.2 the same polynomial π bounds the search depth and the size of the coefficients, it follows that the modified approximation we just constructed obeys a polynomial depth search constraint with constraining polynomial $\pi_\alpha(x) = x^\alpha + S$, where $S := \sup_{f\in\mathcal{C}} \|f\|_{L^2(\Omega)}$. Here, owing to orthonormality of $\mathcal{D}$, S accounts for the size of the expansion coefficients $\langle f,\varphi_i\rangle$. In order to complete the argument, we need to show that the additional approximation error incurred by removing terms in $f_M = \sum_{i\in I}\langle f,\varphi_i\rangle\varphi_i$ is in $\mathcal{O}(M^{-\gamma^*(\mathcal{C},\mathcal{D})})$, i.e., it is of the same order as the error corresponding to the original (unconstrained) best M-term approximation. Due to orthonormality of $\mathcal{D}$ this additional error is given by the norm of $\sum_{i\in I, i>\pi_\alpha(M)}\langle f,\varphi_i\rangle\varphi_i$ and can, by virtue of (26), be bounded as

$$\begin{aligned}\left\| \sum_{i\in I, i>\pi_\alpha(M)} \langle f,\varphi_i\rangle\varphi_i \right\|_{L^2(\Omega)} &\leq \left\| \sum_{i=\pi_\alpha(M)+1}^{\infty} \langle f,\varphi_i\rangle\varphi_i \right\|_{L^2(\Omega)} = \left\| f - \sum_{i=1}^{\pi_\alpha(M)} \langle f,\varphi_i\rangle\varphi_i \right\|_{L^2(\Omega)} \\ &\leq C(\pi_\alpha(M))^{-\beta} \in \mathcal{O}(M^{-\gamma^*(\mathcal{C},\mathcal{D})}),\end{aligned}$$

which establishes the claim. We have hence shown that under tail compactness of arbitrary rate $\beta > 0$, $\gamma^*(\mathcal{C},\mathcal{D}) = \gamma^{*,\text{eff}}(\mathcal{C},\mathcal{D})$, and hence there is no cost incurred by imposing a polynomial depth search constraint combined with a polynomial bound on the size of the expansion coefficients. For the more general case of $\mathcal{D}$ a frame, we refer to [60, Sec. 5.4.3] for an analogous argument. Finally, we remark that the tail compactness inequality (26) can be interpreted as quantifying the rate of linear approximation for $\mathcal{C}$ in $\mathcal{D}$. Two examples of pairs $(\mathcal{C},\mathcal{D})$ satisfying tail compactness, namely Besov spaces with wavelet bases and modulation spaces with Wilson bases, are provided in Appendices B and C, respectively.

As already mentioned, a larger optimal exponent $\gamma^*(\mathcal{C})$ leads to faster error decay (specifically according to $L^{-\gamma^*(\mathcal{C})}$) and hence corresponds to a function class of smaller complexity. As such, techniques for deriving lower bounds on the optimal exponent are often based on variations of the approach employed in the proof of Theorem V.3, namely on the explicit construction of encoder-decoder pairs (in the case of the proof of Theorem V.3 by encoding the dictionary elements participating in the M-term approximation). A powerful method for deriving upper bounds on the optimal exponent is the hypercube embedding approach proposed by Donoho in [79]; the basic idea here is to show that the function class $\mathcal{C}$ under consideration contains a sufficiently complex embedded set of orthogonal hypercubes and to then find the exponent corresponding to this set. An interesting alternative technique for deriving optimal exponents was proposed in the context of modulation spaces in [77]. The essence of this approach is to exploit the isomorphism between weighted modulation spaces and weighted mixed-norm sequence spaces [17] and to then utilize results about entropy numbers of operators between sequence spaces.

VI. Approximation with Deep Neural Networks

Inspired by the theory of best M-term approximation with dictionaries, we now develop the new concept of best M-weight approximation through neural networks. At the heart of this theory lies the interpretation of the network weights as the counterpart of the coefficients c_i in best M-term approximation. In other words, parsimony in terms of the number of participating elements in a dictionary is replaced by parsimony in terms of network connectivity. Our development will parallel that for best M-term approximation in the previous section.

Before proceeding to the specifics, we would like to issue a general remark. While the neural network approximation results in Section III were formulated in terms of L^∞-norm, we shall be concerned with L^2-norm approximation here, on the one hand paralleling the use of L^2-norm in the context of best M-term approximation, and on the other hand allowing for the approximation of discontinuous functions by ReLU neural networks, which, owing to the continuity of the ReLU nonlinearity, necessarily realize continuous functions.

We start by introducing the concept of best M-weight approximation rate.

Definition VI.1. *Given $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and a function class $\mathcal{C} \subseteq L^2(\Omega)$, we define, for $f \in \mathcal{C}$ and $M \in \mathbb{N}$,*

$$\Gamma_M^{\mathcal{N}}(f) := \inf_{\substack{\Phi \in \mathcal{N}_{d,1} \\ \mathcal{M}(\Phi) \leq M}} \|f - \Phi\|_{L^2(\Omega)}. \tag{27}$$

We call $\Gamma_M^{\mathcal{N}}(f)$ the best M-weight approximation error of f. *The supremal $\gamma > 0$ such that*

$$\sup_{f \in \mathcal{C}} \Gamma_M^{\mathcal{N}}(f) \in \mathcal{O}(M^{-\gamma}),\ M \to \infty,$$

will be denoted by $\gamma_{\mathcal{N}}^*(\mathcal{C})$. We say that the best M-weight approximation rate of $\mathcal{C}$ by neural networks *is $\gamma_{\mathcal{N}}^*(\mathcal{C})$.*

We emphasize that the infimum in (27) is taken over all networks with fixed input dimension d, no more than M nonzero (edge and node) weights, and arbitrary depth L. In particular, this means that the infimum is with respect to all possible network topologies and weight choices. The best M-weight approximation rate is fundamental as it benchmarks all algorithms that map a function f and an $\varepsilon > 0$ to a neural network approximating f with error no more than ε.

The two restrictions underlying the concept of effective best M-term approximation through dictionaries, namely polynomial depth search and polynomially bounded coefficients, are next addressed in the context of approximation through deep neural networks. We start by noting that the need for the former is obviated by the tree-like-structure of neural networks. To see this, first note that $\mathcal{W}(\Phi) \leq \mathcal{M}(\Phi)$ and $\mathcal{L}(\Phi) \leq \mathcal{M}(\Phi)$. As the total number of nonzero weights in the network can not exceed $\mathcal{L}(\Phi)\mathcal{W}(\Phi)(\mathcal{W}(\Phi)+1)$, this yields at most $\mathcal{O}(\mathcal{M}(\Phi)^3)$ possibilities for the "locations" (in terms of entries in the A_ℓ and the b_ℓ) of the $\mathcal{M}(\Phi)$ nonzero weights. Encoding the locations of the $\mathcal{M}(\Phi)$ nonzero weights hence requires $\log(\binom{C\mathcal{M}(\Phi)^3}{\mathcal{M}(\Phi)}) = \mathcal{O}(\mathcal{M}(\Phi)\log(\mathcal{M}(\Phi)))$ bits. This assumes, however, that the architecture of the network, i.e., the number of layers $\mathcal{L}(\Phi)$ and the N_k are known. Proposition VI.7 below shows that the architecture can, indeed, also be encoded with $\mathcal{O}(\mathcal{M}(\Phi)\log(\mathcal{M}(\Phi)))$ bits. In summary, we can therefore conclude that the tree-like-structure of neural networks automatically guarantees what we had to enforce through the polynomial depth search constraint in the case of best M-term approximation.

Inspection of the approximation results in Section III reveals that a sublinear growth restriction on $\mathcal{L}(\Phi)$ as a function of $\mathcal{M}(\Phi)$ is natural. Specifically, the approximation results in Section III all have $\mathcal{L}(\Phi)$ proportional to a polynomial in $\log(\varepsilon^{-1})$. As we are interested in approximation error decay according to $\mathcal{M}(\Phi)^{-\gamma}$, see Definition VI.1, this suggests to restrict $\mathcal{L}(\Phi)$ to growth that is polynomial in $\log(\mathcal{M}(\Phi))$.

The second restriction imposed in the definition of effective best M-term approximation, namely polynomially bounded coefficients, will be imposed in monomorphic manner on the magnitude of the weights. This growth condition will turn out natural in the context of the approximation results we are interested in and will, together with polylogarithmic depth growth, be seen below to allow rate-distortion-optimal quantization of the network weights. We remark, however, that networks with weights growing polynomially in $\mathcal{M}(\Phi)$ can be converted into networks with uniformly bounded weights at the expense of increased—albeit still of polylogarithmic scaling in $\mathcal{M}(\Phi)$—depth (see Proposition A.3). In summary, we will develop the concept of "best M-weight approximation subject to polylogarithmic depth and polynomial weight growth".

We start by introducing the following notation for neural networks with depth and weight magnitude bounded polylogarithmically respectively polynomially w.r.t. their connectivity.

Definition VI.2. *For $M, d, d' \in \mathbb{N}$, and π a polynomial, we define*

$$\mathcal{N}^{\pi}_{M,d,d'} := \{\Phi \in \mathcal{N}_{d,d'} : \mathcal{M}(\Phi) \leq M, \mathcal{L}(\Phi) \leq \pi(\log(M)), \mathcal{B}(\Phi) \leq \pi(M)\}.$$

Next, we formalize the notion of effective best M-weight approximation rate subject to polylogarithmic depth and polynomial weight growth.

Definition VI.3. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and let $\mathcal{C} \subseteq L^2(\Omega)$ be compact. We define for $M \in \mathbb{N}$ and π a polynomial*

$$\varepsilon_{\mathcal{N}}^{\pi}(M) := \sup_{f \in \mathcal{C}} \inf_{\Phi \in \mathcal{N}_{M,d,1}^{\pi}} \|f - \Phi\|_{L^2(\Omega)}$$

and

$$\gamma_{\mathcal{N}}^{*,\mathit{eff}}(\mathcal{C}) := \sup\{\gamma \geq 0 \colon \exists \text{ polynomial } \pi \text{ s.t. } \varepsilon_{\mathcal{N}}^{\pi}(M) \in \mathcal{O}(M^{-\gamma}),\, M \to \infty\}.$$

We refer to $\gamma_{\mathcal{N}}^{*,\mathit{eff}}(\mathcal{C})$ as the effective best M-weight approximation rate of $\mathcal{C}$.

We now state the equivalent of Theorem V.3 for approximation by deep neural networks. Specifically, we establish that the optimal exponent $\gamma^*(\mathcal{C})$ constitutes a fundamental bound on the effective best M-weight approximation rate of $\mathcal{C}$ as well.

Theorem VI.4. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and let $\mathcal{C} \subseteq L^2(\Omega)$ be compact. Then, we have*

$$\gamma_{\mathcal{N}}^{*,\mathit{eff}}(\mathcal{C}) \leq \gamma^*(\mathcal{C}).$$

The key ingredients of the proof of Theorem VI.4 are developed throughout this section and the formal proof appears at the end of the section. Before getting started, we note that, in analogy to Definition V.4, what we just found suggests the following.

Definition VI.5. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and let $\mathcal{C} \subseteq L^2(\Omega)$ be compact. We say that the function class $\mathcal{C} \subseteq L^2(\Omega)$ is* optimally representable by neural networks *if*

$$\gamma_{\mathcal{N}}^{*,\mathit{eff}}(\mathcal{C}) = \gamma^*(\mathcal{C}).$$

It is interesting to observe that the fundamental limits of effective best M-term approximation (through dictionaries) and effective best M-weight approximation in neural networks are determined by the same quantity, although the approximants in the two cases are vastly different. We have linear combinations of elements of a dictionary under polynomial weight growth of the coefficients and with the participating functions identified subject to a polynomial-depth search constraint in the former, and concatenations of affine functions followed by nonlinearities under polynomial growth constraints on the coefficients of the affine functions and with a polylogarithmic growth constraint on the number of concatenations in the latter case.

We now commence the program developing the proof of Theorem VI.4. As in the arguments in the proof sketch of Theorem V.3, the main idea is to compare the length of the bitstring needed to encode the approximating network to the minimax code length of the function class $\mathcal{C}$ to be approximated. To this end, we will need to represent the approximating network's nonzero weights, its architecture, i.e., L and the N_k, and the nonzero weights' locations as a bitstring. As the weights are real numbers and hence require, in principle, an infinite number of bits for their binary representations, we will have to suitably quantize them. In particular, the resolution of the corresponding quantizer will have to increase appropriately with decreasing ε. To formalize this idea, we start by defining the quantization employed.

Definition VI.6. *Let $m \in \mathbb{N}$ and $\varepsilon \in (0, 1/2)$. The network Φ is said to have (m, ε)-quantized weights if all its weights are elements of $2^{-m\lceil \log(\varepsilon^{-1})\rceil}\mathbb{Z} \cap [-\varepsilon^{-m}, \varepsilon^{-m}]$.*

A key ingredient of the proof of Theorem VI.4 is the following result, which establishes a fundamental lower bound on the connectivity of networks with quantized weights achieving uniform error ε over a given function class $\mathcal{C}$.

Proposition VI.7. *Let $d, d' \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, $\mathcal{C} \subseteq L^2(\Omega)$, and let π be a polynomial. Further, let*

$$\Psi : \left(0, \tfrac{1}{2}\right) \times \mathcal{C} \to \mathcal{N}_{d,d'}$$

be a map such that for every $\varepsilon \in (0, 1/2)$, $f \in \mathcal{C}$, the network $\Psi(\varepsilon, f)$ has $(\lceil \pi(\log(\varepsilon^{-1}))\rceil, \varepsilon)$-quantized weights and satisfies

$$\sup_{f\in\mathcal{C}} \|f - \Psi(\varepsilon, f)\|_{L^2(\Omega)} \leq \varepsilon.$$

Then,

$$\sup_{f\in\mathcal{C}} \mathcal{M}(\Psi(\varepsilon, f)) \notin \mathcal{O}\Big(\varepsilon^{-1/\gamma}\Big), \; \varepsilon \to 0, \quad \textit{for all } \gamma > \gamma^*(\mathcal{C}).$$

Proof. The proof is by contradiction. Let $\gamma > \gamma^*(\mathcal{C})$ and assume that $\sup_{f\in\mathcal{C}} \mathcal{M}(\Psi(\varepsilon, f)) \in \mathcal{O}(\varepsilon^{-1/\gamma}), \varepsilon \to 0$. The contradiction will be effected by constructing encoder-decoder pairs $(E_\varepsilon, D_\varepsilon) \in \mathfrak{E}^{\ell(\varepsilon)} \times \mathfrak{D}^{\ell(\varepsilon)}$ achieving uniform error ε over $\mathcal{C}$ with

$$\begin{aligned}\ell(\varepsilon) &\leq C_0 \cdot \sup_{f\in\mathcal{C}} \left(\mathcal{M}(\Psi(\varepsilon, f)) \log(\mathcal{M}(\Psi(\varepsilon, f))) + 1\right) (\log(\varepsilon^{-1}))^q \qquad (28)\\ &\leq C_0 \left(\varepsilon^{-1/\gamma} \log(\varepsilon^{-1/\gamma}) + 1\right) (\log(\varepsilon^{-1}))^q\\ &\leq C_1 \left(\varepsilon^{-1/\gamma}(\log(\varepsilon^{-1}))^{q+1} + (\log(\varepsilon^{-1}))^q\right) \in \mathcal{O}\left(\varepsilon^{-1/\nu}\right), \quad \text{for } \varepsilon \to 0,\end{aligned}$$

where $C_0, C_1, q > 0$ are constants not depending on f, ε and $\gamma > \nu > \gamma^*(\mathcal{C})$. The specific form of the upper bound (28) will become apparent in the construction of the bitstring representing Ψ detailed below.

We proceed to the construction of the encoder-decoder pairs $(E_\varepsilon, D_\varepsilon) \in \mathfrak{E}^{\ell(\varepsilon)} \times \mathfrak{D}^{\ell(\varepsilon)}$, which will be accomplished by encoding the network architecture, its topology, and the quantized weights in bitstrings of length $\ell(\varepsilon)$ satisfying (28) while guaranteeing unique reconstruction (of the network). For the sake of notational simplicity, we fix $\varepsilon \in (0, 1/2)$ and $f \in \mathcal{C}$ and set $\Psi := \Psi(\varepsilon, f)$, $M := \mathcal{M}(\Psi)$, and $L := \mathcal{L}(\Psi)$. Recall that the number of nodes in layers $0, \ldots, L$ is denoted by $N_0, \ldots, N_L$ and that $N_0 = d, N_L = d'$ (see Definition II.1). Moreover, note that due to our nondegeneracy assumption (see Remark II.2) we have $\sum_{\ell=0}^{L} N_\ell \leq 2M$ and $L \leq M$. The bitstring representing Ψ is constructed according to the following steps.

Step 1: If $M = 0$, we encode the network by a single 0. Using the convention $0\log(0) = 0$, we then note that (28) holds trivially and we terminate the encoding procedure. Else, we encode the network connectivity, M, by starting the overall bitstring with M many 1's followed by a single 0. The length of this bitstring is therefore given by $M + 1$.

Step 2: We continue by encoding the number of layers which, due to $L \leq M$, requires no more than $\lceil \log(M) \rceil$ bits. We thus reserve the next $\lceil \log(M) \rceil$ bits for the binary representation of L.

Step 3: Next, we store the layer dimensions $N_0, \dots, N_L$. As $L \leq M$ and $N_\ell \leq M$, for all $\ell \in \{0, \dots, L\}$, owing to nondegeneracy, we can encode the layer dimensions using $(M + 1)\lceil \log(M) \rceil$ bits. In combination with Steps 1 and 2 this yields an overall bitstring of length at most

$$M\lceil \log(M) \rceil + M + 2\lceil \log(M) \rceil + 1. \tag{29}$$

Step 4: We encode the topology of the graph associated with the network Ψ. To this end, we enumerate all nodes by assigning a unique index i to each one of them, starting from the 0-th layer and increasing from left to right within a given layer. The indices range from 1 to $N := \sum_{\ell=0}^{L} N_\ell \leq 2M$. Each of these indices can be encoded by a bitstring of length $\lceil \log(N) \rceil$. We denote the bitstring corresponding to index i by $b(i) \in \{0, 1\}^{\lceil \log(N) \rceil}$ and let for all nodes, except for those in the last layer, $n(i)$ be the number of children of the node with index i, i.e., the number of nodes in the next layer connected to the node with index i via an edge. For each of these nodes i, we form a bitstring of length $n(i)\lceil \log(N) \rceil$ by concatenating the bitstrings indexing its children. We follow this string with an all-zeros bitstring of length $\lceil \log(N) \rceil$ to signal that all children of the current node have been encoded. Overall, this yields a bitstring of length

$$\sum_{i=1}^{N-d'} (n(i) + 1)\lceil \log(N) \rceil \leq 3M\lceil \log(2M) \rceil, \tag{30}$$

where we used $\sum_{i=1}^{N-d'} n(i) \leq M$.

Step 5: We encode the weights of Ψ. By assumption, Ψ has $(\lceil \pi(\log(\varepsilon^{-1})) \rceil, \varepsilon)$-quantized weights, which means that each weight of Ψ can be represented by no more than $B_\varepsilon := 2(\lceil \pi(\log(\varepsilon^{-1})) \rceil \lceil \log(\varepsilon^{-1}) \rceil + 1)$ bits. For each node $i = 1, \dots, N$, we reserve the first B_ε bits to encode its associated node weight and, for each of its children a bitstring of length B_ε to encode the weight corresponding to the edge between the current node and that child. Concatenating the results in ascending order of child node indices, we get a bitstring of length $(n(i) + 1)B_\varepsilon$ for node i, and an overall bitstring of length

$$\sum_{i=1}^{N-d'} (n(i) + 1)B_\varepsilon + d'B_\varepsilon \leq 3MB_\varepsilon$$

representing the weights. Combining this with (29) and (30), we find that the overall number of bits needed to encode the network architecture, topology, and weights is no more than

$$3MB_\varepsilon + 3M\lceil \log(2M) \rceil + (M + 2)\lceil \log(M) \rceil + M + 1. \tag{31}$$

The network can be recovered by sequentially reading out M, L, the N_ℓ, the topology, and the quantized weights from the overall bitstring. It is not difficult to verify that the individual steps in the encoding procedure were crafted such that this yields unique recovery. As (31) can be upper-bounded by

$$C_0(M\log(M)+1)(\log(\varepsilon^{-1}))^q$$

for constants $C_0, q > 0$ depending on π only, we have constructed an encoder-decoder pair $(E_\varepsilon, D_\varepsilon) \in \mathfrak{E}^{\ell(\varepsilon)} \times \mathfrak{D}^{\ell(\varepsilon)}$ with $\ell(\varepsilon)$ satisfying (28). This concludes the proof. □

Proposition VI.7 states that the connectivity growth rate of networks with quantized weights achieving uniform approximation error ε over a function class $\mathcal{C}$ must exceed $\mathcal{O}\big(\varepsilon^{-1/\gamma^*(\mathcal{C})}\big)$, $\varepsilon \to 0$. As Proposition VI.7 applies to networks that have each weight represented by a finite number of bits scaling polynomially in $\log(\varepsilon^{-1})$, while guaranteeing that the underlying encoder-decoder pair achieves uniform error ε over $\mathcal{C}$, it remains to establish that such a compatibility is, indeed, possible. Specifically, this requires a careful interplay between the network's depth and connectivity scaling, and its weight growth, all as a function of ε. Establishing that this delicate balancing is implied by our technical assumptions is the subject of the remainder of this section. We start with a perturbation result quantifying how the error induced by weight quantization in the network translates to the output function realized by the network.

Lemma VI.8. *Let $d, d', k \in \mathbb{N}$, $D \in \mathbb{R}_+$, $\Omega \subseteq [-D, D]^d$, $\varepsilon \in (0, 1/2)$, let $\Phi \in \mathcal{N}_{d,d'}$ with $\mathcal{M}(\Phi) \le \varepsilon^{-k}$, $\mathcal{B}(\Phi) \le \varepsilon^{-k}$, and let $m \in \mathbb{N}$ satisfy*

$$m \ge 3k\mathcal{L}(\Phi) + \log(\lceil D \rceil). \tag{32}$$

Then, there exists a network $\tilde{\Phi} \in \mathcal{N}_{d,d'}$ with (m,ε)-quantized weights satisfying

$$\sup_{x\in\Omega} \|\Phi(x) - \tilde{\Phi}(x)\|_\infty \le \varepsilon.$$

More specifically, the network $\tilde{\Phi}$ can be obtained simply by replacing every weight in Φ by a closest element in $2^{-m\lceil\log(\varepsilon^{-1})\rceil}\mathbb{Z} \cap [-\varepsilon^{-m}, \varepsilon^{-m}]$.

Proof of Theorem VI.8. We first consider the case $\mathcal{L}(\Phi) = 1$. Here, it follows from Definition II.1 that the network simply realizes an affine transformation and hence

$$\sup_{x\in\Omega} \|\Phi(x) - \tilde{\Phi}(x)\|_\infty \le \mathcal{M}(\Phi)\lceil D\rceil 2^{-m\lceil\log(\varepsilon^{-1})\rceil - 1} \le \varepsilon.$$

In the remainder of the proof, we can therefore assume that $\mathcal{L}(\Phi) \ge 2$. For simplicity of notation, we set $L := \mathcal{L}(\Phi), M := \mathcal{M}(\Phi)$, and, as usual, write

$$\Phi = W_L \circ \rho \circ W_{L-1} \circ \rho \circ \cdots \circ \rho \circ W_1$$

with $W_\ell(x) = A_\ell x + b_\ell$, $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$, and $b_\ell \in \mathbb{R}^{N_\ell}$. We now consider the partial networks $\Phi^\ell \colon \Omega \to \mathbb{R}^{N_\ell}$, $\ell \in \{1, 2, \dots, L-1\}$, given by

$$\Phi^\ell := \begin{cases} \rho \circ W_1, & \ell = 1 \\ \rho \circ W_2 \circ \rho \circ W_1, & \ell = 2 \\ \rho \circ W_\ell \circ \rho \circ W_{\ell-1} \circ \dots \circ \rho \circ W_1, & \ell = 3, \dots, L-1, \end{cases}$$

and set $\Phi^L := \Phi$. We hasten to add that we decided—for ease of exposition—to deviate from the convention used in Definition II.1 and to have the partial networks include the application of ρ at the end. Now, for $\ell \in \{1, 2, \dots, L\}$, let $\tilde{\Phi}^\ell$ be the (partial) network obtained by replacing all the entries of the A_ℓ and b_ℓ by a closest element in $2^{-m\lceil \log(\varepsilon^{-1}) \rceil}\, \mathbb{Z} \cap [-\varepsilon^{-m}, \varepsilon^{-m}]$. We denote these replacements by $\tilde{A}_\ell$ and $\tilde{b}_\ell$, respectively, and note that

$$\begin{aligned} \max_{i,j} |A_{\ell,i,j} - \tilde{A}_{\ell,i,j}| &\le \tfrac{1}{2}\, 2^{-m\lceil \log(\varepsilon^{-1}) \rceil} \le \tfrac{1}{2}\, \varepsilon^m, \\ \max_{i,j} |b_{\ell,i,j} - \tilde{b}_{\ell,i,j}| &\le \tfrac{1}{2}\, 2^{-m\lceil \log(\varepsilon^{-1}) \rceil} \le \tfrac{1}{2}\, \varepsilon^m. \end{aligned} \tag{33}$$

The proof will be effected by upper-bounding the error building up across layers as a result of this quantization. To this end, we define, for $\ell \in \{1, 2, \dots, L\}$, the error in the ℓ-th layer as

$$e_\ell := \sup_{x \in \Omega} \|\Phi^\ell(x) - \tilde{\Phi}^\ell(x)\|_\infty.$$

We further set $C_0 := \lceil D \rceil$ and $C_\ell := \max\{1, \sup_{x \in \Omega} \|\Phi^\ell(x)\|_\infty\}$. As each entry of the vector $\Phi^\ell(x) \in \mathbb{R}^{N_\ell}$ is obtained by applying[10] the 1-Lipschitz function ρ to the sum of a weighted sum of at most $N_{\ell-1}$ components of the vector $\Phi^{\ell-1}(x) \in \mathbb{R}^{N_{\ell-1}}$ and an affine component $b_{\ell,i}$, and $\mathcal{B}(\Phi) \le \varepsilon^{-k}$ by assumption, we have for all $\ell \in \{1, 2, \dots, L\}$,

$$C_\ell \le N_{\ell-1} \varepsilon^{-k} C_{\ell-1} + \varepsilon^{-k} \le (N_{\ell-1} + 1)\, \varepsilon^{-k} C_{\ell-1},$$

which implies, for all $\ell \in \{1, 2, \dots, L\}$, that

$$C_\ell \le C_0\, \varepsilon^{-k\ell} \prod_{i=0}^{\ell-1} (N_i + 1). \tag{34}$$

Next, note that the components $(\tilde{\Phi}^1(x))_i, i \in \{1, 2, \dots, N_1\}$, of the vector $\tilde{\Phi}^1(x) \in \mathbb{R}^{N_1}$ can be written as

$$(\tilde{\Phi}^1(x))_i = \rho\left(\left(\sum_{j=1}^{N_0} \tilde{A}_{1,i,j} x_j \right) + \tilde{b}_{1,i} \right),$$

which, combined with (33) and the fact that ρ is 1-Lipschitz implies

$$e_1 \le C_0 N_0 \tfrac{\varepsilon^m}{2} + \tfrac{\varepsilon^m}{2} \le C_0 (N_0 + 1) \tfrac{\varepsilon^m}{2}. \tag{35}$$

[10] Note that going from Φ_{L-1} to Φ_L the activation function is not applied anymore, which nevertheless leads to the same estimate as the identity mapping is 1-Lipschitz.

Due to ρ and the identity mapping being 1-Lipschitz, we have, for $\ell = 1, \dots, L$,

$$\begin{aligned} e_\ell &= \sup_{x \in \Omega} \|\Phi^\ell(x) - \tilde{\Phi}^\ell(x)\|_\infty = \sup_{x \in \Omega, i \in \{1,\dots,N_l\}} |(\Phi^\ell(x))_i - (\tilde{\Phi}^\ell(x))_i| \\ &\leq \sup_{x \in \Omega, i \in \{1,\dots,N_\ell\}} \left| \left[\left(\sum_{j=1}^{N_{\ell-1}} A_{\ell,i,j} (\Phi^{\ell-1}(x))_j \right) + b_{\ell,i} \right] - \left[\left(\sum_{j=1}^{N_{\ell-1}} \tilde{A}_{\ell,i,j} (\tilde{\Phi}^{\ell-1}(x))_j \right) + \tilde{b}_{\ell,i} \right] \right| \\ &\leq \sup_{x \in \Omega, i \in \{1,\dots,N_\ell\}} \left[\left(\sum_{j=1}^{N_{\ell-1}} \left| A_{\ell,i,j} (\Phi^{\ell-1}(x))_j - \tilde{A}_{\ell,i,j} (\tilde{\Phi}^{\ell-1}(x))_j \right| \right) + \left| b_{\ell,i} - \tilde{b}_{\ell,i} \right| \right]. \end{aligned} \tag{36}$$

As $|(\Phi^{\ell-1}(x))_j - (\tilde{\Phi}^{\ell-1}(x))_j| \leq e_{\ell-1}$ and $|(\Phi^{\ell-1}(x))_j| \leq C_{\ell-1}$ for all $x \in \Omega$, $j \in \{1, \dots, N_{\ell-1}\}$ by definition, and $|A_{\ell,i,j}| \leq \varepsilon^{-k}$ by assumption, upon invoking (33), we get

$$|A_{\ell,i,j} (\Phi^{\ell-1}(x))_j - \tilde{A}_{\ell,i,j} (\tilde{\Phi}^{\ell-1}(x))_j| \leq e_{\ell-1} \varepsilon^{-k} + C_{\ell-1} \tfrac{\varepsilon^m}{2} + e_{\ell-1} \tfrac{\varepsilon^m}{2}.$$

Since $\varepsilon \in (0, 1/2)$, it therefore follows from (36), that for all $\ell \in \{2, \dots, L\}$,

$$e_\ell \leq N_{\ell-1} (e_{\ell-1} \varepsilon^{-k} + C_{\ell-1} \tfrac{\varepsilon^m}{2} + e_{\ell-1} \tfrac{\varepsilon^m}{2}) + \tfrac{\varepsilon^m}{2} \leq (N_{\ell-1} + 1)(2 e_{\ell-1} \varepsilon^{-k} + C_{\ell-1} \tfrac{\varepsilon^m}{2}). \tag{37}$$

We now claim that, for all $\ell \in \{2, \dots, L\}$,

$$e_\ell \leq \tfrac{1}{2} (2^\ell - 1) C_0 \varepsilon^{m - (\ell-1)k} \prod_{i=0}^{\ell-1} (N_i + 1), \tag{38}$$

which we prove by induction. The base case $\ell = 1$ was already established in (35). For the induction step we assume that (38) holds for a given ℓ which, in combination with (34) and (37), implies

$$\begin{aligned} e_{\ell+1} &\leq \big(N_\ell + 1)(2 e_\ell \varepsilon^{-k} + C_\ell \tfrac{\varepsilon^m}{2}\big) \\ &\leq (N_\ell + 1) \left((2^\ell - 1) C_0 \varepsilon^{m - (\ell-1)k} \varepsilon^{-k} \prod_{i=0}^{\ell-1} (N_i + 1) + C_0 \varepsilon^{-k\ell} \tfrac{\varepsilon^m}{2} \prod_{i=0}^{\ell-1} (N_i + 1) \right) \\ &= \frac{1}{2} (2^{\ell+1} - 1) C_0 \varepsilon^{m - \ell k} \prod_{i=0}^{\ell} (N_i + 1). \end{aligned}$$

This completes the induction argument and establishes (38). Using $2^{L-1} \leq \varepsilon^{-(L-1)}$, $\prod_{i=0}^{L-1} (N_i + 1) \leq M^L \leq \varepsilon^{-kL}$, and $m \geq 3kL + \log(\lceil D \rceil)$ by assumption, we get

$$\begin{aligned} \sup_{x \in \Omega} \|\Phi(x) - \tilde{\Phi}(x)\|_\infty = e_L &\leq \tfrac{1}{2} (2^L - 1) C_0 \varepsilon^{m - (L-1)k} \prod_{i=0}^{L-1} (N_i + 1) \\ &\leq \varepsilon^{m - (L - 1 + kL - k + \log(\lceil D \rceil) + kL)} \\ &\leq \varepsilon^{m - (3kL + \log(\lceil D \rceil) - 1)} \leq \varepsilon. \end{aligned}$$

This completes the proof. □

We are now ready to finalize the proof of Theorem VI.4.

Proof of Theorem VI.4. Suppose towards a contradiction that $\gamma_{\mathcal{N}}^{*,\text{eff}}(\mathcal{C}) > \gamma^*(\mathcal{C})$ and let $\gamma \in \big(\gamma^*(\mathcal{C}), \gamma_{\mathcal{N}}^{*,\text{eff}}(\mathcal{C})\big)$. Then, by Definition VI.3, there exist a polynomial π and a constant $C > 0$ such that

$$\sup_{f\in\mathcal{C}} \inf_{\Phi\in\mathcal{N}^{\pi}_{M,d,1}} \|f-\Phi\|_{L^2(\Omega)} \le CM^{-\gamma}, \text{ for all } M \in \mathbb{N}.$$

Setting $M_\varepsilon := \big\lceil (\varepsilon/(4C))^{-1/\gamma} \big\rceil$, it follows that, for every $f \in \mathcal{C}$ and every $\varepsilon \in (0, 1/2)$, there exists a neural network $\Phi_{\varepsilon,f} \in \mathcal{N}^{\pi}_{M_\varepsilon,d,1}$ such that

$$\|f - \Phi_{\varepsilon,f}\|_{L^2(\Omega)} \le 2 \sup_{f\in\mathcal{C}} \inf_{\Phi\in\mathcal{N}^{\pi}_{M_\varepsilon,d,1}} \|f-\Phi\|_{L^2(\Omega)} \le 2CM_\varepsilon^{-\gamma} \le \frac{\varepsilon}{2}. \tag{39}$$

By Lemma VI.8 there exists a polynomial π^* such that for every $f \in \mathcal{C}$, $\varepsilon \in (0, 1/2)$, there is a network $\widetilde{\Phi}_{\varepsilon,f}$ with $(\lceil \pi^*(\log(\varepsilon^{-1}))\rceil, \varepsilon)$-quantized weights satisfying

$$\left\|\Phi_{\varepsilon,f} - \widetilde{\Phi}_{\varepsilon,f}\right\|_{L^2(\Omega)} \le \frac{\varepsilon}{2}. \tag{40}$$

The conditions of Lemma VI.8 are satisfied as M_ε can be upper-bounded by ε^{-k} with a suitably chosen k, the weights in $\Phi_{\varepsilon,f}$ are polynomially bounded in M_ε, and (32) follows from the depth of networks in $\Phi \in \mathcal{N}^{\pi}_{M_\varepsilon,d,1}$ being polylogarithmically bounded in M_ε due to Definition VI.2. Now, defining

$$\Psi\colon \ \big(0, \tfrac{1}{2}\big) \times \mathcal{C} \to \mathcal{N}_{d,1}, \quad (\varepsilon, f) \mapsto \widetilde{\Phi}_{\varepsilon,f},$$

it follows from (39) and (40), by application of the triangle inequality, that

$$\sup_{f\in\mathcal{C}} \|f - \Psi(\varepsilon, f)\|_{L^2(\Omega)} \le \varepsilon \quad \text{ with } \quad \sup_{f\in\mathcal{C}} \mathcal{M}(\Psi(\varepsilon, f)) \le M_\varepsilon \in \mathcal{O}\big(\varepsilon^{-1/\gamma}\big),\ \varepsilon \to 0.$$

The proof is concluded by noting that $\Psi(\varepsilon, f)$ violates Proposition VI.7. □

We conclude this section with a discussion of the conceptual implications of the results established above. Proposition VI.7 combined with Lemma VI.8 establishes that neural networks achieving uniform approximation error ε while having weights that are polynomially bounded in ε^{-1} and depth growing polylogarithmically in ε^{-1} cannot exhibit connectivity growth rate smaller than $\mathcal{O}(\varepsilon^{-1/\gamma^*(\mathcal{C})}), \varepsilon \to 0$; in other words, a decay of the uniform approximation error, as a function of M, faster than $\mathcal{O}(M^{-\gamma^*(\mathcal{C})}), M \to \infty$, is not possible.

VII. The Transference Principle

We have seen that a wide array of function classes can be approximated in Kolmogorov-Donoho optimal fashion through dictionaries, provided that the dictionary $\mathcal{D}$ is chosen to consort with the function class $\mathcal{C}$ according to $\gamma^{*,\text{eff}}(\mathcal{C}, \mathcal{D}) = \gamma^*(\mathcal{C})$. Examples of such pairs are unit balls in Besov spaces with wavelet bases and unit balls in weighted modulation spaces with Wilson bases. A more extensive list of optimal pairs is provided in Table 1. On the other hand, as shown in [14], Fourier bases are strictly suboptimal—in terms of approximation rate—for balls $\mathcal{C}$ of finite radius in the spaces $BV(\mathbb{R})$ and $W_p^m(\mathbb{R})$.

In light of what was just said, it is hence natural to let neural networks play the role of the dictionary $\mathcal{D}$ and to ask which function classes $\mathcal{C}$ are approximated in Kolmogorov-Donoho-optimal fashion by neural networks. Towards answering this question, we next develop a general framework for transferring results on function approximation through dictionaries to results on approximation by neural networks. This will eventually lead us to a characterization of function classes $\mathcal{C}$ that are optimally representable by neural networks in the sense of Definition VI.5.

We start by introducing the notion of effective representability of dictionaries through neural networks.

Definition VII.1. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$ be a dictionary. We call $\mathcal{D}$* effectively representable by neural networks, *if there exists a bivariate polynomial π such that for all $i \in \mathbb{N}$, $\varepsilon \in (0, 1/2)$, there is a neural network $\Phi_{i,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying $\mathcal{M}(\Phi_{i,\varepsilon}) \leq \pi(\log(\varepsilon^{-1}), \log(i))$, $\mathcal{B}(\Phi_{i,\varepsilon}) \leq \pi(\varepsilon^{-1}, i)$, and*

$$\|\varphi_i - \Phi_{i,\varepsilon}\|_{L^2(\Omega)} \leq \varepsilon.$$

The next result will allow us to conclude that optimality—in the sense of Definition V.4—of a dictionary $\mathcal{D}$ for a function class $\mathcal{C}$ combined with effective representability of $\mathcal{D}$ by neural networks implies optimal representability of $\mathcal{C}$ by neural networks. The proof is, in essence, effected by noting that every element of the effectively representable $\mathcal{D}$ participating in a best M-term-rate achieving approximation f_M of $f \in \mathcal{C}$ can itself be approximated by neural networks well enough for an overall network to approximate f_M with connectivity $M\pi(\log(M))$. As this connectivity is only polylogarithmically larger than the number of terms M participating in the best M-term approximation f_M, we will be able to conclude that the optimal approximation rate, indeed, transfers from approximation in $\mathcal{D}$ to approximation in neural networks. The conditions on $\mathcal{M}(\Phi_{i,\varepsilon})$ and $\mathcal{B}(\Phi_{i,\varepsilon})$ in Definition VII.1 guarantee precisely that the connectivity increase is at most by a polylogarithmic factor. To see this, we first recall that effective best M-term approximation has a polynomial depth search constraint, which implies that the indices i under consideration are upper-bounded by a polynomial in M. In addition, the approximation error behavior we are interested in is $\varepsilon = M^{-\gamma}$. Combining these two insights, it follows that $\mathcal{M}(\Phi_{i,\varepsilon}) \leq \pi(\log(\varepsilon^{-1}), \log(i))$ implies polylogarithmic (in M) connectivity for each network $\Phi_{i,\varepsilon}$ and hence connectivity $M\pi(\log(M))$ for the overall network realizing f_M, as desired. By the same token, $\mathcal{B}(\Phi_{i,\varepsilon}) \leq \pi(\varepsilon^{-1}, i)$ guarantees that the weights of $\Phi_{i,\varepsilon}$ are polynomial in M.

There is another aspect to effective representability by neural networks that we would like to illustrate by way of example, namely that of ordering the dictionary elements. Specifically, we consider, for $d = 1$ and $\Omega = [-\pi, \pi)$, the class $\mathcal{C}$ of real-valued even functions in $\mathcal{C} = L^2(\Omega)$, and take the dictionary as $\mathcal{D} = \{\cos(ix), i \in \mathbb{N}_0\}$. As the index i enumerating the dictionary elements corresponds to frequencies, the basis functions in $\mathcal{D}$ are hence ordered according to increasing frequencies. Next, note that the parameter a in Theorem III.8 corresponds to the frequency index i in our example. As the network $\Psi_{a,D,\varepsilon}$ in Theorem III.8 is of finite width, it hence follows, upon replacing a in the expression for $\mathcal{L}(\Psi_{a,D,\varepsilon})$ by i, that $\mathcal{M}(\Psi_{i,D,\varepsilon}) \leq \pi(\log(\varepsilon^{-1}), \log(i))$. The condition on the weights for effective representability is satisfied trivially, simply as $\mathcal{B}(\Psi_{i,D,\varepsilon}) \leq 1 \leq \pi(\varepsilon^{-1}, i)$.

We are now ready to state the rate optimality transfer result.

Theorem VII.2. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$ be bounded, and consider the function class $\mathcal{C} \subseteq L^2(\Omega)$. Suppose that the dictionary $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$ is effectively representable by neural networks. Then, for every $0 < \gamma < \gamma^{*,\mathit{eff}}(\mathcal{C},\mathcal{D})$, there exist a polynomial π and a map*

$$\Psi : \left(0, \tfrac{1}{2}\right) \times \mathcal{C} \to \mathcal{N}_{d,1},$$

such that for all $f \in \mathcal{C}$, $\varepsilon \in (0, 1/2)$, the network $\Psi(\varepsilon, f)$ has $(\lceil \pi(\log(\varepsilon^{-1}))\rceil, \varepsilon)$-quantized weights while satisfying $\|f - \Psi(\varepsilon, f)\|_{L^2(\Omega)} \leq \varepsilon$, $\mathcal{L}(\Psi(\varepsilon, f)) \leq \pi(\log(\varepsilon^{-1}))$, $\mathcal{B}(\Psi(\varepsilon, f)) \leq \pi(\varepsilon^{-1})$, and we have

$$\mathcal{M}(\Psi(\varepsilon, f)) \in \mathcal{O}(\varepsilon^{-1/\gamma}),\ \varepsilon \to 0, \tag{41}$$

with the implicit constant in (41) being independent of f. In particular, it holds that

$$\gamma^{*,\mathit{eff}}_{\mathcal{N}}(\mathcal{C}) \geq \gamma^{*,\mathit{eff}}(\mathcal{C},\mathcal{D}).$$

Remark VII.3. *Theorem VII.2 allows us to draw the following conclusion. If $\mathcal{D}$ optimally represents the function class $\mathcal{C}$ in the sense of Definition V.4, i.e., $\gamma^{*,\mathit{eff}}(\mathcal{C},\mathcal{D}) = \gamma^*(\mathcal{C})$, and if it is, in addition, effectively representable by neural networks in the sense of Definition VII.1, then, due to Theorem VI.4, which states that $\gamma^{*,\mathit{eff}}_{\mathcal{N}}(\mathcal{C}) \leq \gamma^*(\mathcal{C})$, we have $\gamma^{*,\mathit{eff}}_{\mathcal{N}}(\mathcal{C}) = \gamma^*(\mathcal{C})$ and hence $\mathcal{C}$ is optimally representable by neural networks in the sense of Definition VI.5.*

Proof of Theorem VII.2. Let $\gamma' \in (\gamma, \gamma^{*,\mathrm{eff}}(\mathcal{C},\mathcal{D}))$. According to Definition V.2, there exist a constant $C \geq 1$ and a polynomial π_1, such that for every $f \in \mathcal{C}$, $M \in \mathbb{N}$, there is an index set $I_{f,M} \subseteq \{1, \ldots, \pi_1(M)\}$ of cardinality M and coefficients $(c_i)_{i\in I_{f,M}}$ with $|c_i| \leq \pi_1(M)$, such that

$$\left\| f - \sum_{i\in I_{f,M}} c_i\varphi_i \right\|_{L^2(\Omega)} \leq \frac{CM^{-\gamma'}}{2}. \tag{42}$$

Let $A := \max\{1, |\Omega|^{1/2}\}$. Effective representability of $\mathcal{D}$ according to Definition VII.1 ensures the existence of a bivariate polynomial π_2 such that for all $M \in \mathbb{N}$, $i \in I_{f,M}$, there is a neural network $\Phi_{i,M} \in \mathcal{N}_{d,1}$ satisfying

$$\|\varphi_i - \Phi_{i,M}\|_{L^2(\Omega)} \leq \tfrac{C}{4A\pi_1(M)} M^{-(\gamma'+1)} \tag{43}$$

with

$$\begin{aligned}
\mathcal{M}(\Phi_{i,M}) &\leq \pi_2\left(\log\left(\left(\tfrac{C}{4A\pi_1(M)}M^{-(\gamma'+1)}\right)^{-1}\right), \log(i)\right)\\
&= \pi_2\left((\gamma'+1)\log(M) + \log\left(\tfrac{4A\pi_1(M)}{C}\right), \log(i)\right),\\
\mathcal{B}(\Phi_{i,M}) &\leq \pi_2\left(\left(\tfrac{C}{4A\pi_1(M)}M^{-(\gamma'+1)}\right)^{-1}, i\right) = \pi_2\left(\tfrac{4A\pi_1(M)}{C}M^{\gamma'+1}, i\right).
\end{aligned} \tag{44}$$

Consider now for $f \in \mathcal{C}$, $M \in \mathbb{N}$ the networks given by

$$\Psi_{f,M}(x) := \sum_{i\in I_{f,M}} c_i\Phi_{i,M}(x).$$

Due to $\max(I_{f,M}) \leq \pi_1(M)$, (44) and Lemma A.8 imply the existence of a polynomial π_3 such that $\mathcal{L}(\Psi_{f,M}) \leq \pi_3(\log(M))$, $\mathcal{M}(\Psi_{f,M}) \leq M\pi_3(\log(M))$, and $\mathcal{B}(\Psi_{f,M}) \leq \pi_3(M)$, for all $f \in \mathcal{C}$, $M \in \mathbb{N}$, and, owing to (43), we get

$$\left\|\Psi_{f,M} - \sum_{i\in I_{f,M}} c_i\varphi_i\right\|_{L^2(\Omega)} \leq \sum_{i\in I_{f,M}} |c_i| \tfrac{C}{4A\pi_1(M)} M^{-(\gamma'+1)} \leq \tfrac{CM^{-\gamma'}}{4A} \sum_{i=1}^{|I_{f,M}|} \tfrac{\max_{i\in I_{f,M}}|c_i|}{M\pi_1(M)} \leq \tfrac{CM^{-\gamma'}}{4A}. \tag{45}$$

Lemma VI.8 therefore ensures the existence of a polynomial π_4 such that for all $f \in \mathcal{C}$, $M \in \mathbb{N}$, there is a network $\widetilde{\Psi}_{f,M} \in \mathcal{N}_{d,1}$ with $(\lceil \pi_4(\log(\frac{4A}{C}M^{\gamma'}))\rceil, \frac{CM^{-\gamma'}}{4A})$-quantized weights satisfying $\mathcal{L}(\widetilde{\Psi}_{f,M}) = \mathcal{L}(\Psi_{f,M})$, $\mathcal{M}(\widetilde{\Psi}_{f,M}) = \mathcal{M}(\Psi_{f,M})$, $\mathcal{B}(\widetilde{\Psi}_{f,M}) \leq \mathcal{B}(\Psi_{f,M}) + \frac{CM^{-\gamma'}}{4A}$, and

$$\left\|\Psi_{f,M} - \widetilde{\Psi}_{f,M}\right\|_{L^\infty(\Omega)} \leq \tfrac{CM^{-\gamma'}}{4A}. \tag{46}$$

As Ω is bounded by assumption, we have

$$\left\|\Psi_{f,M} - \widetilde{\Psi}_{f,M}\right\|_{L^2(\Omega)} \leq |\Omega|^{\frac{1}{2}} \left\|\Psi_{f,M} - \widetilde{\Psi}_{f,M}\right\|_{L^\infty(\Omega)} \leq \tfrac{CM^{-\gamma'}}{4}, \tag{47}$$

for all $f \in \mathcal{C}$, $M \in \mathbb{N}$. Combining (47) with (42) and (45), we get, for all $f \in \mathcal{C}$, $M \in \mathbb{N}$,

$$\begin{aligned} \left\|f - \widetilde{\Psi}_{f,M}\right\|_{L^2(\Omega)} &\leq \left\|f - \sum_{i\in I_{f,M}} c_i\varphi_i\right\|_{L^2(\Omega)} + \left\|\sum_{i\in I_{f,M}} c_i\varphi_i - \Psi_{f,M}\right\|_{L^2(\Omega)} + \left\|\Psi_{f,M} - \widetilde{\Psi}_{f,M}\right\|_{L^2(\Omega)} \\ &\leq CM^{-\gamma'}. \end{aligned} \tag{48}$$

For $\varepsilon \in (0, 1/2)$ and $f \in \mathcal{C}$, we now set $M_\varepsilon := \left\lceil (C/\varepsilon)^{1/\gamma'} \right\rceil$ and

$$\Psi(\varepsilon, f) := \widetilde{\Psi}_{f,M_\varepsilon}.$$

Thus, (48) yields

$$\|f - \Psi(\varepsilon, f)\|_{L^2(\Omega)} \leq CM_\varepsilon^{-\gamma'} \leq \varepsilon.$$

Next, we note that, for all polynomials π and $0 \leq m < n$,

$$\mathcal{O}(\varepsilon^{-m}\pi(\log(\varepsilon^{-1}))) \subseteq \mathcal{O}(\varepsilon^{-n}),\ \varepsilon \to 0.$$

As $1/\gamma' < 1/\gamma$, this establishes

$$\mathcal{M}(\Psi(\varepsilon, f)) \in \mathcal{O}(M_\varepsilon \pi_3(\log(M_\varepsilon))) \subseteq \mathcal{O}(\varepsilon^{-1/\gamma}),\ \varepsilon \to 0. \tag{49}$$

Since M_ε and π_3 are independent of f, the implicit constant in (49) does not depend on f.

Next, note that, in general, an (n, η)-quantized network is also (m, δ)-quantized for $n \geq m$ and $\eta \leq \delta$, simply as

$$2^{-m\lceil \log(\delta^{-1})\rceil}\mathbb{Z} \cap [-\delta^{-m}, \delta^{-m}] \subseteq 2^{-n\lceil \log(\eta^{-1})\rceil}\mathbb{Z} \cap [-\eta^{-n}, \eta^{-n}].$$

Since $\frac{CM_\varepsilon^{-\gamma'}}{4A} \leq \varepsilon$ this ensures the existence of a polynomial π such that, for every $f \in \mathcal{C}$, $\varepsilon \in (0, 1/2)$, the network $\Psi(\varepsilon, f)$ is $(\lceil \pi(\log(\varepsilon^{-1})) \rceil, \varepsilon)$-quantized, $\mathcal{L}(\Psi(\varepsilon, f)) \leq \pi(\log(\varepsilon^{-1}))$, and $\mathcal{B}(\Psi(\varepsilon, f)) \leq \pi(\varepsilon^{-1})$. With (49) this establishes the first claim of the theorem. In order to verify the second claim, note that $\Psi(\varepsilon, f) \in \mathcal{N}^{\pi}_{\mathcal{M}(\Psi(\varepsilon,f)),d,1}$, for all $f \in \mathcal{C}$, $\varepsilon \in (0, 1/2)$, which implies

$$\sup_{f\in\mathcal{C}} \inf_{\Phi\in\mathcal{N}^{\pi}_{M,d,1}} \|f - \Phi\|_{L^2(\Omega)} \in \mathcal{O}(M^{-\gamma}),\ M \to \infty.$$

Therefore, owing to Definition VI.3, we get

$$\gamma^{*,\mathrm{eff}}_{\mathcal{N}}(\mathcal{C}) \geq \gamma^{*,\mathrm{eff}}(\mathcal{C}, \mathcal{D}),$$

which concludes the proof. □

Remark VII.4. *We note that Theorem VII.2 continues to hold for $\Omega = \mathbb{R}^n$ if the elements of $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}}$ are compactly supported with the size of their support sets growing no more than polynomially in i. The technical elements required to show this can be found in the context of the approximation of Gabor dictionaries in the proof of Theorem IX.3, but are omitted here for ease of exposition.*

The last piece needed to complete our program is to establish that the conditions in Definition VII.1 guaranteeing effective representability in neural networks are, indeed, satisfied by a wide variety of dictionaries.

Inspecting Table 1, we can see that all example function classes provided therein are optimally represented either by affine dictionaries, i.e., wavelets, the Haar basis, and curvelets or Weyl-Heisenberg dictionaries, namely Fourier bases and Wilson bases. The next two sections will be devoted to proving effective representability of affine dictionaries and Weyl-Heisenberg dictionaries by neural networks, thus allowing us to draw the conclusion that neural networks are universally Kolmogorov-Donoho optimal approximators for all function classes listed in Table 1.

VIII. Affine Dictionaries are Effectively Representable by Neural Networks

The purpose of this section is to establish that *affine dictionaries*, including wavelets [70], ridgelets [39], curvelets [71], shearlets [72], α-shearlets and more generally α-molecules [69], which contain all aforementioned dictionaries as special cases, are effectively representable by neural networks. Due to Theorem VII.2 and Theorem VI.4, this will then allow us to conclude that any function class that is optimally representable—in the sense of Definition V.4—by an affine dictionary with a suitable generator function is optimally representable by neural networks in the sense of Definition VI.5. By “suitable” we mean that the generator function can be approximated well by ReLU networks in a sense to be made precise below.

In order to elucidate the main ideas underlying the general definition of affine dictionaries that are effectively representable by neural networks, we start with a basic example, namely the Haar wavelet dictionary on the unit interval, i.e., the set of functions

$$\psi_{n,k}\colon [0,1] \mapsto \mathbb{R},\ x \mapsto 2^{\frac{n}{2}}\psi(2^n x - k),\ n \in \mathbb{N}_0,\, k = 0, \dots, 2^n - 1,$$

with

$$\psi\colon \mathbb{R} \to \mathbb{R},\ x \mapsto \begin{cases} 1, & x \in [0, 1/2) \\ -1, & x \in [1/2, 1) \\ 0, & \text{else.} \end{cases}$$

We approximate the piecewise constant mother wavelet ψ through a continuous piecewise linear function realized by a neural network as follows

$$\Psi_\delta(x) := \tfrac{1}{2\delta}\rho(x+\delta) - \tfrac{1}{2\delta}\rho(x-\delta) - \tfrac{1}{\delta}\rho(x-(\tfrac{1}{2}-\delta)) + \tfrac{1}{\delta}\rho(x-(\tfrac{1}{2}+\delta)) + \tfrac{1}{2\delta}\rho(x-(1-\delta)) - \tfrac{1}{2\delta}\rho(x-(1+\delta))$$

and, setting $\delta(\varepsilon) := \varepsilon^2$ for $\varepsilon \in (0, 1/2)$, let

$$\Phi_{n,k,\varepsilon}(x) := 2^{\frac{n}{2}}\Psi_{\delta(\varepsilon)}(2^n x - k),\ n \in \mathbb{N}_0,\, k = 0, \dots, 2^n - 1.$$

The basic idea in the approximation of ψ through Ψ_δ is to let the transition regions around $0, 1/2$, and 1 shrink, as a function of ε, sufficiently fast for the construction to realize an approximation error of no more than ε. Now, a direct calculation yields that, indeed, for $\varepsilon \in (0, 1/2)$,

$$\|\psi_{n,k} - \Phi_{n,k,\varepsilon}\|_{L^2([0,1])} \leq \varepsilon.$$

Moreover, we have $\mathcal{M}(\Phi_{n,k,\varepsilon}) = 18$ and $\mathcal{B}(\Phi_{n,k,\varepsilon}) \leq \max\{2^{\frac{n}{2}}\varepsilon^{-2}, 2^n\}$. In order to establish effective representability by neural networks, we need to order the Haar wavelet dictionary suitably. Specifically, we proceed from coarse to fine scales, i.e., we let $(\varphi_i)_{i\in\mathbb{N}} = \mathcal{D} = \{\mathcal{D}_0, \mathcal{D}_1, \dots\}$, with $\mathcal{D}_n := \{\psi_{n,k} \mapsto \mathbb{R}\colon k = 0, \dots, 2^n - 1\}$, where the ordering within the $\mathcal{D}_n$ may be chosen arbitrarily. Next, note that for every pair $n \in \mathbb{N}_0$, $k \in \{0, \dots, 2^n - 1\}$, there exists a unique index $i \in \mathbb{N}$ such that $\varphi_i = \psi_{n,k} = \psi_{n(i),k(i)}$ and, owing to $|\mathcal{D}_n| = 2^n$, we have $2^{n(i)} \leq i$. Finally, taking $\Phi_{i,\varepsilon} := \Phi_{n(i),k(i),\varepsilon}$ and $\pi(a,b) := a^2 b + b + 18$, the conditions in Definition VII.1 for effective representability by neural networks are readily verified. A more elaborate example, namely spline wavelets, is considered at the end of this section.

We are now ready to proceed to the general definition of affine dictionaries with canonical ordering.

A. Affine Dictionaries with Canonical Ordering

Definition VIII.1. *Let $d, S \in \mathbb{N}$, $\delta > 0$, $\Omega \subseteq \mathbb{R}^d$ be bounded, and let $g_s \in L^\infty(\mathbb{R}^d)$, $s \in \{1, \dots, S\}$, be compactly supported. Furthermore, for $s \in \{1, \dots, S\}$, let $J_s \subseteq \mathbb{N}$ and $A_{s,j} \in \mathbb{R}^{d \times d}$, $j \in J_s$, be full-rank and with eigenvalues bounded below by 1 in absolute value. We define the* affine dictionary *$\mathcal{D} \subseteq L^2(\Omega)$ with generator functions $(g_s)_{s=1}^S$ as*

$$\mathcal{D} := \left\{ g_s^{j,e} := \left(|\det(A_{s,j})|^{\frac{1}{2}} g_s(A_{s,j} \cdot - \delta e) \right) |_{\Omega} \colon \ s \in \{1, \dots, S\}, \ e \in \mathbb{Z}^d, \ j \in J_s, \ \text{and } g_s^{j,e} \neq 0 \right\}.$$

Moreover, we define the sub-dictionaries

$$\begin{aligned} \mathcal{D}_{s,j} &:= \{ g_s^{j,e} \in \mathcal{D} : e \in \mathbb{Z}^d \text{ and } g_s^{j,e} \neq 0 \}, \quad \text{for } j \in J_s, \ s \in \{1, \dots, S\} \\ \mathcal{D}_j &:= \bigcup_{s \in \{1, \dots, S\} \colon j \in J_s} \mathcal{D}_{s,j}, \quad \text{for } j \in \mathbb{N}. \end{aligned}$$

We call an affine dictionary canonically ordered if it is arranged according to

$$(\varphi_i)_{i \in \mathbb{N}} = \mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2, \dots), \tag{50}$$

where the elements within each $\mathcal{D}_j$ may be ordered arbitrarily, and there exist constants $a, c > 0$ such that

$$\sum_{k=1}^{j-1} |\det(A_{s,k})| \geq c \|A_{s,j}\|_\infty^a, \quad \textit{for all } j \in J_s \backslash \{1\}, \ s \in \{1, \dots, S\}. \tag{51}$$

We call an affine dictionary nondegenerate if for every $j \in J_s$, $s \in \{1, \dots, S\}$, the sub-dictionary $\mathcal{D}_{s,j}$ contains at least one element.

Note that for sake of greater generality, we associate possibly different sets $J_s \subseteq \mathbb{N}$ with the generator functions g_s and, in particular, also allow these sets to be finite. The Haar wavelet dictionary example above is recovered as a nondegenerate affine dictionary by taking $d = 1$, $\Omega = [0, 1]$, $S = 1$, $J_s = \mathbb{N}$, $g_1 = \psi$, $\delta = 1$, $A_{1,j} = 2^{j-1}$, $a = 1$, $c = 1/2$, and noting that nondegeneracy is verified as for scale j, the sub-dictionary $\mathcal{D}_{s,j}$ contains 2^{j-1} elements. Moreover, the weights of the networks approximating the individual Haar wavelet dictionary elements grow linearly in the index of the dictionary elements. This is a consequence of the weights being determined by the dilation factor 2^n and $2^{n(i)} \leq i$ due to the ordering we chose. As will be shown below, morally this continues to hold for general nondegenerate affine dictionaries, thereby revealing what informed our definition of canonical ordering. Besides, our notion of canonical ordering is also inspired by the ordering employed in the tail compactness considerations for Besov spaces and orthonormal wavelet dictionaries as detailed in Appendix B. We remark that (51) constitutes a very weak restriction on how fast the size of dilations may grow; in fact, we are not aware of any affine dictionaries in the literature that would violate this condition. Finally, we note that the dilations $A_{s,j}$ are not required to be ordered in ascending size, as was the case in the Haar wavelet dictionary example. Canonical ordering does, however, ensure a modicum of ordering.

B. Invariance to Affine Transformations

Affine dictionaries consist of dilations and translations of a given generator function. It is therefore important to understand the impact of these operations on the approximability—by neural networks—of a given function. As neural networks realize concatenations of affine functions and nonlinearities, it is clear that translations and dilations can be absorbed into the first layer of the network and the transformed function should inherit the approximability properties of the generator function. However, what we will have to understand is how the weights, the connectivity, and the domain of approximation of the resulting network are impacted. The following result makes this quantitative.

Proposition VIII.2. *Let $d \in \mathbb{N}$, $p \in [1, \infty]$, and $f \in L^p(\mathbb{R}^d)$. Assume that there exists a bivariate polynomial π such that for all $D \in \mathbb{R}_+$, $\varepsilon \in (0, 1/2)$, there is a network $\Phi_{D,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying*

$$\|f - \Phi_{D,\varepsilon}\|_{L^p([-D,D]^d)} \leq \varepsilon, \tag{52}$$

with $\mathcal{M}(\Phi_{D,\varepsilon}) \leq \pi(\log(\varepsilon^{-1}), \log(\lceil D \rceil))$. Then, for all full-rank matrices $A \in \mathbb{R}^{d \times d}$, and all $e \in \mathbb{R}^d$, $E \in \mathbb{R}_+$, and $\eta \in (0, 1/2)$, there is a network $\Psi_{A,e,E,\eta} \in \mathcal{N}_{d,1}$ satisfying

$$\left\| |\det(A)|^{\frac{1}{p}} f(A \cdot - e) - \Psi_{A,e,E,\eta} \right\|_{L^p([-E,E]^d)} \leq \eta,$$

with $\mathcal{M}(\Psi_{A,e,E,\eta}) \leq \pi'(\log(\eta^{-1}), \log(\lceil F \rceil))$ and $\mathcal{B}(\Psi_{A,e,E,\eta}) \leq \max\{\mathcal{B}(\Phi_{F,\eta}), |\det(A)|^{\frac{1}{p}}, \|A\|_\infty, \|e\|_\infty\}$, where $F = dE\|A\|_\infty + \|e\|_\infty$ and π' is of the same degree as π.

Proof. By a change of variables, we have for every $\Phi \in \mathcal{N}_{d,1}$,

$$\left\| |\det(A)|^{\frac{1}{p}} f(A \cdot - e) - |\det(A)|^{\frac{1}{p}} \Phi(A \cdot - e) \right\|_{L^p([-E,E]^d)} = \|f - \Phi\|_{L^p(A \cdot [-E,E]^d - e)}. \tag{53}$$

Furthermore, observe that

$$A \cdot [-E, E]^d - e \subseteq [-(dE\|A\|_\infty + \|e\|_\infty), (dE\|A\|_\infty + \|e\|_\infty)]^d = [-F, F]^d. \tag{54}$$

Next, we consider the affine transformations $W_{A,e}(x) := Ax - e$, $W'_A(x) := |\det(A)|^{\frac{1}{p}} x$ as depth-1 networks and take $\Psi_{A,e,E,\eta} := W'_A \circ \Phi_{F,\eta} \circ W_{A,e}$ according to Lemma II.3. Combining (53) and (54) yields

$$\left\| |\det(A)|^{\frac{1}{p}} f(A \cdot - e) - \Psi_{A,e,E,\eta} \right\|_{L^p([-E,E]^d)} = \|f - \Phi_{F,\eta}\|_{L^p(A \cdot [-E,E]^d - e)} \leq \|f - \Phi_{F,\eta}\|_{L^p([-F,F]^d)} \leq \eta.$$

The desired bounds on $\mathcal{M}(\Psi_{A,e,E,\eta})$ and $\mathcal{B}(\Psi_{A,e,E,\eta})$ follow directly by construction. □

C. Canonically Ordered Affine Dictionaries are Effectively Representable

The next result establishes that canonically ordered affine dictionaries with generator functions that can be approximated well by neural networks are effectively representable by neural networks.

Theorem VIII.3. *Let $d, S \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$ be bounded with nonempty interior, $(g_s)_{s=1}^S \in L^\infty(\mathbb{R}^d)$ compactly supported, and $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$ a nondegenerate canonically ordered affine dictionary with generator functions $(g_s)_{s=1}^S$. Assume that there exists a polynomial π such that, for all $s \in \{1,\dots,S\}$, $\varepsilon \in (0,1/2)$, there is a network $\Phi_{s,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying*

$$\|g_s - \Phi_{s,\varepsilon}\|_{L^2(\mathbb{R}^d)} \le \varepsilon, \tag{55}$$

with $\mathcal{M}(\Phi_{s,\varepsilon}) \le \pi(\log(\varepsilon^{-1}))$ and $\mathcal{B}(\Phi_{s,\varepsilon}) \le \pi(\varepsilon^{-1})$. Then, $\mathcal{D}$ is effectively representable by neural networks.

Proof. By Definition VII.1 we need to establish the existence of a bivariate polynomial π such that for each $i \in \mathbb{N}$, $\eta \in (0,1/2)$, there is a network $\Phi_{i,\eta} \in \mathcal{N}_{d,1}$ satisfying

$$\|\varphi_i - \Phi_{i,\eta}\|_{L^2(\Omega)} \le \eta, \tag{56}$$

with $\mathcal{M}(\Phi_{i,\eta}) \le \pi(\log(\eta^{-1}), \log(i))$ and $\mathcal{B}(\Phi_{i,\eta}) \le \pi(\eta^{-1}, i)$. Note that we have

$$\varphi_i = g_{s_i}^{j_i,e_i} = \Big(|\det(A_{s_i,j_i})|^{\frac{1}{2}} g_{s_i}(A_{s_i,j_i}\,\cdot - \delta e_i)\Big)\big|_\Omega,$$

for $s_i \in \{1,\dots,S\}$, $j_i \in J_{s_i}$, and $e_i \in \mathbb{Z}^d$. In order to devise networks satisfying (56), we employ Proposition VIII.2, upon noting that, by virtue of (55), the networks $\Phi_{s,\varepsilon}$ satisfy (52) with $p = 2$, $f = g_s$, for every $D \in \mathbb{R}_+$. Consequently Proposition VIII.2 yields a connectivity bound that is even slightly stronger than needed, as it is independent of i. It remains to ensure that the desired bound on $\mathcal{B}(\Phi_{i,\eta})$ holds. This is the case for $\|A_{s_i,j_i}\|_\infty$ and $\|e_i\|_\infty$ both bounded polynomially in i. In order to verify this, we first bound $\|e_i\|_\infty$ relative to $\|A_{s_i,j_i}\|_\infty$. As the generators $(g_s)_{s=1}^S$ are compactly supported by assumption, there exists $E \in \mathbb{R}_+$ such that, for every $s \in \{1,\dots,S\}$, the support of g_s is contained in $[-E,E]^d$. We thus get, for all $s \in \{1,\dots,S\}$, $j \in J_s$, and $e \in \mathbb{Z}^d$, that

$$\|\delta e\|_\infty \ge \sup_{x\in\Omega} \|A_{s,j}x\|_\infty + E \implies g_s^{j,e}(x) = 0,\ \forall x \in \Omega \implies g_s^{j,e} \notin \mathcal{D}_j.$$

Since Ω is bounded by assumption, there hence exists a constant $c = c(\Omega, (g_s)_{s=1}^S, \delta, d)$ such that, for all $s \in \{1,\dots,S\}$, $j \in J_s$, and $e \in \mathbb{Z}^d$, we have

$$g_s^{j,e} \in \mathcal{D}_j \implies \|e\|_\infty \le c\|A_{s,j}\|_\infty.$$

It remains to show that $\|A_{s_i,j_i}\|_\infty$ is polynomially bounded in i. We start by claiming that, for every $s \in \{1,\dots,S\}$, there is a constant $c_s := c_s(\Omega,\delta,d) > 0$ such that

$$|\det(A_{s,j})| \le c_s|\mathcal{D}_{s,j}|, \text{ for all } j \in J_s. \tag{57}$$

To verify this claim, first note that $|\mathcal{D}_{s,j}| \ge 1$, for all $s \in \{1,\dots,S\}, j \in J_s$, owing to the nondegeneracy condition. Thus, for every $s \in \{1,\dots,S\}$, $j \in J_s$, there exist $x_0 \in \Omega$ and $e_0 \in \mathbb{Z}^d$ such that $g_s^{j,e_0}(x_0) \ne 0$, which implies

$$g_s^{j,e}(x_0 + A_{s,j}^{-1}\delta(e - e_0)) = |\det(A_{s,j})|^{\frac{1}{2}} g_s(A_{s,j}x_0 - \delta e_0) = g_s^{j,e_0}(x_0) \ne 0.$$

We can therefore conclude that $x_0 + A_{s,j}^{-1}\delta(e - e_0) \in \Omega$ implies $g_s^{j,e} \in \mathcal{D}_{s,j}$. Consequently, we have

$$|\mathcal{D}_{s,j}| \geq |\{e \in \mathbb{Z}^d \colon x_0 + A_{s,j}^{-1}\delta(e - e_0) \in \Omega\}| = |\{e \in \mathbb{Z}^d \colon A_{s,j}^{-1}\delta e \in \Omega - x_0\}| = |\mathbb{Z}^d \cap \tfrac{1}{\delta}A_{s,j}(\Omega - x_0)|.$$

As Ω was assumed to have nonempty interior, there exists a constant $C = C(\Omega)$ such that

$$|\mathbb{Z}^d \cap \tfrac{1}{\delta}A_{s,j}(\Omega - x_0)| \geq C\, vol\left(\tfrac{1}{\delta}A_{s,j}(\Omega - x_0)\right) = C\,\delta^{-d}|\det(A_{s,j})|\, vol(\Omega).$$

We have hence established the claim (57). Combining (51) and (57), we obtain, for all $s_i \in \{1, \dots, S\}$, $j \in J_s \backslash \{1\}$,

$$c\|A_{s_i,j_i}\|_\infty^a \leq \sum_{k=1}^{j_i-1} |\det(A_{s_i,k})| \leq c_{s_i} \sum_{k=1}^{j_i-1} |\mathcal{D}_{k,s_i}| \leq c_s i,$$

where the last inequality follows from the fact that $\varphi_i \in \mathcal{D}_{j_i,s_i}$ and hence its index i must be larger than the number of elements contained in preceding sub-dictionaries. This ensures that

$$\|A_{s_i,j_i}\|_\infty \leq \left(\frac{1}{c} \max_{s=1,\dots,S} c_s\right)^{\frac{1}{a}} i^{\frac{1}{a}} + \max_{s=1,\dots,S} \|A_{s,1}\|_\infty, \quad \text{for all } i \in \mathbb{N},$$

thereby completing the proof. □

Remark VIII.4. *Theorem VIII.3 is restricted, for ease of exposition, to bounded Ω and compactly supported generator functions g_s. The result can be extended to $\Omega = \mathbb{R}^d$ and to generator functions g_s of unbounded support but sufficiently fast decay. This extension requires additional technical steps and an alternative definition of canonical ordering. For conciseness we do not provide the details here, but instead refer to the proofs of Theorems IX.3 and IX.5, which deal with the corresponding technical aspects in the context of approximation of Gabor dictionaries by neural networks.*

We can now put the results together to conclude a remarkable universality and optimality property of neural networks: Consider an affine dictionary generated by functions g_s that can be approximated well by neural networks. If this dictionary provides Kolmogorov-Donoho-optimal approximation for a given function class, then so do neural networks.

Theorem VIII.5. *Let $d, S \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$ be bounded with nonempty interior, $(g_s)_{s=1}^S \in L^\infty(\mathbb{R}^d)$ compactly supported, and $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$ a nondegenerate canonically ordered affine dictionary with generator functions $(g_s)_{s=1}^S$. Assume that there exists a polynomial π such that, for all $s \in \{1, \dots, S\}$, $\varepsilon \in (0, 1/2)$, there is a network $\Phi_{s,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying $\|g_s - \Phi_{s,\varepsilon}\|_{L^2(\mathbb{R}^d)} \leq \varepsilon$ with $\mathcal{M}(\Phi_{s,\varepsilon}) \leq \pi(\log(\varepsilon^{-1}))$ and $\mathcal{B}(\Phi_{s,\varepsilon}) \leq \pi(\varepsilon^{-1})$. Then, we have*

$$\gamma_{\mathcal{N}}^{*,\mathit{eff}}(\mathcal{C}) \geq \gamma^{*,\mathit{eff}}(\mathcal{C}, \mathcal{D})$$

for all function classes $\mathcal{C} \subseteq L^2(\Omega)$. In particular, if $\mathcal{C}$ is optimally representable by $\mathcal{D}$ (in the sense of Definition V.4), then $\mathcal{C}$ is optimally representable by neural networks (in the sense of Definition VI.5).

Proof. The first statement follows from Theorem VII.2 and Theorem VIII.3, the second from Theorem VI.4. □

D. Spline wavelets

We next particularize the results developed above to show that neural networks Kolmogorov-Donoho optimally represent all function classes $\mathcal{C}$ that are optimally representable by spline wavelet dictionaries. As spline wavelet dictionaries have B-splines as generator functions, we start by showing how B-splines can be realized through neural networks. For simplicity of exposition, we restrict ourselves to the univariate case throughout.

Definition VIII.6. *Let $N_1 := \chi_{[0,1]}$ and for $m \in \mathbb{N}$, define*

$$N_{m+1} := N_1 * N_m,$$

where $*$ stands for convolution. We refer to N_m as the univariate cardinal B-spline of order m.

Recognizing that B-splines are piecewise polynomial, we can build on Proposition III.5 to get the following statement on the approximation of B-splines by deep neural networks.

Lemma VIII.7. *Let $m \in \mathbb{N}$. There exists a constant $C > 0$ such that for all $\varepsilon \in (0, 1/2)$, there is a neural network $\Phi_\varepsilon \in \mathcal{N}_{1,1}$ satisfying*

$$\|\Phi_\varepsilon - N_m\|_{L^\infty(\mathbb{R})} \leq \varepsilon,$$

with $\mathcal{M}(\Phi_\varepsilon) \leq C \log(\varepsilon^{-1})$ and $\mathcal{B}(\Phi_\varepsilon) \leq 1$.

Proof. The proof is based on the following representation [81, Eq. 19]

$$N_m(x) = \frac{1}{m!} \sum_{k=0}^{m+1} (-1)^k \binom{m+1}{k} \rho((x-k)^m). \tag{58}$$

While N_m is supported on $[0, m]$, the networks Φ_ε can have support outside $[0, m]$ as well. We only need to ensure that Φ_ε is "close" to N_m on $[0, m]$ and at the same time "small" outside the interval $[0, m]$. To accomplish this, we first approximate N_m on the slightly larger domain $[-1, m+1]$ by a linear combination of networks realizing shifted monomials according to (58), and then multiply the resulting network by another one that takes on the value 1 on $[0, m]$ and 0 outside of $[-1, m+1]$. Specifically, we proceed as follows. Proposition III.5 ensures the existence of a constant C_1 such that for all $\varepsilon \in (0, 1/2)$, there is a network $\Psi_{m+2,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying

$$\|\Psi_{m+2,\varepsilon}(x) - x^m\|_{L^\infty([-(m+2),m+2])} \leq \tfrac{\varepsilon}{4(m+2)},$$

with $\mathcal{M}(\Psi_{m+2,\varepsilon}) \leq C_1 \log(\varepsilon^{-1})$ and $\mathcal{B}(\Psi_{m+2,\varepsilon}) \leq 1$. Note that we did not make the dependence of $\mathcal{M}(\Psi_{m+2,\varepsilon})$ on m explicit as we consider m to be fixed. Next, let $T_k(x) := x - k$ and observe that $\rho((x-k)^m)$ can be realized as a neural network according to $\rho \circ \Psi_{m+2,\varepsilon} \circ T_k$, where T_k is taken pursuant to Corollary A.2. Next, we define, for $\varepsilon \in (0, 1/2)$, the network

$$\widetilde{\Phi}_\varepsilon := \frac{1}{m!} \sum_{k=0}^{m+1} (-1)^k \binom{m+1}{k} \rho \circ \Psi_{m+2,\varepsilon} \circ T_k$$

and note that

$$\frac{1}{m!}\binom{m+1}{k} = \frac{m+1}{k!(m-k+1)!} \leq 2,$$

for $k = 0, \dots, m+1$. As ρ is 1-Lipschitz, we have, for all $\varepsilon \in (0, 1/2)$,

$$\begin{aligned} \|\widetilde{\Phi}_\varepsilon - N_m\|_{L^\infty([-1,m+1])} &\leq \sum_{k=0}^{m+1} \frac{1}{m!}\binom{m+1}{k} \|\rho \circ \Psi_{m+2,\varepsilon} \circ T_k - \rho \circ T_k^m\|_{L^\infty([-1,m+1])} \\ &\leq 2\sum_{k=0}^{m+1} \|\Psi_{m+2,\varepsilon}(x) - x^m\|_{L^\infty([-(m+2),m+2])} \leq \tfrac{\varepsilon}{2}. \end{aligned} \tag{59}$$

Let now $\Gamma(x) := \rho(x+1) - \rho(x) - \rho(x-m) + \rho(x-(m+1))$, note that $0 \leq \Gamma(x) \leq 1$, and take $\Phi^{\text{mult}}_{1+\varepsilon/2,\varepsilon/2}$ to be the multiplication network from Lemma III.3. We define $\Phi_\varepsilon := \Phi^{\text{mult}}_{1+\varepsilon/2,\varepsilon/2} \circ (\widetilde{\Phi}_\varepsilon, \Gamma)$ according to Lemma II.3 and Lemma A.7 and note that

$$\|\Phi_\varepsilon - N_m\|_{L^\infty(\mathbb{R})} \leq \|\Phi^{\text{mult}}_{1+\varepsilon/2,\varepsilon/2} \circ (\widetilde{\Phi}_\varepsilon, \Gamma) - \widetilde{\Phi}_\varepsilon \cdot \Gamma\|_{L^\infty([-1,m+1])} + \|\widetilde{\Phi}_\varepsilon \cdot \Gamma - N_m\|_{L^\infty([-1,m+1])} \tag{60}$$

as both N_m and Γ vanish outside $[-1, m+1]$ and $\Phi^{\text{mult}}_{1+\varepsilon/2,\varepsilon/2}$ delivers zero whenever at least one of its inputs is zero. Note that the first term on the right-hand-side of (60) is upper-bounded by $\frac{\varepsilon}{2}$ as a consequence of $N_m(x) \leq 1$ and hence $\widetilde{\Phi}_\varepsilon(x) \leq 1 + \frac{\varepsilon}{2}$, for $x \in [-1, m+1]$, owing to (59). For the second term, we split up the interval $[-1, m+1]$ and first note that, for $x \in [0, m]$, $\Gamma(x) = 1$, which implies $\|\widetilde{\Phi}_\varepsilon \cdot \Gamma - N_m\|_{L^\infty([0,m])} = \|\widetilde{\Phi}_\varepsilon - N_m\|_{L^\infty([0,m])} \leq \varepsilon/2$, again owing to (59). For $x \in [-1, m+1] \setminus [0, m]$, we have $N_m(x) = 0$ and $\Gamma(x) \leq 1$, which yields

$$|\widetilde{\Phi}_\varepsilon(x) \cdot \Gamma(x) - N_m(x)| \leq |\widetilde{\Phi}_\varepsilon(x)| \leq |\widetilde{\Phi}_\varepsilon(x) - N_m(x)| + |N_m(x)| = |\widetilde{\Phi}_\varepsilon(x) - N_m(x)| \leq \varepsilon/2,$$

again by (59). In summary, (59) hence ensures that the second term in (60) is also upper-bounded by $\frac{\varepsilon}{2}$ and therefore $\|\Phi_\varepsilon - N_m\|_{L^\infty(\mathbb{R})} \leq \varepsilon$. Combining Lemma II.3, Proposition III.3, Corollary A.2, Lemma A.4, and Lemma A.7 establishes the desired bounds on $\mathcal{M}(\Phi_{D,\varepsilon})$ and $\mathcal{B}(\Phi_{D,\varepsilon})$. □

Remark VIII.8. *As both N_m and the approximating networks Φ_ε we constructed in the proof of Lemma VIII.7 are supported in $[-1, m+1]$, we have $\|\Phi_\varepsilon - N_m\|_{L^2(\mathbb{R})} \leq (m+2)^{1/2}\|\Phi_\varepsilon - N_m\|_{L^\infty(\mathbb{R})}$, which shows that Lemma VIII.7 continues to hold when the approximation error is measured in $L^2(\mathbb{R})$-norm, albeit with a different constant C.*

We are now ready to introduce spline wavelet dictionaries. For $n, j \in \mathbb{Z}$, set

$$V_n := \operatorname{clos}_{L^2}\Big(\operatorname{span}\{N_m(2^n x - k) : k \in \mathbb{Z}\}\Big),$$

where $\operatorname{clos}_{L^2}$ denotes closure with respect to L^2-norm. Spline spaces V_n, $n \in \mathbb{Z}$, constitute a multiresolution analysis [82] of $L^2(\mathbb{R})$ according to

$$\{0\} \subseteq \dots V_{-1} \subseteq V_0 \subseteq V_1 \subseteq \dots \subseteq L^2(\mathbb{R}).$$

Moreover, with the orthogonal complements $(\dots, W_{-1}, W_0, W_1, \dots)$ such that $V_{n+1} = V_n \oplus W_n$, where $\oplus$ denotes the orthogonal sum, we have

$$L^2(\mathbb{R}) = V_0 \oplus \bigoplus_{k=0}^{\infty} W_k.$$

Theorem VIII.9 ([83, Theorem 1])**.** *Let $m \in \mathbb{N}$. The m-th order spline*

$$\psi_m(x) = \frac{1}{2^{m-1}} \sum_{j=0}^{2m-2} (-1)^j N_{2m}(j+1) \frac{d^m}{dx^m} N_{2m}(2x-j), \tag{61}$$

with support $[0, 2m-1]$, is a basic wavelet that generates W_0 and thereby all the spaces W_n, $n \in \mathbb{Z}$. Consequently, the set

$$\mathcal{W}_m := \{\psi_{k,n}(x) = 2^{n/2}\psi_m(2^n x - k) : n \in \mathbb{N}_0, k \in \mathbb{Z}\} \cup \{\phi_k(x) = N_m(x-k) : k \in \mathbb{Z}\} \tag{62}$$

is a countable complete orthonormal wavelet basis in $L^2(\mathbb{R})$.

Taking $\Omega \subseteq \mathbb{R}$, $S = 2$, $J_1 = \mathbb{N}$, $J_2 = \{1\}$, $A_{1,j} = 2^{j-1}$ for $j \in \mathbb{N}$, and $A_{2,1} = 1$, we get that

$$\mathcal{D} := \left\{ g_s^{j,e}(x) := \left(|A_j|^{\frac{1}{2}} g_s(A_j \, \cdot \, - \delta e) \right)\Big|_{\Omega} : s \in \{1,2\},\ e \in \mathbb{Z},\ j \in J_s, \text{ and } g_s^{j,e} \neq 0 \right\} = \mathcal{W}_m \tag{63}$$

is a nondegenerate canonically ordered affine dictionary with generators $g_1 = \psi_m$ and $g_2 = N_m$. The canonical ordering condition (51) is satisfied with $a = 1$ and $c = 1/2$. Nondegeneracy follows upon noting that $\operatorname{supp}(\psi_{k,n}) = [2^{-n}k, 2^{-n}(2m-1+k)]$ and $\operatorname{supp}(N_m(\,\cdot\, - k)) = [k, m+k]$, which implies that all sub-dictionaries contain at least one element as required.

We have therefore established the following.

Theorem VIII.10. *Let $\Omega \subseteq \mathbb{R}$ be bounded and of nonempty interior and $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subseteq L^2(\Omega)$ a spline wavelet dictionary according to* (63) *ordered per* (50)*. Then, all function classes $\mathcal{C} \subseteq L^2(\Omega)$ that are optimally representable by $\mathcal{D}$ (in the sense of Definition V.4) are optimally representable by neural networks (in the sense of Definition VI.5).*

Proof. As the canonical ordering and the nondegeneracy conditions were already verified, it remains to establish that the generators ψ_m and N_m satisfy the antecedent of Theorem VIII.3. To this end, we first devise an alternative representation of (61). Specifically, using the identity [83, Eq. 2.2]

$$\frac{d^m}{dx^m} N_{2m}(x) = \sum_{j=0}^{m} (-1)^j \binom{m}{j} N_m(x-j),$$

we get

$$\psi_m(x) = \sum_{n=1}^{3m-1} q_n N_m(2x - n + 1), \tag{64}$$

with

$$q_n = \frac{(-1)^{n+1}}{2^{m-1}} \sum_{j=0}^{m} \binom{m}{j} N_{2m}(n-j).$$

As (64) shows that ψ_m is a linear combination of shifts and dilations of N_m, combining Lemma VIII.7 and Remark VIII.8 with Lemma II.6 and Proposition VIII.2 ensures that (55) is satisfied. Application of Theorem VIII.5 then establishes the claim. □

IX. Weyl-Heisenberg dictionaries

In this section, we consider Weyl-Heisenberg a.k.a. Gabor dictionaries [17], which consist of time-frequency translates of a given generator function. Gabor dictionaries play a fundamental role in time-frequency analysis [17] and in the study of partial differential equations [84]. We start with the formal definition of Gabor dictionaries.

Definition IX.1 (Gabor dictionaries)**.** *Let $d \in \mathbb{N}$, $f \in L^2(\mathbb{R}^d)$, and $x, \xi \in \mathbb{R}^d$. We define the translation operator $T_x \colon L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ as*

$$T_x f(t) := f(t-x)$$

and the modulation operator $M_\xi \colon L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d, \mathbb{C})$ as

$$M_\xi f(t) := e^{2\pi i \langle \xi, t\rangle} f(t).$$

Let $\Omega \subseteq \mathbb{R}^d$, $\alpha, \beta > 0$, and $g \in L^2(\mathbb{R}^d)$. The Gabor dictionary $\mathcal{G}(g, \alpha, \beta, \Omega) \subseteq L^2(\Omega)$ is defined as

$$\mathcal{G}(g, \alpha, \beta, \Omega) := \left\{ M_\xi T_x g\big|_\Omega \colon (x, \xi) \in \alpha\mathbb{Z}^d \times \beta\mathbb{Z}^d \right\}.$$

In order to describe representability in neural networks in the sense of Definition VII.1, we need to order the elements in $\mathcal{G}(g, \alpha, \beta, \Omega)$. To this end, let $\mathcal{G}_0(g, \alpha, \beta, \Omega) := \{g\big|_\Omega\}$ and define $\mathcal{G}_n(g, \alpha, \beta, \Omega)$, $n \in \mathbb{N}$, recursively according to

$$\mathcal{G}_n(g, \alpha, \beta, \Omega) := \{M_\xi T_x g\big|_\Omega \colon (x, \xi) \in \alpha\mathbb{Z}^d \times \beta\mathbb{Z}^d, \|x\|_\infty \leq n\alpha, \|\xi\|_\infty \leq n\beta\} \setminus \bigcup_{k=0}^{n-1} \mathcal{G}_k(g, \alpha, \beta, \Omega).$$

We then organize $\mathcal{G}(g, \alpha, \beta, \Omega)$ as

$$\mathcal{G}(g, \alpha, \beta, \Omega) = (\mathcal{G}_0(g, \alpha, \beta, \Omega), \mathcal{G}_1(g, \alpha, \beta, \Omega), \ldots), \tag{65}$$

where the ordering within the sets $\mathcal{G}_n(g, \alpha, \beta, \Omega)$ is arbitrary. We hasten to add that the specifics of the overall ordering in (65) are irrelevant as long as $\mathcal{G}(g, \alpha, \beta, \Omega) = (\varphi_i)_{i \in \mathbb{N}}$ with $\varphi_i = \mathcal{M}_{\xi(i)} T_{x(i)} g\big|_\Omega$ is such that $\|x(i)\|_\infty$ and $\|\xi(i)\|_\infty$ do not grow faster than polynomially in i; this will become apparent in the proof of Theorem IX.3. We note that this ordering is also inspired by that employed in the tail compactness considerations for modulation spaces and Wilson bases as detailed in Appendix C.

As Gabor dictionaries are built from time-shifted and modulated versions of the generator function g, and invariance to time-shifts was already established in Proposition VIII.2, we proceed to showing that the approximation-theoretic properties of the generator function are inherited by its modulated versions. This result can be interpreted as an invariance property to frequency shifts akin to that established in Proposition VIII.2 for affine transformations in the context of affine dictionaries. In summary, neural networks exhibit a remarkable invariance property both to the affine group operations of scaling and translation and to the Weyl-Heisenberg group operations of modulation and translation.

Lemma IX.2. *Let $d \in \mathbb{N}$, $f \in L^2(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d)$, and for every $D \in \mathbb{R}_+$, $\varepsilon \in (0, 1/2)$, let $\Phi_{D,\varepsilon} \in \mathcal{N}_{d,1}$ satisfy*

$$\|f - \Phi_{D,\varepsilon}\|_{L^\infty([-D,D]^d)} \le \varepsilon.$$

Then, there exists a constant $C > 0$ (which does not depend on f) such that for all $D \in \mathbb{R}_+$, $\varepsilon \in (0, 1/2)$, $\xi \in \mathbb{R}^d$, there are networks $\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}, \Phi^{\mathrm{Im}}_{D,\xi,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying

$$\|\mathrm{Re}(M_\xi f) - \Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}\|_{L^\infty([-D,D]^d)} + \|\mathrm{Im}(M_\xi f) - \Phi^{\mathrm{Im}}_{D,\xi,\varepsilon}\|_{L^\infty([-D,D]^d)} \le 3\varepsilon$$

with

$$\mathcal{L}(\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}), \mathcal{L}(\Phi^{\mathrm{Im}}_{D,\xi,\varepsilon}) \le C((\log(\varepsilon^{-1}))^2 + \log(\lceil dD\|\xi\|_\infty\rceil) + (\log(\lceil S_f\rceil))^2) + \mathcal{L}(\Phi_{D,\varepsilon}),$$

$$\mathcal{M}(\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}), \mathcal{M}(\Phi^{\mathrm{Im}}_{D,\xi,\varepsilon}) \le C((\log(\varepsilon^{-1}))^2 + \log(\lceil dD\|\xi\|_\infty\rceil) + (\log(\lceil S_f\rceil))^2 + d) + 4\mathcal{M}(\Phi_{D,\varepsilon}) + 4\mathcal{L}(\Phi_{D,\varepsilon}),$$

and $\mathcal{B}(\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}) \le 1$, where $S_f := \max\{1, \|f\|_{L^\infty(\mathbb{R}^d)}\}$.

Proof. All statements in the proof involving ε pertain to $\varepsilon \in (0, 1/2)$ without explicitly stating this every time. We start by observing that

$$\mathrm{Re}(M_\xi f)(t) = \cos(2\pi\langle\xi, t\rangle) f(t)$$
$$\mathrm{Im}(M_\xi f)(t) = \sin(2\pi\langle\xi, t\rangle) f(t)$$

due to $f \in \mathbb{R}$. Note that for given $\xi \in \mathbb{R}^d$, the map $t \mapsto \langle\xi, t\rangle = \xi^T t = t_1\xi_1 + \cdots + t_d\xi_d$ is simply a linear transformation. Hence, combining Lemma II.3, Theorem III.8, and Corollary A.2 establishes the existence of a constant C_1 such that for all $D \in \mathbb{R}_+$, $\xi \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$, there is a network $\Psi_{D,\xi,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying

$$\sup_{t\in[-D,D]^d} |\cos(2\pi\langle\xi, t\rangle) - \Psi_{D,\xi,\varepsilon}(t)| \le \tfrac{\varepsilon}{6S_f} \tag{66}$$

with

$$\begin{aligned}\mathcal{L}(\Psi_{D,\xi,\varepsilon}) &\le C_1((\log(\varepsilon^{-1}))^2 + (\log(S_f))^2 + \log(\lceil dD\|\xi\|_\infty\rceil)),\\ \mathcal{M}(\Psi_{D,\xi,\varepsilon}) &\le C_1((\log(\varepsilon^{-1}))^2 + (\log(S_f))^2 + \log(\lceil dD\|\xi\|_\infty\rceil) + d),\end{aligned} \tag{67}$$

and $\mathcal{B}(\Psi_{D,\xi,\varepsilon}) \le 1$. Moreover, Proposition III.3 guarantees the existence of a constant $C_2 > 0$ such that for all $\varepsilon \in (0, 1/2)$, there is a network $\mu_\varepsilon \in \mathcal{N}_{2,1}$ satisfying

$$\sup_{x,y\in[-S_f-1/2, S_f+1/2]} |\mu_\varepsilon(x, y) - xy| \le \tfrac{\varepsilon}{6} \tag{68}$$

with

$$\mathcal{L}(\mu_\varepsilon), \mathcal{M}(\mu_\varepsilon) \le C_2(\log(\varepsilon^{-1}) + \log(\lceil S_f\rceil)) \tag{69}$$

and $\mathcal{B}(\mu_\varepsilon) \le 1$. Using Lemmas II.4 and II.5, we get that the network $\Gamma_{D,\xi,\varepsilon} := (\Psi_{D,\xi,\varepsilon}, \Phi_{D,\varepsilon}) \in \mathcal{N}_{d,2}$ satisfies

$$\mathcal{L}(\Gamma_{D,\xi,\varepsilon}) \le \max\{\mathcal{L}(\Psi_{D,\xi,\varepsilon}), \mathcal{L}(\Phi_{D,\varepsilon})\},$$

$$\mathcal{M}(\Gamma_{D,\xi,\varepsilon}) \le 2\,\mathcal{M}(\Psi_{D,\xi,\varepsilon}) + 2\,\mathcal{M}(\Phi_{D,\varepsilon}) + 2\,\mathcal{L}(\Psi_{D,\xi,\varepsilon}) + 2\,\mathcal{L}(\Phi_{D,\varepsilon}),$$

and $\mathcal{B}(\Gamma_{D,\xi,\varepsilon}) \leq 1$. Finally, applying Lemma II.3 to concatenate the networks $\Gamma_{D,\xi,\varepsilon}$ and μ_ε, we obtain the network

$$\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon} := \mu_\varepsilon \circ \Gamma_{D,\xi,\varepsilon} = \mu_\varepsilon \circ (\Psi_{D,\xi,\varepsilon}, \Phi_{D,\varepsilon}) \in \mathcal{N}_{d,1}$$

satisfying

$$\mathcal{L}(\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}) \leq \max\{\mathcal{L}(\Psi_{D,\xi,\varepsilon}), \mathcal{L}(\Phi_{D,\varepsilon})\} + \mathcal{L}(\mu_\varepsilon), \tag{70}$$

$$\mathcal{M}(\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}) \leq 4\mathcal{M}(\Psi_{D,\xi,\varepsilon}) + 4\mathcal{M}(\Phi_{D,\varepsilon}) + 4\mathcal{L}(\Psi_{D,\xi,\varepsilon}) + 4\mathcal{L}(\Phi_{D,\varepsilon}) + 2\mathcal{M}(\mu_\varepsilon), \tag{71}$$

and $\mathcal{B}(\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}) \leq 1$. Next, observe that (66) and (68) imply that

$$\begin{aligned}
\|\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon} - \mathrm{Re}(M_\xi f)\|_{L^\infty([-D,D]^d)} &= \|\mu_\varepsilon(\Psi_{D,\xi,\varepsilon}(\,\cdot\,), \Phi_{D,\varepsilon}(\,\cdot\,)) - \cos(2\pi\langle \xi, \cdot\,\rangle) f(\,\cdot\,)\|_{L^\infty([-D,D]^d)} \\
&\leq \|\mu_\varepsilon(\Psi_{D,\xi,\varepsilon}(\,\cdot\,), \Phi_{D,\varepsilon}(\,\cdot\,)) - \Psi_{D,\xi,\varepsilon}(\,\cdot\,)\Phi_{D,\varepsilon}(\,\cdot\,)\|_{L^\infty([-D,D]^d)} \\
&\quad + \|\Psi_{D,\xi,\varepsilon}(\,\cdot\,)\Phi_{D,\varepsilon}(\,\cdot\,) - \cos(2\pi\langle \xi, \cdot\,\rangle) f(\,\cdot\,)\|_{L^\infty([-D,D]^d)} \\
&\leq \|\mu_\varepsilon(\Psi_{D,\xi,\varepsilon}(\,\cdot\,), \Phi_{D,\varepsilon}(\,\cdot\,)) - \Psi_{D,\xi,\varepsilon}(\,\cdot\,)\Phi_{D,\varepsilon}(\,\cdot\,)\|_{L^\infty([-D,D]^d)} \\
&\quad + \|\Psi_{D,\xi,\varepsilon}(\,\cdot\,)(\Phi_{D,\varepsilon}(\,\cdot\,) - f(\,\cdot\,))\|_{L^\infty([-D,D]^d)} \\
&\quad + \|\Psi_{D,\xi,\varepsilon}(\,\cdot\,) f(\,\cdot\,) - \cos(2\pi\langle \xi, \cdot\,\rangle) f(\,\cdot\,)\|_{L^\infty([-D,D]^d)} \\
&\leq \tfrac{\varepsilon}{6} + (1 + \tfrac{\varepsilon}{6S_f})\varepsilon + \tfrac{\varepsilon}{6} \leq \tfrac{3}{2}\varepsilon.
\end{aligned}$$

Combining (67), (69), (71), and (70) we can further see that there exists a constant $C > 0$ such that

$$\begin{aligned}
\mathcal{L}(\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}) &\leq C((\log(\varepsilon^{-1}))^2 + \log(\lceil dD\|\xi\|_\infty\rceil) + (\log(\lceil S_f\rceil))^2) + \mathcal{L}(\Phi_{D,\varepsilon}), \\
\mathcal{M}(\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon}) &\leq C((\log(\varepsilon^{-1}))^2 + \log(\lceil dD\|\xi\|_\infty\rceil) + (\log(\lceil S_f\rceil))^2 + d) + 4\mathcal{M}(\Phi_{D,\varepsilon}) + 4\mathcal{L}(\Phi_{D,\varepsilon}),
\end{aligned}$$

and $\mathcal{B}(\Phi^{\mathrm{Re}}_{D,\xi,\varepsilon})) \leq 1$. The results for $\Phi^{\mathrm{Im}}_{D,\xi,\varepsilon}$ follow analogously, simply by using $\sin(x) = \cos(x - \pi/2)$. □

Note that Gabor dictionaries necessarily contain complex-valued functions. The theory developed so far was, however, phrased for neural networks with real-valued outputs. As is evident from the proof of Lemma IX.2, this is not problematic when the generator function g is real-valued. For complex-valued generator functions we would need a version of Proposition III.3 that applies to the multiplication of complex numbers. Due to $(a+ib)(a'+ib') = (aa' - bb') + i(ab' + a'b)$ such a network can be constructed by realizing the real and imaginary parts of the product as a sum of real-valued multiplication networks and then proceeding as in the proof above. We omit the details as they are straightforward and would not lead to new conceptual insights. Furthermore, an extension—to the complex-valued case—of the concept of effective representability by neural networks according to Definition VII.1 would be needed. This can be effected by considering the set of neural networks with 1-dimensional complex-valued output as neural networks with 2-dimensional real-valued output, i.e., by setting

$$\mathcal{N}^{\mathbb{C}}_{d,1} := \mathcal{N}_{d,2},$$

with the convention that the first component represents the real part and the second the imaginary part.

We proceed to establish conditions for effective representability of Gabor dictionaries by neural networks.

Theorem IX.3. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, $\alpha, \beta > 0$, $g \in L^2(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d)$, and let $\mathcal{G}(g, \alpha, \beta, \Omega)$ be the corresponding Gabor dictionary with ordering as defined in* (65)*. Assume that Ω is bounded or that $\Omega = \mathbb{R}^d$ and g is compactly supported. Further, suppose that there exists a polynomial π such that for every $x \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$, there is a network $\Phi_{x,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying*

$$\|g - \Phi_{x,\varepsilon}\|_{L^\infty(x+\Omega)} \leq \varepsilon, \tag{72}$$

with $\mathcal{M}(\Phi_{x,\varepsilon}) \leq \pi(\log(\varepsilon^{-1}), \log(\|x\|_\infty))$, $\mathcal{B}(\Phi_{x,\varepsilon}) \leq \pi(\varepsilon^{-1}, \|x\|_\infty)$. Then, $\mathcal{G}(g, \alpha, \beta, \Omega)$ is effectively representable by neural networks.

Proof. We start by noting that owing to (65), we have $\mathcal{G}(g, \alpha, \beta, \Omega) = (\varphi_i)_{i \in \mathbb{N}}$ with $\varphi_i = \mathcal{M}_{\xi(i)} T_{x(i)} g \in \mathcal{G}_{n(i)}(g, \alpha, \beta, \Omega)$, where

$$\|\xi(i)\|_\infty \leq n(i)\beta \leq i\beta \quad \text{and} \quad \|x(i)\|_\infty \leq n(i)\alpha \leq i\alpha. \tag{73}$$

Next, we take the affine transformation $W_x(y) := y - x$ to be a depth-1 network and observe that, due to (72) and Lemma II.3, we have, for all $x \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$,

$$\|T_x g - \Phi_{-x,\varepsilon} \circ W_x\|_{L^\infty(\Omega)} = \|g - \Phi_{-x,\varepsilon}\|_{L^\infty(-x+\Omega)} \leq \varepsilon, \tag{74}$$

with

$$\begin{aligned} \mathcal{M}(\Phi_{-x,\varepsilon} \circ W_x) &\leq 2(\pi(\log(\varepsilon^{-1}), \log(\|x\|_\infty)) + 2d) \\ \mathcal{B}(\mathcal{M}(\Phi_{-x,\varepsilon} \circ W_x)) &\leq \max\{\mathcal{B}(\Phi_{-x,\varepsilon}), \|x\|_\infty\} \leq \pi(\varepsilon^{-1}, \|x\|_\infty) + \|x\|_\infty. \end{aligned}$$

We first consider the case where Ω is bounded and let $E \in \mathbb{R}_+$ be such that $\Omega \subseteq [-E, E]^d$. Combining (74) with Proposition VIII.2 and Lemma IX.2, we can infer the existence of a multivariate polynomial π_1 such that for all $i \in \mathbb{N}$, $\varepsilon \in (0, 1/2)$, there is a network $\Phi_{i,\varepsilon} = (\Phi_{i,\varepsilon}^{\text{Re}}, \Phi_{i,\varepsilon}^{\text{Im}}) \in \mathcal{N}_{d,1}^{\mathbb{C}}$ satisfying

$$\|\text{Re}(\mathcal{M}_{\xi(i)} T_{x(i)} g) - \Phi_{i,\varepsilon}^{\text{Re}}\|_{L^\infty(\Omega)} + \|\text{Im}(\mathcal{M}_{\xi(i)} T_{x(i)} g) - \Phi_{i,\varepsilon}^{\text{Im}}\|_{L^\infty(\Omega)} \leq (2E)^{-\frac{d}{2}}\varepsilon, \tag{75}$$

with

$$\begin{aligned} \mathcal{M}(\Phi_{i,\varepsilon}^{\text{Re}}), \mathcal{M}(\Phi_{i,\varepsilon}^{\text{Im}}) &\leq \pi_1(\log(\varepsilon^{-1}), \log(\|\xi(i)\|_\infty), \log(\|x(i)\|_\infty)), \\ \mathcal{B}(\Phi_{i,\varepsilon}^{\text{Re}}), \mathcal{B}(\Phi_{i,\varepsilon}^{\text{Im}}) &\leq \pi_1(\varepsilon^{-1}, \|\xi(i)\|_\infty, \|x(i)\|_\infty). \end{aligned} \tag{76}$$

Note that here we did not make the dependence of the connectivity and the weight upper bounds on d and E explicit as these quantities are irrelevant for the purposes of what we want to show, as long as they are finite, of course, which is the case by assumption. Likewise, we did not explicitly indicate the dependence of π_1 on g. As $|z| \leq |\mathrm{Re}(z)| + |\mathrm{Im}(z)|$, it follows from (75) that for all $i \in \mathbb{N}$, $\varepsilon \in (0, 1/2)$,

$$\begin{aligned}\|\varphi_i - \Phi_{i,\varepsilon}\|_{L^2(\Omega,\mathbb{C})} &\leq (2E)^{\frac{d}{2}} \|\varphi_i - \Phi_{i,\varepsilon}\|_{L^\infty(\Omega,\mathbb{C})} \\ &\leq (2E)^{\frac{d}{2}} \left(\|\mathrm{Re}(\varphi_i) - \Phi^{\mathrm{Re}}_{i,\varepsilon}\|_{L^\infty(\Omega)} + \|\mathrm{Im}(\varphi_i) - \Phi^{\mathrm{Im}}_{i,\varepsilon}\|_{L^\infty(\Omega)}\right) \leq \varepsilon.\end{aligned}$$

Moreover, (73) and (76) imply the existence of a polynomial π_2 such that

$$\mathcal{M}(\Phi^{\mathrm{Re}}_{i,\varepsilon}), \mathcal{M}(\Phi^{\mathrm{Im}}_{i,\varepsilon}) \leq \pi_2(\log(\varepsilon^{-1}), \log(i)), \quad \mathcal{B}(\Phi^{\mathrm{Re}}_{i,\varepsilon}), \mathcal{B}(\Phi^{\mathrm{Im}}_{i,\varepsilon}) \leq \pi_2(\varepsilon^{-1}, i),$$

for all $i \in \mathbb{N}$, $\varepsilon \in (0, 1/2)$. We can therefore conclude that $\mathcal{G}(g, \alpha, \beta, \Omega)$ is effectively representable by neural networks.

We proceed to proving the statement for the case $\Omega = \mathbb{R}^d$ and g compactly supported, i.e., there exists $E \in \mathbb{R}_+$ such that $\mathrm{supp}(g) \subseteq [-E, E]^d$. This implies

$$\mathrm{supp}(M_\xi T_x g) = \mathrm{supp}(T_x g) \subseteq x + [-E, E]^d \subseteq [-(\|x\|_\infty + E), \|x\|_\infty + E]^d.$$

Again, combining (74) with Proposition VIII.2 and Lemma IX.2 establishes the existence of a polynomial π_3 such that for all $x, \xi \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$, there are networks $\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}, \Psi^{\mathrm{Im}}_{x,\xi,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying

$$\|\mathrm{Re}(M_\xi T_x g) - \Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}\|_{L^\infty(S_x)} + \|\mathrm{Im}(M_\xi T_x g) - \Psi^{\mathrm{Im}}_{x,\xi,\varepsilon}\|_{L^\infty(S_x)} \leq \tfrac{\varepsilon}{2 s_x}, \tag{77}$$

with

$$\begin{aligned}\mathcal{M}(\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}), \mathcal{M}(\Psi^{\mathrm{Im}}_{x,\xi,\varepsilon}) &\leq \pi_3(\log(\varepsilon^{-1}), \log(\|x\|_\infty), \log(\|\xi\|_\infty)), \\ \mathcal{B}(\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}), \mathcal{B}(\Psi^{\mathrm{Im}}_{x,\xi,\varepsilon}) &\leq \pi_3(\varepsilon^{-1}, \|x\|_\infty, \|\xi\|_\infty),\end{aligned}$$

where we set $S_x := [-(\|x\|_\infty + E + 1), \|x\|_\infty + E + 1]^d$ and $s_x := |S_x|^{1/2}$ to simplify notation. As we want to establish effective representability for $\Omega = \mathbb{R}^d$, the estimate in (77) is insufficient. In particular, we have no control over the behavior of the networks $\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}, \Psi^{\mathrm{Im}}_{x,\xi,\varepsilon}$ outside the set S_x. We can, however, construct networks which exhibit the same scaling behavior in terms of $\mathcal{M}$ and $\mathcal{B}$, are supported in S_x, and realize the same output for all inputs in S_x. To this end let, for $y \in \mathbb{R}_+$, the network $\alpha_y \in \mathcal{N}_{1,1}$ be given by

$$\alpha_y(t) := \rho(t - (-y - 1)) - \rho(t - (-y)) - \rho(t - y) + \rho(t - (y + 1)), \quad t \in \mathbb{R}.$$

Note that $\alpha_y(t) = 1$ for $t \in [-y, y]$, $\alpha_y(t) = 0$ for $t \notin [-y - 1, y + 1]$, and $\alpha_y(t) \in (0, 1)$ else. Next, consider, for $x \in \mathbb{R}^d$, the network given by

$$\chi_x(t) := \rho\left(\left[\sum_{i=1}^{d} \alpha_{\|x\|_\infty + E}(t_i)\right] - (d - 1)\right), \quad t = (t_1, t_2, \ldots, t_d) \in \mathbb{R}^d,$$

and note that

$$\begin{aligned}\chi_x(t) &= 1, \quad \forall t \in [-(\|x\|_\infty + E), \|x\|_\infty + E]^d \\ \chi_x(t) &= 0, \quad \forall t \notin [-(\|x\|_\infty + E + 1), \|x\|_\infty + E + 1]^d \\ 0 \leq \chi_x(t) &\leq 1, \quad \forall t \in \mathbb{R}^d.\end{aligned}$$

As d and E are considered fixed here, there exists a constant C_1 such that, for all $x \in \mathbb{R}^d$, we have $\mathcal{M}(\chi_x) \leq C_1$ and $\mathcal{B}(\chi_x) \leq C_1 \max\{1, \|x\|_\infty\}$. Now, let $B := \max\{1, \|g\|_{L^\infty(\mathbb{R})}\}$. Next, by Proposition III.3 there exists a constant C_2 such that, for all $x \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$, there is a network $\mu_{x,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying

$$\sup_{y,z \in [-2B, 2B]} |\mu_{x,\varepsilon}(y, z) - yz| \leq \tfrac{\varepsilon}{4s_x}, \tag{78}$$

and, for all $y \in \mathbb{R}$,

$$\mu_{x,\varepsilon}(0, y) = \mu_{x,\varepsilon}(y, 0) = 0, \tag{79}$$

with $\mathcal{M}(\mu_{x,\varepsilon}) \leq C_2(\log(\varepsilon^{-1}) + \log(s_x))$ and $\mathcal{B}(\mu_{x,\varepsilon}) \leq 1$. Note that in the upper bound on $\mathcal{M}(\mu_{x,\varepsilon})$, we did not make the dependence on B explicit as we consider g fixed for the purposes of the proof. Next, as E is fixed, there exists a constant C_3 such that $\mathcal{M}(\mu_{x,\varepsilon}) \leq C_3(\log(\varepsilon^{-1}) + \log(\|x\|_\infty + 1))$, for all $x \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$.

We now take

$$\Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon} := \mu_{x,\varepsilon} \circ (\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}, \chi_x) \quad \text{and} \quad \Gamma^{\mathrm{Im}}_{x,\xi,\varepsilon} := \mu_{x,\varepsilon} \circ (\Psi^{\mathrm{Im}}_{x,\xi,\varepsilon}, \chi_x)$$

according to Lemmas II.5 and II.3, which ensures the existence of a polynomial π_4 such that, for all $x, \xi \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$,

$$\begin{aligned}\mathcal{M}(\Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon}), \mathcal{M}(\Gamma^{\mathrm{Im}}_{x,\xi,\varepsilon}) &\leq \pi_4(\log(\varepsilon^{-1}), \log(\|x\|_\infty), \log(\|\xi\|_\infty)), \\ \mathcal{B}(\Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon}), \mathcal{B}(\Gamma^{\mathrm{Im}}_{x,\xi,\varepsilon}) &\leq \pi_4(\varepsilon^{-1}, \|x\|_\infty, \|\xi\|_\infty).\end{aligned} \tag{80}$$

Furthermore,

$$\begin{aligned}\|\Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon} - \mathrm{Re}(M_\xi T_x g)\|_{L^\infty(S_x)} &\leq \|\mu_{x,\varepsilon} \circ (\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}, \chi_x) - \Psi^{\mathrm{Re}}_{x,\xi,\varepsilon} \cdot \chi_x\|_{L^\infty(S_x)} \\ &\quad + \|\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon} \cdot \chi_x - \mathrm{Re}(M_\xi T_x g)\|_{L^\infty(S_x)},\end{aligned} \tag{81}$$

where the first term is upper-bounded by $\frac{\varepsilon}{4s_x}$ due to (78). The second term on the right-hand side of (81) is upper-bounded as follows. First, note that for $t \in S_x \setminus [-(\|x\|_\infty + E), \|x\|_\infty + E]^d$, we have $\mathrm{Re}(M_\xi T_x g)(t) = 0$ and $|\chi_x(t)| \leq 1$, which implies

$$\begin{aligned}|\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}(t) \cdot \chi_x(t) - \mathrm{Re}(M_\xi T_x g)(t)| &\leq |\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}(t)| \leq |\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}(t) - \mathrm{Re}(M_\xi T_x g)(t)| + |\mathrm{Re}(M_\xi T_x g)(t)| \\ &= |\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon}(t) - \mathrm{Re}(M_\xi T_x g)(t)|.\end{aligned}$$

As $|\chi_x(t)| = 1$ for $t \in [-(\|x\|_\infty + E), \|x\|_\infty + E]^d$, together with (81), this yields

$$\|\Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon} - \mathrm{Re}(M_\xi T_x g)\|_{L^\infty(S_x)} \leq \tfrac{\varepsilon}{4s_x} + \|\Psi^{\mathrm{Re}}_{x,\xi,\varepsilon} - \mathrm{Re}(M_\xi T_x g)\|_{L^\infty(S_x)}.$$

The analogous estimate for $\|\Gamma^{\mathrm{Im}}_{x,\xi,\varepsilon} - \mathrm{Im}(M_\xi T_x g)\|_{L^\infty(S_x)}$ is obtained in exactly the same manner. Together with (77), we can finally infer that, for all $x, \xi \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$,

$$\|\mathrm{Re}(M_\xi T_x g) - \Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon}\|_{L^\infty(S_x)} + \|\mathrm{Im}(M_\xi T_x g) - \Gamma^{\mathrm{Im}}_{x,\xi,\varepsilon}\|_{L^\infty(S_x)} \leq \tfrac{\varepsilon}{s_x}.$$

As $M_\xi T_x g$, $\Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon}$, and $\Gamma^{\mathrm{Im}}_{x,\xi,\varepsilon}$ are supported in S_x for all $x, \xi \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$, using (79), we get

$$\begin{aligned}
&\|\mathrm{Re}(M_\xi T_x g) - \Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon}\|_{L^2(\mathbb{R}^d)} + \|\mathrm{Im}(M_\xi T_x g) - \Gamma^{\mathrm{Im}}_{x,\xi,\varepsilon}\|_{L^2(\mathbb{R}^d)} \\
&= \|\mathrm{Re}(M_\xi T_x g) - \Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon}\|_{L^2(S_x)} + \|\mathrm{Im}(M_\xi T_x g) - \Gamma^{\mathrm{Im}}_{x,\xi,\varepsilon}\|_{L^2(S_x)} \\
&\leq s_x \|\mathrm{Re}(M_\xi T_x g) - \Gamma^{\mathrm{Re}}_{x,\xi,\varepsilon}\|_{L^\infty(S_x)} + s_x \|\mathrm{Im}(M_\xi T_x g) - \Gamma^{\mathrm{Im}}_{x,\xi,\varepsilon}\|_{L^\infty(S_x)} \leq \varepsilon.
\end{aligned} \tag{82}$$

Consider now, for $i \in \mathbb{N}$, $\varepsilon \in (0, 1/2)$, the complex-valued network $\Gamma_{i,\varepsilon} \in \mathcal{N}^{\mathbb{C}}_{d,1}$ given by

$$\Gamma_{i,\varepsilon} := (\Gamma^{\mathrm{Re}}_{x(i),\xi(i),\varepsilon}, \Gamma^{\mathrm{Im}}_{x(i),\xi(i),\varepsilon})$$

and note that, for $f \in L^2(\Omega, \mathbb{C})$,

$$\begin{aligned}
\|f\|_{L^2(\Omega,\mathbb{C})} &= \left(\int_\Omega |f(t)|^2 \mathrm{d}t\right)^{\frac{1}{2}} = \left(\int_\Omega |\mathrm{Re}(f(t))|^2 + |\mathrm{Im}(f(t))|^2 \mathrm{d}t\right)^{\frac{1}{2}} = \left(\|\mathrm{Re}(f)\|^2_{L^2(\Omega)} + \|\mathrm{Im}(f)\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}} \\
&\leq \|\mathrm{Re}(f)\|_{L^2(\Omega)} + \|\mathrm{Im}(f)\|_{L^2(\Omega)}.
\end{aligned}$$

Hence, (82) implies that, for all $i \in \mathbb{N}$, $\varepsilon \in (0, 1/2)$,

$$\|\varphi_i - \Gamma_{i,\varepsilon}\|_{L^2(\mathbb{R}^d,\mathbb{C})} = \|M_{\xi(i)} T_{x(i)} g - (\Gamma^{\mathrm{Re}}_{x(i),\xi(i),\varepsilon}, \Gamma^{\mathrm{Im}}_{x(i),\xi(i),\varepsilon})\|_{L^2(\mathbb{R}^d,\mathbb{C})} \leq \varepsilon.$$

Finally, using (73) in (80), it follows that there exists a polynomial π_5 such that for all $i \in \mathbb{N}$, $\varepsilon \in (0, 1/2)$, we have $\mathcal{M}(\Gamma^{\mathrm{Re}}_{x(i),\xi(i),\varepsilon}), \mathcal{M}(\Gamma^{\mathrm{Im}}_{x(i),\xi(i),\varepsilon}) \leq \pi_5(\log(\varepsilon^{-1}), \log(i))$ and $\mathcal{B}(\Gamma^{\mathrm{Re}}_{x(i),\xi(i),\varepsilon}), \mathcal{B}(\Gamma^{\mathrm{Im}}_{x(i),\xi(i),\varepsilon}) \leq \pi_5(\varepsilon^{-1}, i)$, which finalizes the proof. □

Next, we establish the central result of this section. To this end, we first recall that according to Theorem VIII.5 neural networks provide optimal approximations for all function classes that are optimally approximated by affine dictionaries (generated by functions f that can be approximated well by neural networks). While this universality property is significant as it applies to all affine dictionaries, it is perhaps not completely surprising as affine dictionaries are generated by affine transformations and neural networks consist of concatenations of affine transformations and nonlinearities. Gabor dictionaries, on the other hand, exhibit a fundamentally different mathematical structure. The next result shows that neural networks also provide optimal approximations for all function classes that are optimally approximated by Gabor dictionaries (again, with generator functions that can be approximated well by neural networks).

Theorem IX.4. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, $\alpha, \beta > 0$, $g \in L^2(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d)$, and let $\mathcal{G}(g, \alpha, \beta, \Omega)$ be the corresponding Gabor dictionary with ordering as defined in* (65)*. Assume that Ω is bounded or that $\Omega = \mathbb{R}^d$ and g is compactly supported. Further, suppose that there exists a polynomial π such that for every $x \in \mathbb{R}^d$, $\varepsilon \in (0, 1/2)$, there is a network $\Phi_{x,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying*

$$\|g - \Phi_{x,\varepsilon}\|_{L^\infty(x+\Omega)} \leq \varepsilon,$$

with $\mathcal{M}(\Phi_{x,\varepsilon}) \leq \pi(\log(\varepsilon^{-1}), \log(\|x\|_\infty))$, $\mathcal{B}(\Phi_{x,\varepsilon}) \leq \pi(\varepsilon^{-1}, \|x\|_\infty)$. Then, for all function classes $\mathcal{C} \subseteq L^2(\Omega)$, we have

$$\gamma_{\mathcal{N}}^{*,\mathit{eff}}(\mathcal{C}) \geq \gamma^{*,\mathit{eff}}(\mathcal{C}, \mathcal{G}(g, \alpha, \beta, \Omega)).$$

In particular, if $\mathcal{C}$ is optimally representable by $\mathcal{G}(g, \alpha, \beta, \Omega)$ (in the sense of Definition V.4), then $\mathcal{C}$ is optimally representable by neural networks (in the sense of Definition VI.5).

Proof. The first statement follows from Theorem VII.2 and Theorem IX.3, the second is by Theorem VI.4. □

We complete the program in this section by showing that the Gaussian function satisfies the conditions on the generator g in Theorem IX.3 for bounded Ω. Gaussian functions are widely used generator functions for Gabor dictionaries owing to their excellent time-frequency localization and their frame-theoretic optimality properties [17]. We hasten to add that the result below can be extended to any generator function g of sufficiently fast decay and sufficient smoothness.

Lemma IX.5. *For $d \in \mathbb{N}$, let $g_d \in L^2(\mathbb{R}^d)$ be given by*

$$g_d(x) := e^{-\|x\|_2^2}.$$

There exists a constant $C > 0$ such that, for all $d \in \mathbb{N}$ and $\varepsilon \in (0, 1/2)$, there is a network $\Phi_{d,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying

$$\|\Phi_{d,\varepsilon} - g\|_{L^\infty(\mathbb{R}^d)} \leq \varepsilon,$$

with $\mathcal{M}(\Phi_{d,\varepsilon}) \leq Cd(\log(\varepsilon^{-1}))^2((\log(\varepsilon^{-1}))^2 + \log(d))$, $\mathcal{B}(\Phi_{d,\varepsilon}) \leq 1$.

Proof. Observe that g_d can be written as the composition $h \circ f_d$ of the functions $f_d \colon \mathbb{R}^d \to \mathbb{R}_+$ and $h \colon \mathbb{R}_+ \to \mathbb{R}$ given by

$$f_d(x) := \|x\|_2^2 = \sum_{i=1}^d x_i^2 \quad \text{and} \quad h(y) := e^{-y}.$$

By Proposition III.3 and Lemma II.6, there exists a constant $C_1 > 0$ such that, for every $d \in \mathbb{N}$, $D \in [1, \infty)$, $\varepsilon \in (0, 1/2)$, there is a network $\Psi_{d,D,\varepsilon} \in \mathcal{N}_{d,1}$ satisfying

$$\sup_{x \in [-D,D]^d} |\Psi_{d,D,\varepsilon}(x) - \|x\|_2^2| \leq \tfrac{\varepsilon}{2}, \tag{83}$$

$$\mathcal{M}(\Psi_{d,D,\varepsilon}) \leq C_1 d(\log(\varepsilon^{-1}) + \log(\lceil D \rceil)), \quad \mathcal{B}(\Psi_{d,D,\varepsilon}) \leq 1. \tag{84}$$

Moreover, as $|\frac{d^n}{dy^n}e^{-y}| = |e^{-y}| \leq 1$ for all $n \in \mathbb{N}$, $y \geq 0$, Lemma A.6 implies the existence of a constant $C_2 > 0$ such that for every $d \in \mathbb{N}$, $D \in [1, \infty)$, $\varepsilon \in (0, 1/2)$, there is a network $\Gamma_{d,D,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying

$$\sup_{y \in [0, dD^2]} |\Gamma_{d,D,\varepsilon}(y) - e^{-y}| \leq \tfrac{\varepsilon}{2}, \tag{85}$$

$$\mathcal{M}(\Gamma_{d,D,\varepsilon}) \leq C_2 dD^2((\log(\varepsilon^{-1}))^2 + \log(d) + \log(\lceil D \rceil)), \quad \mathcal{B}(\Gamma_{D,\varepsilon}) \leq 1. \tag{86}$$

Now, let $D_\varepsilon := \log(\varepsilon^{-1})$ and take $\widetilde{\Phi}_{d,\varepsilon} := \Gamma_{d,D_\varepsilon,\varepsilon} \circ \Psi_{d,D_\varepsilon,\varepsilon}$ according to Lemma II.3. Consequently, it follows from (84) and (86) that there exists a constant $C_2 > 0$ such that for all $d \in \mathbb{N}$, $\varepsilon \in (0, 1/2)$, we have $\mathcal{M}(\widetilde{\Phi}_{d,\varepsilon}) \leq C_2 d(\log(\varepsilon^{-1}))^2((\log(\varepsilon^{-1}))^2 + \log(d))$ and $\mathcal{B}(\widetilde{\Phi}_{d,\varepsilon}) \leq 1$. Moreover, as $|e^{-y}| \leq 1$ for all $y \geq 0$, combining (83) and (85) yields for all $\varepsilon \in (0, 1/2)$, $x \in [-D_\varepsilon, D_\varepsilon]^d$,

$$\begin{aligned}
|g(x) - \widetilde{\Phi}_{d,\varepsilon}(x)| &= |e^{-\|x\|_2^2} - \Gamma_{d,D_\varepsilon,\varepsilon}(\Psi_{d,D_\varepsilon,\varepsilon}(x))| \\
&\leq |e^{-\|x\|_2^2} - e^{-\Psi_{d,D_\varepsilon,\varepsilon}(x)}| + |e^{-\Psi_{d,D_\varepsilon,\varepsilon}(x)} - \Gamma_{d,D_\varepsilon,\varepsilon}(\Psi_{d,D_\varepsilon,\varepsilon}(x))| \\
&\leq \tfrac{\varepsilon}{2} + \tfrac{\varepsilon}{2} = \varepsilon.
\end{aligned}$$

We can now use the same approach as in the proof of Theorem IX.3 to construct networks $\Phi_{d,\varepsilon}$ supported on the interval $[-D_\varepsilon, D_\varepsilon]^d$ over which they approximate g to within error ε, and obey $\mathcal{M}(\Phi_\varepsilon) \leq Cd(\log(\varepsilon^{-1}))^2((\log(\varepsilon^{-1}))^2 + \log(d))$, $\mathcal{B}(\Phi_{d,\varepsilon}) \leq 1$ for some absolute constant C. Together with $|g(x)| \leq \varepsilon$, for all $x \in \mathbb{R}^d \backslash [-D_\varepsilon, D_\varepsilon]^d$, this completes the proof. □

Remark IX.6. *Note that Lemma IX.5 establishes an approximation result that is even stronger than what is required by Theorem IX.3. Specifically, we achieve ε-approximation over all of $\mathbb{R}^d$ with a network that does not depend on the shift parameter x, while exhibiting the desired growth rates on $\mathcal{M}$ and $\mathcal{B}$, which consequently do not depend on the shift parameter as well. The idea underlying this construction can be used to strengthen Theorem IX.3 to apply to $\Omega = \mathbb{R}^d$ and generator functions of unbounded support, but sufficiently rapid decay.*

We conclude this section with a remark on the neural network approximation of the real-valued counterpart of Gabor dictionaries known as Wilson dictionaries [74], [17] and consisting of cosine-modulated and time-shifted versions of a given generator function, see also Appendix C. The techniques developed in this section, mutatis mutandis, show that neural networks provide Kolmogorov-Donoho optimal approximation for all function classes that are optimally approximated by Wilson dictionaries (generated by functions that can be approximated well by neural networks). Specifically, we point out that the proofs of Lemma IX.2 and Theorem IX.3 explicitly construct neural network approximations of time-shifted and cosine- and sine-modulated versions of the generator g. As identified in Table 1, Wilson bases provide optimal nonlinear approximation of (unit) balls in modulation spaces [85], [74]. Finally, we note that similarly the techniques developed in the proofs of Lemma IX.2 and Theorem IX.3 can be used to establish optimal representability of Fourier bases.

X. Improving Polynomial Approximation Rates to Exponential Rates

Having established that for all function classes listed in Table 1, Kolmogorov-Donoho-optimal approximation through neural networks is possible, this section proceeds to show that neural networks, in addition to their striking Kolmogorov-Donoho universality property, can also do something that has no classical equivalent.

Specifically, as mentioned in the introduction, for the class of oscillatory textures as considered below and for the Weierstrass function, there are no known methods that achieve exponential accuracy, i.e., an approximation error that decays exponentially in the number of parameters employed in the approximant. We establish below that deep networks fill this gap.

Let us start by defining one-dimensional "oscillatory textures" according to [18]. To this end, we recall the following definition from Lemma A.6,

$$\mathcal{S}_{[a,b]} = \left\{ f \in C^\infty([a,b],\mathbb{R})\colon \|f^{(n)}(x)\|_{L^\infty([a,b])} \leq n!, \text{ for all } n \in \mathbb{N}_0 \right\}.$$

Definition X.1. *Let the sets $\mathcal{F}_{D,a}$, $D, a \in \mathbb{R}_+$, be given by*

$$\mathcal{F}_{D,a} = \left\{ \cos(ag)h\colon g, h \in \mathcal{S}_{[-D,D]} \right\}.$$

The efficient approximation of functions in $\mathcal{F}_{D,a}$ with a large represents a notoriously difficult problem due to the combination of the rapidly oscillating cosine term and the warping function g. The best approximation results available in the literature [18] are based on wave-atom dictionaries[11] and yield low-order polynomial approximation rates. In what follows we show that finite-width deep networks drastically improve these results to exponential approximation rates.

We start with our statement on the neural network approximation of oscillatory textures.

Proposition X.2. *There exists a constant $C > 0$ such that for all $D, a \in \mathbb{R}_+, f \in \mathcal{F}_{D,a}$, and $\varepsilon \in (0, 1/2)$, there is a network $\Gamma_{f,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying*

$$\|f - \Gamma_{f,\varepsilon}\|_{L^\infty([-D,D])} \leq \varepsilon,$$

with $\mathcal{L}(\Gamma_{f,\varepsilon}) \leq C\lceil D \rceil((\log(\varepsilon^{-1}) + \log(\lceil a \rceil))^2 + \log(\lceil D \rceil) + \log(\lceil D^{-1} \rceil))$, $\mathcal{W}(\Gamma_{f,\varepsilon}) \leq 32$, $\mathcal{B}(\Gamma_{f,\varepsilon}) \leq 1$.

Proof. For all $D, a \in \mathbb{R}_+$, $f \in \mathcal{F}_{D,a}$, let $g_f, h_f \in \mathcal{S}_{[-D,D]}$ be functions such that $f = \cos(ag_f)h_f$. Note that Lemma A.6 guarantees the existence of a constant $C_1 > 0$ such that for all $D, a \in \mathbb{R}_+$, $\varepsilon \in (0, 1/2)$, there are networks $\Psi_{g_f,\varepsilon}, \Psi_{h_f,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying

$$\|\Psi_{g_f,\varepsilon} - g_f\|_{L^\infty([-D,D])} \leq \tfrac{\varepsilon}{12\lceil a \rceil}, \quad \|\Psi_{h_f,\varepsilon} - h_f\|_{L^\infty([-D,D])} \leq \tfrac{\varepsilon}{12\lceil a \rceil} \tag{87}$$

[11] To be precise, the results of [18] are concerned with the two-dimensional case, whereas here we focus on the one-dimensional case. Note, however, that all our results are readily extended to the multi-dimensional case.

with

$$\mathcal{L}(\Psi_{g_f,\varepsilon}), \mathcal{L}(\Psi_{h_f,\varepsilon}) \leq C_1 \lceil D \rceil (\log((\tfrac{\varepsilon}{12\lceil a \rceil})^{-1})^2 + \log(\lceil D \rceil) + \log(\lceil D^{-1} \rceil)),$$

$\mathcal{W}(\Psi_{g_f,\varepsilon}), \mathcal{W}(\Psi_{h_f,\varepsilon}) \leq 16$, and $\mathcal{B}(\Psi_{g_f,\varepsilon}), \mathcal{B}(\Psi_{h_f,\varepsilon}) \leq 1$. Furthermore, Theorem III.8 ensures the existence of a constant $C_2 > 0$ such that for all $D, a \in \mathbb{R}_+$, $\varepsilon \in (0, 1/2)$, there is a neural network $\Phi_{a,D,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying

$$\|\Phi_{a,D,\varepsilon} - \cos(a\,\cdot)\|_{L^\infty([-3/2,3/2])} \leq \tfrac{\varepsilon}{3}, \tag{88}$$

with $\mathcal{L}(\Phi_{a,D,\varepsilon}) \leq C_2((\log(\varepsilon^{-1}))^2 + \log(\lceil 3a/2 \rceil))$, $\mathcal{W}(\Phi_{a,D,\varepsilon}) \leq 9$, and $\mathcal{B}(\Phi_{a,D,\varepsilon}) \leq 1$. Moreover, due to Proposition III.3, there exists a constant $C_3 > 0$ such that for all $\varepsilon \in (0, 1/2)$, there is a network $\mu_\varepsilon \in \mathcal{N}_{2,1}$ satisfying

$$\sup_{x,y \in [-3/2,3/2]} |\mu_\varepsilon(x, y) - xy| \leq \tfrac{\varepsilon}{3}, \tag{89}$$

with $\mathcal{L}(\mu_\varepsilon) \leq C_3 \log(\varepsilon^{-1})$, $\mathcal{W}(\mu_\varepsilon) \leq 5$, and $\mathcal{B}(\mu_\varepsilon) \leq 1$. By Lemma II.3 there exists a network Ψ^1 satisfying $\Psi^1 = \Phi_{a,D,\varepsilon} \circ \Psi_{g_f,\varepsilon}$ with $\mathcal{W}(\Psi^1) \leq 16$, $\mathcal{L}(\Psi^1) = \mathcal{L}(\Phi_{a,D,\varepsilon}) + \mathcal{L}(\Psi_{g_f,\varepsilon})$, and $\mathcal{B}(\Psi^1) \leq 1$. Furthermore, combining Lemma II.4 and Lemma A.7, we can conclude the existence of a network $\Psi^2(x) = (\Psi^1(x), \Psi_{h_f,\varepsilon}(x)) = (\Phi_{a,D,\varepsilon}(\Psi_{g_f,\varepsilon}(x)), \Psi_{h_f,\varepsilon}(x))$ with $\mathcal{W}(\Psi^2) \leq 32$, $\mathcal{L}(\Psi^2) = \max\{\mathcal{L}(\Phi_{a,D,\varepsilon}) + \mathcal{L}(\Psi_{g_f,\varepsilon}), \mathcal{L}(\Psi_{h_f,\varepsilon})\}$, and $\mathcal{B}(\Psi^2) \leq 1$. Next, for all $D, a \in \mathbb{R}_+$, $f \in \mathcal{F}_{D,a}$, $\varepsilon \in (0, 1/2)$, we define the network $\Gamma_{f,\varepsilon} := \mu_\varepsilon \circ \Psi^2$. By (87), (88), and $\sup_{x\in\mathbb{R}} |\frac{d}{dx} \cos(ax)| = a$, we have, for all $x \in [-D, D]$,

$$\begin{aligned}
|\Phi_{a,D,\varepsilon}(\Psi_{g_f,\varepsilon}(x)) - \cos(ag_f(x))| &\leq |\Phi_{a,D,\varepsilon}(\Psi_{g_f,\varepsilon}(x)) - \cos(a\Psi_{g_f,\varepsilon}(x))| \\
&\quad + |\cos(a\Psi_{g_f,\varepsilon}(x)) - \cos(ag_f(x))| \\
&\leq \tfrac{\varepsilon}{3} + a\tfrac{\varepsilon}{12\lceil a \rceil} \leq \tfrac{5\varepsilon}{12}.
\end{aligned}$$

Combining this with (87), (89), and $\|\cos\|_{L^\infty([-D,D])}, \|f\|_{L^\infty([-D,D])} \leq 1$ yields for all $x \in [-D, D]$,

$$\begin{aligned}
|\Gamma_{f,\varepsilon}(x) - f(x)| &= |\mu_\varepsilon(\Phi_{a,D,\varepsilon}(\Psi_{g_f,\varepsilon}(x)), \Psi_{h_f,\varepsilon}(x)) - \cos(ag_f(x))h_f(x)| \\
&\leq |\mu_\varepsilon(\Phi_{a,D,\varepsilon}(\Psi_{g_f,\varepsilon}(x)), \Psi_{h_f,\varepsilon}(x)) - \Phi_{a,D,\varepsilon}(\Psi_{g_f,\varepsilon}(x))\Psi_{h_f,\varepsilon}(x)| \\
&\quad + |\Phi_{a,D,\varepsilon}(\Psi_{g_f,\varepsilon}(x))\Psi_{h_f,\varepsilon}(x) - \cos(ag_f(x))\Psi_{h_f,\varepsilon}(x)| \\
&\quad + |\cos(ag_f(x))\Psi_{h_f,\varepsilon}(x) - \cos(ag_f(x))h_f(x)| \\
&\leq \tfrac{\varepsilon}{3} + \tfrac{5\varepsilon}{12}\left(1 + \tfrac{\varepsilon}{12\lceil a \rceil}\right) + \tfrac{\varepsilon}{12\lceil a \rceil} \leq \varepsilon.
\end{aligned}$$

Finally, by Lemma II.3 there exists a constant C_4 such that for all $D, a \in \mathbb{R}_+$, $f \in \mathcal{F}_{D,a}$, $\varepsilon \in (0, 1/2)$, it holds that $\mathcal{W}(\Gamma_{f,\varepsilon}) \leq 32$,

$$\begin{aligned}
\mathcal{L}(\Gamma_{f,\varepsilon}) &\leq \mathcal{L}(\mu_\varepsilon) + \max\{\mathcal{L}(\Phi_{a,D,\varepsilon}) + \mathcal{L}(\Psi_{g_f,\varepsilon}), \mathcal{L}(\Psi_{h_f,\varepsilon})\} \\
&\leq C_4 \lceil D \rceil ((\log(\varepsilon^{-1}) + \log(\lceil a \rceil))^2 + \log(\lceil D \rceil) + \log(\lceil D^{-1} \rceil)),
\end{aligned}$$

and $\mathcal{B}(\Gamma_{f,\varepsilon}) \leq 1$. □

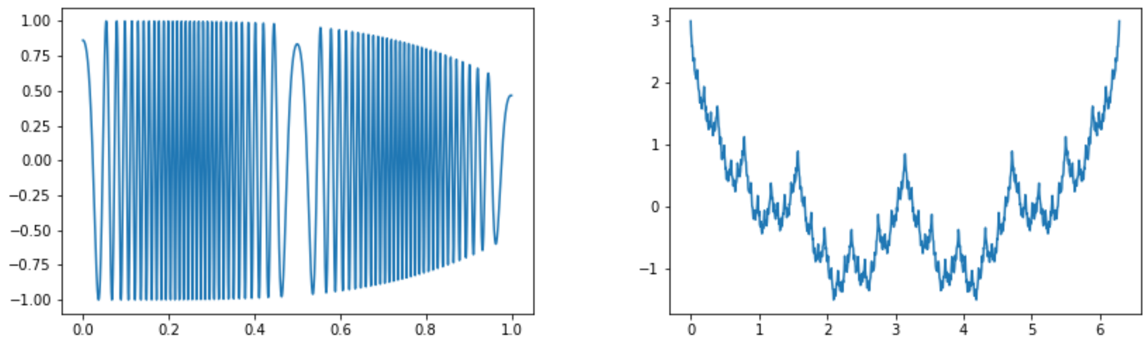


Fig. 4: Left: A function in $\mathcal{F}_{1,100}$. Right: The function $W_{\frac{1}{\sqrt{2}},2}$.

Finally, we show how the Weierstrass function—a fractal function, which is continuous everywhere but differentiable nowhere—can be approximated with exponential accuracy by deep ReLU networks. Specifically, we consider

$$W_{p,a}(x) = \sum_{k=0}^{\infty} p^k \cos(a^k \pi x), \quad \text{for } p \in (0,1/2),\ a \in \mathbb{R}_+,\ \text{with } ap \geq 1,$$

and let $\alpha = -\frac{\log(p)}{\log(a)}$, see Figure 4 right for an example. It is well known [86] that $W_{p,a}$ possesses Hölder smoothness α which may be made arbitrarily small by suitable choice of a. While classical approximation methods achieve polynomial approximation rates only, it turns out that finite-width deep networks yield exponential approximation rates. This is formalized as follows.

Proposition X.3. *There exists a constant $C > 0$ such that for all $\varepsilon, p \in (0,1/2)$, $D, a \in \mathbb{R}_+$, there is a network $\Psi_{p,a,D,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying*

$$\|\Psi_{p,a,D,\varepsilon} - W_{p,a}\|_{L^\infty([-D,D])} \leq \varepsilon,$$

with $\mathcal{L}(\Psi_{p,a,D,\varepsilon}) \leq C((\log(\varepsilon^{-1}))^3 + (\log(\varepsilon^{-1}))^2 \log(\lceil a \rceil) + \log(\varepsilon^{-1}) \log(\lceil D \rceil))$, $\mathcal{W}(\Psi_{p,a,D,\varepsilon}) \leq 13$, $\mathcal{B}(\Psi_{p,a,D,\varepsilon}) \leq 1$.

Proof. For every $N \in \mathbb{N}$, $p \in (0,1/2)$, $a \in \mathbb{R}_+$, $x \in \mathbb{R}$, let $S_{N,p,a}(x) = \sum_{k=0}^{N} p^k \cos(a^k \pi x)$ and note that

$$|S_{N,p,a}(x) - W_{p,a}(x)| \leq \sum_{k=N+1}^{\infty} |p^k \cos(a^k \pi x)| \leq \sum_{k=N+1}^{\infty} p^k = \tfrac{1}{1-p} - \tfrac{1-p^{N+1}}{1-p} \leq 2^{-N}. \tag{90}$$

Let $N_\varepsilon := \lceil \log(2/\varepsilon) \rceil$ for $\varepsilon \in (0,1/2)$. Next, note that Theorem III.8 ensures the existence of a constant $C_1 > 0$ such that for all $D, a \in \mathbb{R}_+$, $k \in \mathbb{N}_0$, $\varepsilon \in (0,1/2)$, there is a network $\phi_{a^k,D,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying

$$\|\phi_{a^k,D,\varepsilon} - \cos(a^k \pi \,\cdot\,)\|_{L^\infty([-D,D])} \leq \tfrac{\varepsilon}{4}, \tag{91}$$

with $\mathcal{L}(\phi_{a^k,D,\varepsilon}) \leq C_1((\log(\varepsilon^{-1}))^2 + \log(\lceil a^k \pi D \rceil))$, $\mathcal{W}(\phi_{a^k,D,\varepsilon}) \leq 9$, $\mathcal{B}(\phi_{a^k,D,\varepsilon}) \leq 1$. Let $A\colon \mathbb{R}^3 \to \mathbb{R}^3$ and $B\colon \mathbb{R}^3 \to \mathbb{R}$ be the affine transformations given by $A(x_1,x_2,x_3) = (x_1,x_1,x_2+x_3)^T$ and $B(x_1,x_2,x_3) = x_2+x_3$, respectively. We now define, for all $p \in (0,1/2)$, $D,a \in \mathbb{R}_+$, $k \in \mathbb{N}_0$, $\varepsilon \in (0,1/2)$, the networks

$$\psi^{p,a,0}_{D,\varepsilon}(x) = \begin{pmatrix} x \\ p^0 \phi_{a^0,D,\varepsilon}(x) \\ 0 \end{pmatrix} \quad \text{and} \quad \psi^{p,a,k}_{D,\varepsilon}(x_1,x_2,x_3) = \begin{pmatrix} x_1 \\ p^k \phi_{a^k,D,\varepsilon}(x_2) \\ x_3 \end{pmatrix}, \; k > 0,$$

and, for all $p \in (0,1/2)$, $D,a \in \mathbb{R}_+$, $\varepsilon \in (0,1/2)$, the network

$$\Psi_{p,a,D,\varepsilon} := B \circ \psi^{p,a,N_\varepsilon}_{D,\varepsilon} \circ A \circ \psi^{p,a,N_\varepsilon-1}_{D,\varepsilon} \circ \cdots \circ A \circ \psi^{p,a,0}_{D,\varepsilon}.$$

Due to (91) we get, for all $p \in (0,1/2)$, $D,a \in \mathbb{R}_+$, $\varepsilon \in (0,1/2)$, $x \in [-D,D]$, that

$$\begin{aligned} |\Psi_{p,a,D,\varepsilon}(x) - S_{N_\varepsilon,p,a}(x)| &= \left| \sum_{k=0}^{N_\varepsilon} p^k \phi_{a^k,D,\varepsilon}(x) - \sum_{k=0}^{N_\varepsilon} p^k \cos(a^k \pi x) \right| \\ &\leq \sum_{k=0}^{N_\varepsilon} p^k |\phi_{a^k,D,\varepsilon}(x) - \cos(a^k \pi x)| \leq \tfrac{\varepsilon}{4} \sum_{k=0}^{N_\varepsilon} 2^{-k} \leq \tfrac{\varepsilon}{2}. \end{aligned}$$

Combining this with (90) establishes, for all $p \in (0,1/2)$, $D,a \in \mathbb{R}_+$, $\varepsilon \in (0,1/2)$, $x \in [-D,D]$,

$$|\Psi_{p,a,D,\varepsilon}(x) - W_{p,a}(x)| \leq 2^{-\lceil \log(\frac{2}{\varepsilon}) \rceil} + \tfrac{\varepsilon}{2} \leq \tfrac{\varepsilon}{2} + \tfrac{\varepsilon}{2} = \varepsilon.$$

Applying Lemmas II.3, II.4, and II.5 establishes the existence of a constant C_2 such that for all $p \in (0,1/2)$, $D,a \in \mathbb{R}_+$, $\varepsilon \in (0,1/2)$,

$$\begin{aligned} \mathcal{L}(\Psi_{p,a,D,\varepsilon}) &\leq \sum_{k=0}^{N_\varepsilon} (\mathcal{L}(\phi_{a^k,D,\varepsilon}) + 1) \leq N_\varepsilon + 1 + (N_\varepsilon + 1) C_1((\log(\varepsilon^{-1}))^2 + \log(\lceil a^{N_\varepsilon} \pi D \rceil)) \\ &\leq C_2((\log(\varepsilon^{-1}))^3 + (\log(\varepsilon^{-1}))^2 \log(\lceil a \rceil) + \log(\varepsilon^{-1}) \log(\lceil D \rceil)), \end{aligned}$$

$\mathcal{W}(\Psi_{p,a,D,\varepsilon}) \leq 13$, and $\mathcal{B}(\Psi_{p,a,D,\varepsilon}) \leq 1$. □

We finally note that the restriction $p \in (0,1/2)$ in Proposition X.3 was made for simplicity of exposition and can be relaxed to $p \in (0,r)$, with $r < 1$, while only changing the constant C.

XI. Impossibility results for finite-depth networks

The recent successes of neural networks in machine learning applications have been enabled by various technological factors, but they all have in common the use of deep networks as opposed to shallow networks studied intensely in the 1990s. It is hence of interest to understand whether the use of depth offers fundamental advantages. In this spirit, the goal of this section is to make a formal case for depth in neural network approximation by establishing that, for nonconstant periodic functions, finite-width deep networks require asymptotically—in the function's "highest frequency"—smaller connectivity than finite-depth wide networks. This statement is then extended to sufficiently smooth nonperiodic functions, thereby formalizing the benefit of deep networks over shallow networks for the approximation of a broad class of functions.

We start with preparatory material taken from [26].

Definition XI.1 ([26])**.** *Let $k \in \mathbb{N}$. A function $f : \mathbb{R} \to \mathbb{R}$ is called k-sawtooth if it is piecewise linear with no more than k pieces, i.e., its domain $\mathbb{R}$ can be partitioned into k intervals such that f is linear on each of these intervals.*

Lemma XI.2 ([26])**.** *Every $\Phi \in \mathcal{N}_{1,1}$ is $(2\mathcal{W}(\Phi))^{\mathcal{L}(\Phi)}$-sawtooth.*

Definition XI.3. *For a u-periodic function $f \in C(\mathbb{R})$, we define*

$$\xi(f) := \sup_{\delta \in [0,u)} \inf_{c,d \in \mathbb{R}} \|f(x) - (cx + d)\|_{L^\infty([\delta, \delta+u])}.$$

The quantity $\xi(f)$ measures the error incurred by the best linear approximation of f on any segment of length equal to the period of f; $\xi(f)$ can hence be interpreted as quantifying the nonlinearity of f. The next result states that finite-depth networks with width and hence also connectivity scaling polylogarithmically in the "highest frequency" of the periodic function to be approximated can not achieve arbitrarily small approximation error.

Proposition XI.4. *Let $f \in C(\mathbb{R})$ be a nonconstant u-periodic function, $L \in \mathbb{N}$, and π a polynomial. Then, there exists an $a \in \mathbb{N}$ such that for every network $\Phi \in \mathcal{N}_{1,1}$ with $\mathcal{L}(\Phi) \leq L$ and $\mathcal{W}(\Phi) \leq \pi(\log(a))$, we have*

$$\|f(a\,\cdot) - \Phi\|_{L^\infty([0,u])} \geq \xi(f) > 0.$$

Proof. First note that there exists an even $a \in \mathbb{N}$ such that $a/2 > (2\pi(\log(a)))^L$. Lemma XI.2 now implies that every network $\Phi \in \mathcal{N}_{1,1}$ with $\mathcal{L}(\Phi) \leq L$ and $\mathcal{W}(\Phi) \leq \pi(\log(a))$ is $(2\pi(\log(a)))^L$-sawtooth and therefore consists of no more than $a/2$ different linear pieces. Hence, there exists an interval $[u_1, u_2] \subseteq [0, u]$ with $u_2 - u_1 \geq (2u/a)$ on which Φ is linear. Since $u_2 - u_1 \geq (2u/a)$ the interval supports two full periods of $f(a\,\cdot)$ and we can therefore conclude that

$$\begin{aligned}\|f(a\,\cdot) - \Phi\|_{L^\infty([0,u])} \geq \|f(a\,\cdot) - \Phi\|_{L^\infty([u_1,u_2])} &\geq \inf_{c,d \in \mathbb{R}} \|f(x) - (cx + d)\|_{L^\infty([0,2u])} \\ &\geq \sup_{\delta \in [0,u)} \inf_{c,d \in \mathbb{R}} \|f(x) - (cx + d)\|_{L^\infty([\delta, u+\delta])} = \xi(f).\end{aligned}$$

Finally, note that $\xi(f) > 0$ as $\xi(f) = 0$ for u-periodic $f \in C(\mathbb{R})$ necessarily implies that f is constant, which, however, is ruled out by assumption. □

Application of Proposition XI.4 to $f(x) = \cos(x)$ shows that finite-depth networks, owing to $\xi(\cos) > 0$, require faster than polylogarithmic growth of connectivity in a to approximate $x \mapsto \cos(ax)$ with arbitrarily small error, whereas finite-width networks, due to Theorem III.8, can accomplish this with polylogarithmic connectivity growth.

The following result from [87] allows a similar observation for functions that are sufficiently smooth.

Theorem XI.5 ([87])**.** *Let $[a, b] \subseteq \mathbb{R}$, $f \in C^3([a, b])$, and for $\varepsilon \in (0, 1/2)$, let $s(\varepsilon) \in \mathbb{N}$ denote the smallest number such that there exists a piecewise linear approximation of f with $s(\varepsilon)$ pieces and error at most ε in $L^\infty([a, b])$-norm. Then, it holds that*

$$s(\varepsilon) \sim \frac{c}{\sqrt{\varepsilon}}, \ \varepsilon \to 0, \ \ \textit{where} \ \ c = \frac{1}{4}\int_a^b \sqrt{|f''(x)|}dx.$$

Combining this with Lemma XI.2 yields the following result on depth-width tradeoff for three-times continuously differentiable functions.

Theorem XI.6. *Let $f \in C^3([a, b])$ with $\int_a^b \sqrt{|f''(x)|}dx > 0$, $L \in \mathbb{N}$, and π a polynomial. Then, there exists $\varepsilon > 0$ such that for every network $\Phi \in \mathcal{N}_{1,1}$ with $\mathcal{L}(\Phi) \leq L$ and $\mathcal{W}(\Phi) \leq \pi(\log(\varepsilon^{-1}))$, we have*

$$\|f - \Phi\|_{L^\infty([a,b])} > \varepsilon.$$

Proof. The proof will be effected by contradiction. Assume that for every $\varepsilon > 0$, there exists a network $\Phi_\varepsilon \in \mathcal{N}_{1,1}$ with $\mathcal{L}(\Phi_\varepsilon) \leq L$, $\mathcal{W}(\Phi_\varepsilon) \leq \pi(\log(\varepsilon^{-1}))$, and $\|f - \Phi_\varepsilon\|_{L^\infty([a,b])} \leq \varepsilon$. By Lemma XI.2 every (ReLU) neural network realizes a piecewise linear function. Application of Theorem XI.5 hence allows us to conclude the existence of a constant C such that, for all $\varepsilon > 0$, the network Φ_ε must have at least $C\varepsilon^{-\frac{1}{2}}$ different linear pieces. This, however, leads to a contradiction as, by Lemma XI.2, Φ_ε is at most $(2\pi(\log(\varepsilon^{-1})))^L$-sawtooth and $\tilde{\pi}(\log(\varepsilon^{-1})) \in o(\varepsilon^{-1/2})$, $\varepsilon \to 0$, for every polynomial $\tilde{\pi}$. □

In summary, we have hence established that any function which is at least three times continuously differentiable (and does not have a vanishing second derivative) cannot be approximated by finite-depth networks with connectivity scaling polylogarithmically in the inverse of the approximation error. Our results in Section III establish that, in contrast, this "is" possible with finite-width deep networks for various interesting types of smooth functions such as polynomials and sinusoidal functions. Further results on the limitations of finite-depth networks akin to Theorem XI.6 were reported in [23].

Acknowledgments

The authors are indebted to R. Gül and W. Ou for their careful proofreading of the paper, to E. Riegler and the reviewers for their constructive and insightful comments, and to the handling editor, P. Narayan, for his helpful comments and his patience.

APPENDIX A

AUXILIARY NEURAL NETWORK CONSTRUCTIONS

The following three results are concerned with the realization of affine transformations of arbitrary weights by neural networks with weights upper-bounded by 1.

Lemma A.1. *Let $d \in \mathbb{N}$ and $a \in \mathbb{R}$. There exists a network $\Phi_a \in \mathcal{N}_{d,d}$ satisfying $\Phi_a(x) = ax$, with $\mathcal{L}(\Phi_a) \leq \lfloor \log(|a|) \rfloor + 4$, $\mathcal{W}(\Phi_a) \leq 3d$, $\mathcal{B}(\Phi_a) \leq 1$.*

Proof. First note that for $|a| \leq 1$ the claim holds trivially, which can be seen by taking Φ_a to be the affine transformation $x \mapsto ax$ and interpreting it according to Definition II.1 as a depth-1 neural network. Next, we consider the case $|a| > 1$ for $d = 1$, set $K := \lfloor \log(a) \rfloor$, $\alpha := a2^{-(K+1)}$, and define $A_1 := (1, -1)^T \in \mathbb{R}^{2\times 1}$,

$$A_2 := \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 2}, \quad A_k := \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 3}, \quad k \in \{3, \ldots, K+3\},$$

and $A_{K+4} := (\alpha, 0, -\alpha)$. Note that $(\rho \circ A_2 \circ \rho \circ A_1)(x) = (\rho(x), \rho(x) + \rho(-x), \rho(-x))$ and $\rho(A_k(x, x+y, y)^T) = 2(x, x+y, y)$, for $k \in \{3, \ldots, K+3\}$. The network $\Psi_a := A_{K+4} \circ \rho \circ \cdots \circ \rho \circ A_1$ hence satisfies $\Psi_a(x) = ax$, $\mathcal{L}(\Psi_a) = \lfloor \log(a) \rfloor + 4$, $\mathcal{W}(\Psi_a) = 3$, and $\mathcal{B}(\Phi_a) \leq 1$. Applying Lemma II.5 to get a parallelization of d copies of Ψ_a completes the proof. □

Corollary A.2. *Let $d, d' \in \mathbb{N}$, $a \in \mathbb{R}_+$, $A \in [-a, a]^{d' \times d}$, and $b \in [-a, a]^{d'}$. There exists a network $\Phi_{A,b} \in \mathcal{N}_{d,d'}$ satisfying $\Phi_{A,b}(x) = Ax + b$, with $\mathcal{L}(\Phi_{A,b}) \leq \lfloor \log(|a|) \rfloor + 5$, $\mathcal{W}(\Phi_{A,b}) \leq \max\{d, 3d'\}$, $\mathcal{B}(\Phi_{A,b}) \leq 1$.*

Proof. Let $\Phi_a \in \mathcal{N}_{d',d'}$ be the multiplication network from Lemma A.1, consider $W(x) := a^{-1}(Ax + b)$ as a 1-layer network, and take $\Phi_{A,b} := \Phi_a \circ W$ according to Lemma II.3. □

Proposition A.3. *Let $d, d' \in \mathbb{N}$ and $\Phi \in \mathcal{N}_{d,d'}$. There exists a network $\Psi \in \mathcal{N}_{d,d'}$ satisfying $\Psi(x) = \Phi(x)$, for all $x \in \mathbb{R}^d$, and with $\mathcal{L}(\Psi) \leq (\lceil \log(\mathcal{B}(\Phi)) \rceil + 5)\mathcal{L}(\Phi)$, $\mathcal{W}(\Psi) \leq \max\{3d', \mathcal{W}(\Phi)\}$, $\mathcal{B}(\Psi) \leq 1$.*

Proof. We write $\Phi = W_{\mathcal{L}(\Phi)} \circ \rho \circ \ldots \circ \rho \circ W_1$ and set $\widetilde{W_\ell} := (\mathcal{B}(\Phi))^{-1} W_\ell$, for $\ell \in \{1, \ldots, \mathcal{L}(\Phi)\}$, and $a := \mathcal{B}(\Phi)^{\mathcal{L}(\Phi)}$. Let $\Phi_a \in \mathcal{N}_{d',d'}$ be the multiplication network from Lemma A.1 and define

$$\widetilde{\Phi} := \widetilde{W}_{\mathcal{L}(\Phi)} \circ \rho \circ \cdots \circ \rho \circ \widetilde{W_1},$$

and $\Psi := \Phi_a \circ \widetilde{\Phi}$ according to Lemma II.3. Note that $\widetilde{\Phi}$ has weights upper-bounded by 1 and is of the same depth and width as Φ. As ρ is positively homogeneous, i.e., $\rho(\lambda x) = \lambda \rho(x)$, for all $\lambda \geq 0$, $x \in \mathbb{R}$, we have $\Psi(x) = \Phi(x)$, for all $x \in \mathbb{R}^d$. Application of Lemma II.3 and Lemma A.1 completes the proof. □

Next we record a technical Lemma on how to realize a sum of networks with the same input by a network whose width is independent of the number of constituent networks.

Lemma A.4. *Let $d, d' \in \mathbb{N}$, $N \in \mathbb{N}$, and $\Phi_i \in \mathcal{N}_{d,d'}$, $i \in \{1, \dots, N\}$. There exists a network $\Phi \in \mathcal{N}_{d,d'}$ satisfying*

$$\Phi(x) = \sum_{i=1}^{N} \Phi_i(x), \quad \textit{for all}\, x \in \mathbb{R}^d,$$

with $\mathcal{L}(\Phi) = \sum_{i=1}^{N} \mathcal{L}(\Phi_i)$, $\mathcal{W}(\Phi) \leq 2d + 2d' + \max\{2d, \max_i\{\mathcal{W}(\Phi_i)\}\}$, $\mathcal{B}(\Phi) = \max\{1, \max_i \mathcal{B}(\Phi_i)\}$.

Proof. We set $L_i = \mathcal{L}(\Phi_i)$ and write the networks Φ_i as

$$\Phi_i = W^i_{L_i} \circ \rho \circ W^i_{L_i - 1} \circ \rho \circ \cdots \circ \rho \circ W^i_1,$$

with $W^i_\ell(x) = A^i_\ell x + b^i_\ell$, where $A^i_\ell \in \mathbb{R}^{N^i_\ell \times N^i_{\ell-1}}$ and $b^i_\ell \in \mathbb{R}^{N^i_\ell}$. Next, using Lemma II.4, we turn the identity matrices $\mathbb{I}_d$ and $\mathbb{I}_{d'}$ into networks $\mathbb{I}^i_d$ and $\mathbb{I}^i_{d'}$, respectively, of depth L_i and then parallelize these networks, according to Lemma II.5, to get $\Psi_i := (\mathbb{I}^i_d, \mathbb{I}^i_{d'}, \Phi_i)$. Let $V^i_1(x) = E^i_1 x + f^i_1$ and $V^i_{L_i}(x) = E^i_{L_i} x + f^i_{L_i}$ denote the first and last, respectively, affine transformation of the network Ψ_i. By construction we have

$$E^i_1 = \begin{pmatrix} \mathbb{I}_d & 0 & 0 \\ -\mathbb{I}_d & 0 & 0 \\ 0 & \mathbb{I}_{d'} & 0 \\ 0 & -\mathbb{I}_{d'} & 0 \\ 0 & 0 & A^i_1 \end{pmatrix} \in \mathbb{R}^{(2d+2d'+N^i_1)\times(2d+d')}, \quad f^i_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ b^i_1 \end{pmatrix} \in \mathbb{R}^{2d+2d'+N^i_1}$$

and

$$E^i_{L_i} = \begin{pmatrix} \mathbb{I}_d & -\mathbb{I}_d & 0 & 0 & 0 \\ 0 & 0 & \mathbb{I}_{d'} & -\mathbb{I}_{d'} & 0 \\ 0 & 0 & 0 & 0 & A^i_{L_i} \end{pmatrix} \in \mathbb{R}^{(d+2d')\times(2d+2d'+N^i_{L_i-1})}, \quad f^i_{L_i} = \begin{pmatrix} 0 \\ 0 \\ b^i_{L_i} \end{pmatrix} \in \mathbb{R}^{d+2d'}.$$

Next, we define the matrices

$$A_{\text{in}} := \begin{pmatrix} \mathbb{I}_d \\ 0 \\ \mathbb{I}_d \end{pmatrix} \in \mathbb{R}^{(2d+d')\times d}, \quad A := \begin{pmatrix} \mathbb{I}_d & 0 & 0 \\ 0 & \mathbb{I}_{d'} & \mathbb{I}_{d'} \\ \mathbb{I}_d & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(2d+d')\times(d+2d')},$$

$$A_{\text{out}} := \begin{pmatrix} 0 & \mathbb{I}_{d'} & \mathbb{I}_{d'} \end{pmatrix} \in \mathbb{R}^{d' \times (d+2d')},$$

and note that $A_{\text{in}} x = (x, 0, x)$, $A(x, y, z)^T = (x, y + z, x)^T$, and $A_{\text{out}}(x, y, z)^T = y + z$, for $x \in \mathbb{R}^d, y, z \in \mathbb{R}^{d'}$. We construct

- the network $\widetilde{\Psi}_1$ by taking Ψ_1 and replacing E^1_1 with $E^1_1 A_{\text{in}}$, $E^1_{L_1}$ with $AE^1_{L_1}$, and $f^1_{L_1}$ with $Af^1_{L_1}$,
- the network $\widetilde{\Psi}_N$ by taking Ψ_N and replacing $E^N_{L_N}$ with $A_{\text{out}} E^N_{L_N}$ and $f^N_{L_N}$ with $A_{\text{out}} f^N_{L_N}$,
- the networks $\widetilde{\Psi}_i$, $i \in \{2, \dots, N-1\}$ by taking Ψ_i and replacing $E^i_{L_i}$ with $AE^i_{L_i}$ and $f^i_{L_i}$ with $Af^i_{L_i}$.

We can now verify that

$$\Phi = \widetilde{\Psi}_N \circ \widetilde{\Psi}_{N-1} \circ \cdots \circ \widetilde{\Psi}_1,$$

when the compositions are taken in the sense of Lemma II.3. Due to Lemmas II.4 and II.5, we have $\mathcal{L}(\Psi_i) = \mathcal{L}(\Phi_i)$, $\mathcal{W}(\Psi_i) = 2d + 2d' + \mathcal{W}(\Phi_i)$, and $\mathcal{B}(\Psi_i) = \max\{1, \mathcal{B}(\Phi_i)\}$. The proof is finalized by noting that, owing to the structure of the involved matrices, the depth and the weight magnitude remain unchanged by turning Ψ_i into $\widetilde{\Psi}_i$, whereas the width can not increase, but may decrease owing to the replacement of E_1^1 by $E_1^1 A_{\text{in}}$. □

The following lemma shows how to patch together local approximations using multiplication networks and a partition of unity consisting of hat functions. We note that this argument can be extended to higher dimensions using tensor products (which can be realized efficiently through multiplication networks) of the one-dimensional hat function.

Lemma A.5. *Let $\varepsilon \in (0, 1/2)$, $n \in \mathbb{N}$, $a_0 < a_1 < \cdots < a_n \in \mathbb{R}$, $f \in L^\infty([a_0, a_n])$, and*

$$A := \lceil \max\{|a_0|, |a_n|, 2 \max_{i \in \{2,\ldots,n-1\}} \tfrac{1}{|a_i - a_{i-1}|}\} \rceil, \quad B := \max\{1, \|f\|_{L^\infty([a_0,a_n])}\}.$$

Assume that for every $i \in \{1, \ldots, n-1\}$, there exists a network $\Phi_i \in \mathcal{N}_{1,1}$ with $\|f - \Phi_i\|_{L^\infty([a_{i-1},a_{i+1}])} \leq \varepsilon/3$. Then, there is a network $\Phi \in \mathcal{N}_{1,1}$ satisfying

$$\|f - \Phi\|_{L^\infty([a_0,a_n])} \leq \varepsilon,$$

with $\mathcal{L}(\Phi) \leq \big(\sum_{i=1}^{n-1} \mathcal{L}(\Phi_i)\big) + Cn(\log(\varepsilon^{-1}) + \log(B) + \log(A))$, $\mathcal{W}(\Phi) \leq 7 + \max\{2, \max_{i \in \{1,\ldots,n-1\}} \mathcal{W}(\Phi_i)\}$, $\mathcal{B}(\Phi) = \max\{1, \max_i \mathcal{B}(\Phi_i)\}$, and with $C > 0$ an absolute constant, i.e., independent of $\varepsilon, n, f, a_0, \ldots, a_n$.

Proof. We first define the neural networks $(\Psi_i)_{i=1}^{n-1} \in \mathcal{N}_{1,1}$ forming a partition of unity according to

$$\begin{aligned}
\Psi_1(x) &:= 1 - \tfrac{1}{a_2 - a_1}\, \rho(x - a_1) + \tfrac{1}{a_2 - a_1}\, \rho(x - a_2), \\
\Psi_i(x) &:= \tfrac{1}{a_i - a_{i-1}}\, \rho(x - a_{i-1}) - \big(\tfrac{1}{a_i - a_{i-1}} + \tfrac{1}{a_{i+1} - a_i}\big)\, \rho(x - a_i) + \tfrac{1}{a_{i+1} - a_i}\, \rho(x - a_{i+1}), \quad i \in \{2, \ldots, n-2\}, \\
\Psi_{n-1}(x) &:= \tfrac{1}{a_{n-1} - a_{n-2}}\, \rho(x - a_{n-2}) - \tfrac{1}{a_{n-1} - a_{n-2}}\, \rho(x - a_{n-1}).
\end{aligned}$$

Note that $\text{supp}(\Psi_1) = (\infty, a_2)$, $\text{supp}(\Psi_{n-1}) = [a_{n-2}, \infty)$, and $\text{supp}(\Psi_i) = [a_{i-1}, a_{i+1}]$. Proposition A.3 now ensures that, for all $i \in \{1, \ldots, n-1\}$, Ψ_i can be realized as a network with $\mathcal{L}(\Psi_i) \leq 2(\lceil \log(A) \rceil + 5)$, $\mathcal{W}(\Psi_i) \leq 3$, and $\mathcal{B}(\Psi_i) \leq 1$. Next, let $\Phi_{B+1/6,\varepsilon/3} \in \mathcal{N}_{2,1}$ be the multiplication network according to Proposition III.3 and define the networks

$$\widetilde{\Phi}_i(x) := \Phi_{B+1/6,\varepsilon/3}(\Phi_i(x), \Psi_i(x))$$

according to Lemma II.5 and Lemma II.3, along with their sum

$$\Phi(x) := \sum_{i=1}^{n-1} \widetilde{\Phi}_i(x)$$

according to Lemma A.4. Proposition III.3 ensures, for all $i \in \{1, \ldots, n-1\}$, $x \in [a_{i-1}, a_{i+1}]$, that

$$\begin{aligned}|f(x)\Psi_i(x) - \widetilde{\Phi}_i(x)| &\leq |f(x)\Psi_i(x) - \Phi_i(x)\Psi_i(x)| + |\Phi_i(x)\Psi_i(x) - \Phi_{B+1/6,\varepsilon/3}(\Phi_i(x), \Psi_i(x))| \\ &\leq (\Psi_i(x) + 1)\tfrac{\varepsilon}{3}\end{aligned}$$

and $\operatorname{supp}(\widetilde{\Phi}_i) = [a_{i-1}, a_{i+1}]$. In particular, for every $x \in [a_0, a_n]$, the set

$$I(x) := \{i \in \{1, \ldots, n-1\} \colon \widetilde{\Phi}_i(x) \neq 0\}$$

of active indices contains at most two elements. Moreover, we have $\sum_{i \in I(x)} \Psi_i(x) = 1$ by construction, which implies that, for all $x \in \mathbb{R}$,

$$|f(x) - \Phi(x)| = \left| \sum_{i \in I(x)} \Psi_i(x) f(x) - \sum_{i \in I(x)} \tilde{\Phi}_i(x) \right| \leq \sum_{i \in I(x)} (\Psi_i(x) + 1)\tfrac{\varepsilon}{3} \leq \varepsilon.$$

Due to Lemma II.3, Lemma II.5, Proposition III.3, and Lemma A.4, we can conclude that Φ, indeed, satisfies the claimed properties. □

Next, we present an extension of Lemma III.7 to arbitrary (finite) intervals.

Lemma A.6. *For $a, b \in \mathbb{R}$ with $a < b$, let*

$$\mathcal{S}_{[a,b]} := \left\{ f \in C^\infty([a,b], \mathbb{R}) \colon \|f^{(n)}(x)\|_{L^\infty([a,b])} \leq n!, \ \text{ for all } n \in \mathbb{N}_0 \right\}.$$

There exists a constant $C > 0$ such that for all $a, b \in \mathbb{R}$ with $a < b$, $f \in \mathcal{S}_{[a,b]}$, and $\varepsilon \in (0, 1/2)$, there is a network $\Psi_{f,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying

$$\|\Psi_{f,\varepsilon} - f\|_{L^\infty([a,b])} \leq \varepsilon,$$

with $\mathcal{L}(\Psi_{f,\varepsilon}) \leq C \max\{2, (b-a)\}((\log(\varepsilon^{-1}))^2 + \log(\lceil \max\{|a|, |b|\} \rceil) + \log(\lceil \frac{1}{b-a} \rceil))$, $\mathcal{W}(\Psi_{f,\varepsilon}) \leq 16$, $\mathcal{B}(\Psi_{f,\varepsilon}) \leq 1$.

Proof. We first recall that the case $[a,b] = [-1,1]$ has already been dealt with in Lemma III.7. Here, we will first prove the statement for the interval $[-D, D]$ with $D \in (0,1)$ and then use this result to establish the general case through a patching argument according to Lemma A.5. We start by noting that for $g \in \mathcal{S}_{[-D,D]}$, the function $f_g \colon [-1,1] \to \mathbb{R}, x \mapsto g(Dx)$ is in $\mathcal{S}_{[-1,1]}$ due to $D < 1$. Hence, by Lemma III.7, there exists a constant $C > 0$ such that for all $g \in \mathcal{S}_{[-D,D]}$ and $\varepsilon \in (0, 1/2)$, there is a network $\widetilde{\Psi}_{g,\varepsilon} \in \mathcal{N}_{1,1}$ satisfying $\|\widetilde{\Psi}_{g,\varepsilon} - f_g\|_{L^\infty([-1,1])} \leq \varepsilon$, with $\mathcal{L}(\widetilde{\Psi}_{g,\varepsilon}) \leq C(\log(\varepsilon^{-1}))^2$, $\mathcal{W}(\widetilde{\Psi}_{g,\varepsilon}) \leq 9$, $\mathcal{B}(\widetilde{\Psi}_{g,\varepsilon}) \leq 1$. The claim is then established by taking the network approximating g to be $\Psi_{g,\varepsilon} := \widetilde{\Psi}_{g,\varepsilon} \circ \Phi_{D^{-1}}$, where $\Phi_{D^{-1}}$ is the scalar multiplication network from Lemma A.1, and noting that

$$\begin{aligned}\|\Psi_{g,\varepsilon}(x) - g(x)\|_{L^\infty([-D,D])} &= \sup_{x \in [-D,D]} |\widetilde{\Psi}_{g,\varepsilon}(\tfrac{x}{D}) - f_g(\tfrac{x}{D})| \\ &= \sup_{x \in [-1,1]} |\widetilde{\Psi}_{g,\varepsilon}(x) - f_g(x)| \leq \varepsilon.\end{aligned}$$

Due to Lemma II.3, we have $\mathcal{L}(\Psi_{g,\varepsilon}) \leq C((\log(\varepsilon^{-1}))^2 + \log(\lceil \frac{1}{D} \rceil))$, $\mathcal{W}(\Psi_{g,\varepsilon}) \leq 9$, and $\mathcal{B}(\Psi_{g,\varepsilon}) \leq 1$. We are now ready to proceed to the proof of the statement for general intervals $[a, b]$. This will be accomplished by approximating f on intervals of length no more than 2 and stitching the resulting approximations together according to Lemma A.5. We start with the case $b-a \leq 2$ and note that here we can simply shift the function by $(a+b)/2$ to center its domain around the origin and then use the result above for approximation on $[-D, D]$ with $D \in (0, 1)$ or Lemma III.7 if $b - a = 2$, both in combination with Corollary A.2 to realize the shift through a neural network with weights bounded by 1. Using Lemma II.3 to implement the composition of the network realizing this shift with that realizing g, we can conclude the existence of a constant $C' > 0$ such that, for all $[a, b] \subseteq \mathbb{R}$ with $b - a \leq 2$, $g \in \mathcal{S}_{[a,b]}$, $\varepsilon \in (0, 1/2)$, there is a network satisfying $\|g - \Psi_{g,\varepsilon}\|_{L^\infty([a,b])} \leq \varepsilon$ with $\mathcal{L}(\Psi_{g,\varepsilon}) \leq C'((\log(\varepsilon^{-1}))^2 + \log(\lceil \frac{1}{b-a} \rceil))$, $\mathcal{W}(\Psi_{g,\varepsilon}) \leq 9$, and $\mathcal{B}(\Psi_{g,\varepsilon}) \leq 1$. Finally, for $b - a > 2$, we partition the interval $[a, b]$ and apply Lemma A.5 as follows. We set $n := \lceil b - a \rceil$ and define

$$a_i := a + i\tfrac{b-a}{n}, \quad i \in \{0, \dots, n\}.$$

Next, for $i \in \{1, \dots, n-1\}$, let $g_i \colon [a_{i-1}, a_{i+1}] \to \mathbb{R}$ be the restriction of g to the interval $[a_{i-1}, a_{i+1}]$, and note that $a_{i+1} - a_{i-1} = \frac{2(b-a)}{n} \in (\frac{4}{3}, 2]$. Furthermore, for $i \in \{1, \dots, n-1\}$, let $\Psi_{g_i,\varepsilon/3}$ be the network approximating g_i with error $\varepsilon/3$ as constructed above. Then, for every $i \in \{1, \dots, n-1\}$, it holds that $\|g - \Psi_{g_i,\varepsilon/3}\|_{L^\infty([a_{i-1},a_{i+1}])} \leq \frac{\varepsilon}{3}$ and application of Lemma A.5 yields the desired result. □

We finally record, for technical purposes, slight variations of Lemmas II.5 and II.6 to account for parallelizations and linear combinations, respectively, of neural networks with shared input.

Lemma A.7. *Let $n, d, L \in \mathbb{N}$ and, for $i \in \{1, 2, \dots, n\}$, let $d'_i \in \mathbb{N}$ and $\Phi_i \in \mathcal{N}_{d,d'_i}$ with $\mathcal{L}(\Phi_i) = L$. Then, there exists a network $\Psi \in \mathcal{N}_{d,\sum_{i=1}^n d'_i}$ with $\mathcal{L}(\Psi) = L$, $\mathcal{M}(\Psi) = \sum_{i=1}^n \mathcal{M}(\Phi_i)$, $\mathcal{W}(\Psi) \leq \sum_{i=1}^n \mathcal{W}(\Phi_i)$, $\mathcal{B}(\Psi) = \max_i \mathcal{B}(\Phi_i)$, and satisfying*

$$\Psi(x) = (\Phi_1(x), \Phi_2(x), \dots, \Phi_n(x)) \in \mathbb{R}^{\sum_{i=1}^n d'_i},$$

for $x \in \mathbb{R}^d$.

Proof. The claim is established by following the construction in the proof of Lemma II.5, but with the matrix $A_1 = \text{diag}(A_1^1, A_1^2, \dots, A_1^n)$ replaced by

$$A_1 = \begin{pmatrix} A_1^1 \\ \vdots \\ A_1^n \end{pmatrix} \in \mathbb{R}^{(\sum_{i=1}^n N_1^i) \times d},$$

where N_1^i is the dimension of the first layer of Φ_i. □

Lemma A.8. *Let $n, d, d', L \in \mathbb{N}$ and, for $i \in \{1, 2, \dots, n\}$, let $a_i \in \mathbb{R}$ and $\Phi_i \in \mathcal{N}_{d,d'}$ with $\mathcal{L}(\Phi_i) = L$. Then, there exists a network $\Psi \in \mathcal{N}_{d,d'}$ with $\mathcal{L}(\Psi) = L$, $\mathcal{M}(\Psi) \leq \sum_{i=1}^{n} \mathcal{M}(\Phi_i)$, $\mathcal{W}(\Psi) \leq \sum_{i=1}^{n} \mathcal{W}(\Phi_i)$, $\mathcal{B}(\Psi) = \max_i\{|a_i|\mathcal{B}(\Phi_i)\}$, and satisfying*

$$\Psi(x) = \sum_{i=1}^{n} a_i \Phi_i(x) \in \mathbb{R}^{d'},$$

for $x \in \mathbb{R}^d$.

Proof. The proof follows directly from that of Lemma A.7 with the same modifications as those needed in the proof of Lemma II.6 relative to that of Lemma II.5. □

APPENDIX B
TAIL COMPACTNESS FOR BESOV SPACES

We consider the Besov space $B^m_{p,q}([0,1])$ [16] given by the set of functions $f \in L^2([0,1])$ satisfying

$$\|f\|_{m,p,q} := \|(2^{n(m+\frac{1}{2}-\frac{1}{p})}\|(\langle f, \psi_{n,k}\rangle)_{k=0}^{2^n-1}\|_{\ell^p})_{n\in\mathbb{N}_0}\|_{\ell^q} < \infty, \tag{92}$$

with $\mathcal{D} = \{\psi_{n,k} \colon n \in \mathbb{N}_0, k = 0, \dots, 2^n - 1\}$ an orthonormal wavelet basis[12] for $L^2([0,1])$ and ℓ^p denoting the usual sequence norm

$$\|(a_i)_{i\in I}\|_{\ell^p} = \begin{cases} \left(\sum_{i\in I} |a_i|^p\right)^{\frac{1}{p}}, & 1 \leq p < \infty \\ \sup_{i\in I} |a_i|, & p = \infty \end{cases}.$$

The unit ball in $B^m_{p,q}([0,1])$ is

$$\mathcal{U}(B^m_{p,q}([0,1])) = \{f \in L^2([0,1]) \colon \|f\|_{m,p,q} \leq 1\}. \tag{93}$$

For simplicity of notation, we set $a_{n,k}(f) := \langle f, \psi_{n,k}\rangle$ and $A_n(f) := (a_{n,k}(f))_{k=0}^{2^n-1} \in \mathbb{R}^{2^n}$, for $n \in \mathbb{N}_0$. We now want to verify that for $q \in [1,2]$ tail compactness holds for the pair $(\mathcal{U}(B^m_{p,q}([0,1])), \mathcal{D})$ under the ordering $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1, \dots)$, where $\mathcal{D}_n := \{\psi_{n,k} \colon k = 0, \dots, 2^n - 1\}$. To this end, we first note that owing to $\sum_{n=0}^{N} |\mathcal{D}_n| = 2^{N+1} - 1$, we have tail compactness according to (26) if there exist $C, \beta > 0$ such that for all $f \in \mathcal{U}(B^m_{p,q}([0,1]))$, $N \in \mathbb{N}$,

$$\left\| f - \sum_{n=0}^{N} \sum_{k=0}^{2^n-1} a_{n,k}(f)\psi_{n,k} \right\|_{L^2([0,1])} \leq C(2^{N+1})^{-\beta}. \tag{94}$$

To see that (92) implies (94), we note that by orthonormality of $\mathcal{D}$,

$$\left\| f - \sum_{n=0}^{N} \sum_{k=0}^{2^n-1} a_{n,k}(f)\psi_{n,k} \right\|_{L^2([0,1])} = \left\| \sum_{n=N+1}^{\infty} \sum_{k=0}^{2^n-1} a_{n,k}(f)\psi_{n,k} \right\|_{L^2([0,1])} = \left(\sum_{n=N+1}^{\infty} \sum_{k=0}^{2^n-1} |a_{n,k}(f)|^2 \right)^{\frac{1}{2}}$$
$$= \|(\|A_n(f)\|_{\ell^2})_{n=N+1}^{\infty}\|_{\ell^2}.$$

[12] The space does not depend on the particular choice of mother wavelet ψ as long as ψ has at least r vanishing moments and is in $C^r([0,1])$ for some $r > m$. For further details we refer to Section 9.2.3 in [16].

As the $A_n(f)$ are finite sequences of length $|\mathcal{D}_n| = 2^n$, it follows, by application of Hölder's inequality, that $\|A_n(f)\|_{\ell^2} \leq 2^{n(\frac{1}{2}-\frac{1}{p})}\|A_n(f)\|_{\ell^p}$. Together with $\|\cdot\|_{\ell^2} \leq \|\cdot\|_{\ell^q}$, for $q \leq 2$, (92) then ensures, for all $f \in \mathcal{U}(B^m_{p,q}([0,1]))$ and $q \in [1,2]$, that

$$\begin{aligned}\|(\|A_n(f)\|_{\ell^2})_{n=N+1}^{\infty}\|_{\ell^2} &\leq \|(2^{n(\frac{1}{2}-\frac{1}{p})}\|A_n(f)\|_{\ell^p})_{n=N+1}^{\infty}\|_{\ell^q} \leq 2^{-(N+1)m}\|(2^{n(m+\frac{1}{2}-\frac{1}{p})}\|A_n(f)\|_{\ell^p})_{n=N+1}^{\infty}\|_{\ell^q}\\ &\leq 2^{-(N+1)m}\|f\|_{m,p,q} \leq (2^{N+1})^{-m},\end{aligned}$$

which establishes (94) with $C = 1$ and $\beta = m$.

Appendix C
Tail compactness for modulation spaces

We consider tail compactness for unit balls in (polynomially) weighted modulation spaces, which, for $p, q \in [1, \infty)$, are defined as follows

$$M^s_{p,q}(\mathbb{R}) := \{f \colon \|f\|_{M^s_{p,q}(\mathbb{R})} < \infty\},$$

with

$$\|f\|_{M^s_{p,q}(\mathbb{R})} := \left(\int_{\mathbb{R}} \left(\int_{\mathbb{R}} |V_w f(x,\xi)|^p (1+|x|+|\xi|)^{sp} \mathrm{d}x\right)^{\frac{q}{p}} \mathrm{d}\xi\right)^{\frac{1}{q}},$$

where

$$V_w f(x,\xi) := \int_{\mathbb{R}} f(t)\, \overline{w(t-x)} e^{-2\pi i t\xi} \mathrm{d}t, \quad x, \xi \in \mathbb{R},$$

is the short-time Fourier transform of f with respect to the window function[13] $w \in \mathcal{S}(\mathbb{R})$.

Next, let $g \in L^2(\mathbb{R})$ with $\|g\|_{L^2(\mathbb{R})} = 1$ and $g(x) = \overline{g(-x)}$ such that the Gabor dictionary $\mathcal{G}(g, \frac{1}{2}, 1, \mathbb{R})$ is a tight frame [68] for $L^2(\mathbb{R})$. Then, the Wilson dictionary $\mathcal{D} = \{\psi_{k,n} \colon (k,n) \in \mathbb{Z} \times \mathbb{N}_0\}$ with

$$\begin{aligned}\psi_{k,0} &= T_k g, && k \in \mathbb{Z},\\ \psi_{k,n} &= \tfrac{1}{\sqrt{2}} T_{\frac{k}{2}}(M_n + (-1)^{k+n} M_{-n})g, && (k,n) \in \mathbb{Z} \times \mathbb{N},\end{aligned}$$

is an orthonormal basis for $L^2(\mathbb{R})$ (see [17, Thm. 8.5.1]). We have, for every $f \in M^s_{p,q}(\mathbb{R})$, the expansion [17, Thm. 12.3.4]

$$f = \sum_{(k,n)\in\mathbb{Z}\times\mathbb{N}_0} c_{k,n}(f)\psi_{k,n}, \quad \text{where} \quad c_{k,n}(f) = \langle f, \psi_{k,n}\rangle, \quad c(f) \in \ell^s_{p,q}(\mathbb{Z}\times\mathbb{N}_0),$$

[13] The resulting modulation space does not depend on the specific choice of window function w as long as w is in the Schwartz space $\mathcal{S}(\mathbb{R}) = \{f \in C^\infty(\mathbb{R}) \colon \sup_{x\in\mathbb{R}} |x^\alpha f^{(\beta)}(x)| < \infty$, for all $\alpha, \beta \in \mathbb{N}_0\}$, where $f^{(n)}$ stands for the n-th derivative of f.

with $\ell^s_{p,q}(\mathbb{Z}\times\mathbb{N}_0)$ the space of sequences $c\in\mathbb{R}^{\mathbb{Z}\times\mathbb{N}_0}$ satisfying

$$\|c\|_{\ell^s_{p,q}(\mathbb{Z}\times\mathbb{N}_0)} := \left(\sum_{n\in\mathbb{N}_0}\left(\sum_{k\in\mathbb{Z}}|c_{k,n}|^p(1+|\tfrac{k}{2}|+|n|)^{sp}\right)^{\frac{q}{p}}\right)^{\frac{1}{q}} < \infty.$$

Moreover, there exists [17, Thm. 12.3.1] a constant $D\geq 1$ such that, for all $f\in M^s_{p,q}(\mathbb{R})$,

$$\tfrac{1}{D}\|f\|_{M^s_{p,q}(\mathbb{R})} \leq \|c(f)\|_{\ell^s_{p,q}(\mathbb{Z}\times\mathbb{N}_0)} \leq D\|f\|_{M^s_{p,q}(\mathbb{R})}.$$

In particular, we can characterize the unit ball of $M^s_{p,q}(\mathbb{R})$ according to

$$\mathcal{U}(M^s_{p,q}(\mathbb{R})) = \{f\colon \|c(f)\|_{\ell^s_{p,q}(\mathbb{Z}\times\mathbb{N}_0)} \leq D\}.$$

We now order the Wilson basis dictionary as follows. Define $\mathcal{D}_0 := \{\psi_{0,0}\}$ and

$$\mathcal{D}_\ell := \{\psi_{k,n}\colon |k|, n\leq \ell\}\setminus\bigcup_{i=0}^{\ell-1}\mathcal{D}_i$$

for $\ell\geq 1$, and order the overall dictionary according to $\mathcal{D}=(\mathcal{D}_0,\mathcal{D}_1,\dots)$. Owing to $\sum_{\ell=0}^N|\mathcal{D}_\ell| = (2N{+}1)(N{+}1)$, we have tail compactness for the pair $(\mathcal{U}(M^s_{p,q}(\mathbb{R})),\mathcal{D})$ if there exist $C,\beta>0$ such that, for all $f\in\mathcal{U}(M^s_{p,q}(\mathbb{R}))$, $N\in\mathbb{N}$,

$$\left\|f-\sum_{n=0}^N\sum_{k=-N}^N c_{k,n}(f)\psi_{k,n}\right\|_{L^2(\mathbb{R})} \leq CN^{-\beta}. \tag{95}$$

We restrict our attention to $p,q\leq 2$ and use orthonormality of $\mathcal{D}$ and the fact that $\|\cdot\|_{\ell^2}\leq\|\cdot\|_{\ell^p}$, for $p\leq 2$, to obtain, for all $f\in\mathcal{U}(M^s_{p,q}(\mathbb{R}))$,

$$\begin{aligned}\left\|f-\sum_{n=0}^N\sum_{k=-N}^N c_{k,n}(f)\psi_{k,n}\right\|_{L^2(\mathbb{R})} &= \left\|\sum_{n>N}\sum_{|k|>N}c_{k,n}(f)\psi_{k,n}\right\|_{L^2(\mathbb{R})} = \left(\sum_{n>N}\sum_{|k|>N}|c_{k,n}(f)|^2\right)^{\frac{1}{2}}\\ &\leq \left(\sum_{n>N}\left(\sum_{|k|>N}|c_{k,n}(f)|^p\right)^{\frac{q}{p}}\right)^{\frac{1}{q}}\\ &\leq (1+\tfrac{3}{2}N)^{-s}\left(\sum_{n>N}\left(\sum_{|k|>N}|c_{k,n}(f)|^p(1+|\tfrac{k}{2}|+|n|)^{sp}\right)^{\frac{q}{p}}\right)^{\frac{1}{q}}\\ &\leq (1+\tfrac{3}{2}N)^{-s}\|c(f)\|_{\ell^s_{p,q}(\mathbb{Z}\times\mathbb{N}_0)} \leq (3/2)^{-s}DN^{-s},\end{aligned}$$

which establishes tail compactness with $C=(3/2)^{-s}D$ and $\beta=s$.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] Y. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Müller, E. Säckinger, P. Simard, and V. Vapnik, “Comparison of learning algorithms for handwritten digit recognition,” *International Conference on Artificial Neural Networks*, pp. 53–60, 1995.

[3] 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.*, vol. 29, no. 6, pp. 82–97, 2012.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016. [Online]. Available: http://www.nature.com/nature/journal/v529/n7587/abs/nature16961.html#supplementary-information

[5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: http://dx.doi.org/10.1038/nature14539

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986. [Online]. Available: http://dx.doi.org/10.1038/323533a0

[8] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[9] W. McCulloch and W. Pitts, “A logical calculus of ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.

[10] A. N. Kolmogorov, “On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition,” *Dokl. Akad. Nauk SSSR*, vol. 114, no. 5, pp. 953–956, 1957.

[11] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989. [Online]. Available: http://dx.doi.org/10.1007/BF02551274

[12] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4, no. 2, pp. 251 – 257, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/089360809190009T

[13] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen, “Optimal approximation with sparsely connected deep neural networks,” *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 1, pp. 8–45, 2019.

[14] D. L. Donoho, “Unconditional bases are optimal bases for data compression and for statistical estimation,” *Appl. Comput. Harmon. Anal.*, vol. 1, no. 1, pp. 100 – 115, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1063520383710080

[15] ——, “Unconditional bases and bit-level compression,” *Appl. Comput. Harm. Anal.*, vol. 3, pp. 388–392, 1996.

[16] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed. USA: Academic Press, Inc., 2008.

[17] K. Gröchenig, *Foundations of time-frequency analysis*. Springer Science & Business Media, 2013.

[18] L. Demanet and L. Ying, “Wave atoms and sparsity of oscillatory patterns,” *Appl. Comput. Harmon. Anal.*, vol. 23, no. 3, pp. 368–387, 2007.

[19] C. L. Fefferman, “Reconstructing a neural net from its output,” *Revista Matemática Iberoamericana*, vol. 10, no. 3, pp. 507–555, 1994.

[20] D. M. Elbrächter, J. Berner, and P. Grohs, “How degenerate is the parametrization of neural networks with the ReLU activation function?” in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, p. 7788–7799. [Online]. Available: https://arxiv.org/abs/1905.09803

[21] V. Vlačić and H. Bölcskei, “Neural network identifiability for a family of sigmoidal nonlinearities,” *Constructive Approximation*, 2021. [Online]. Available: https://arxiv.org/abs/1906.06994

[22] ——, “Affine symmetries and neural network identifiability,” *Advances in Mathematics*, vol. 376, no. 107485, pp. 1–72, 2021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001870820305132

[23] P. Petersen and F. Voigtlaender, “Optimal approximation of piecewise smooth functions using deep ReLU neural networks,” *Neural Networks*, vol. 108, pp. 296–330, Sep. 2018.

[24] D. Yarotsky, “Error bounds for approximations with deep ReLU networks,” *Neural Networks*, vol. 94, pp. 103–114, 2017.

[25] J. Schmidt-Hieber, “Nonparametric regression using deep neural networks with ReLU activation function,” *Annals of Statistics*, vol. 48, no. 4, pp. 1875–1897, 2020. [Online]. Available: https://arxiv.org/abs/1708.06633

[26] M. Telgarsky, “Representation benefits of deep feedforward networks,” *arXiv:1509.08101*, 2015.

[27] B. Hanin and D. Rolnick, “Deep ReLU networks have surprisingly few activation patterns,” in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 361–370. [Online]. Available: http://papers.nips.cc/paper/8328-deep-relu-networks-have-surprisingly-few-activation-patterns.pdf

[28] D. Fokina and I. Oseledets, “Growing axons: Greedy learning of neural networks with application to function approximation,” 2019. [Online]. Available: https://arxiv.org/abs/1910.12686

[29] C. Schwab and J. Zech, “Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ,” *Analysis and Applications*, vol. 17, no. 1, pp. 19–55, 2019.

[30] J. A. A. Opschoor, P. C. Petersen, and C. Schwab, “Deep ReLU networks and high-order finite element methods,” *Analysis and Applications*, vol. 18, no. 5, pp. 715–770, 2020. [Online]. Available: https://doi.org/10.1142/S0219530519410136

[31] I. Gühring, G. Kutyniok, and P. Petersen, “Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms,” *Analysis and Applications*, vol. 18, no. 5, pp. 803–859, 2020. [Online]. Available: https://doi.org/10.1142/S0219530519410021

[32] M. H. Stone, “The generalized Weierstrass approximation theorem,” *Mathematics Magazine*, vol. 21, pp. 167–184, 1948.

[33] S. Liang and R. Srikant, “Why deep neural networks for function approximation?” *International Conference on Learning Representations*, 2017. [Online]. Available: https://arxiv.org/abs/1610.04161

[34] A. Gil, J. Segura, and N. M. Temme, *Numerical Methods for Special Functions*. Society for Industrial and Applied Mathematics, 2007. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9780898717822

[35] A. R. Barron, “Universal approximation bounds for superpositions of a sigmoidal function,” *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 930–945, 1993.

[36] ——, “Approximation and estimation bounds for artificial neural networks,” *Mach. Learn.*, vol. 14, no. 1, pp. 115–133, 1994. [Online]. Available: http://dx.doi.org/10.1007/BF00993164

[37] C. K. Chui, X. Li, and H. N. Mhaskar, “Neural networks for localized approximation,” *Math. Comp.*, vol. 63, no. 208, pp. 607–623, 1994. [Online]. Available: http://dx.doi.org/10.2307/2153285

[38] R. DeVore, K. Oskolkov, and P. Petrushev, “Approximation by feed-forward neural networks,” *Ann. Numer. Math.*, vol. 4, pp. 261–287, 1996.

[39] E. J. Candès, “Ridgelets: Theory and applications,” Ph.D. dissertation, Stanford University, 1998.

[40] H. N. Mhaskar, “Neural networks for optimal approximation of smooth and analytic functions,” *Neural Comput.*, vol. 8, no. 1, pp. 164–177, 1996.

[41] H. N. Mhaskar and C. A. Micchelli, “Degree of approximation by neural and translation networks with a single hidden layer,” *Adv. Appl. Math.*, vol. 16, no. 2, pp. 151–183, 1995.

[42] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[43] H. N. Mhaskar, “Approximation properties of a multilayered feedforward artificial neural network,” *Advances in Computational Mathematics*, vol. 1, no. 1, pp. 61–80, Feb 1993. [Online]. Available: https://doi.org/10.1007/BF02070821

[44] K.-I. Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989. [Online]. Available: //www.sciencedirect.com/science/article/pii/0893608089900038

[45] T. Nguyen-Thien and T. Tran-Cong, “Approximation of functions and their derivatives: A neural network implementation with applications,” *Appl. Math. Model.*, vol. 23, no. 9, pp. 687–704, 1999. [Online]. Available: //www.sciencedirect.com/science/article/pii/S0307904X99000062

[46] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," in *Proceedings of the 29th Conference on Learning Theory*, 2016, pp. 907–940.

[47] H. N. Mhaskar and T. Poggio, "Deep vs. shallow networks: An approximation theory perspective," *Analysis and Applications*, vol. 14, no. 6, pp. 829–848, 2016. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0219530516400042

[48] N. Cohen, O. Sharir, and A. Shashua, "On the expressive power of deep learning: A tensor analysis," in *Proceedings of the 29th Conference on Learning Theory*, vol. 49, 2016, pp. 698–728.

[49] N. Cohen and A. Shashua, "Convolutional rectifier networks as generalized tensor decompositions," in *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, 2016, pp. 955–963.

[50] 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 e-prints*, p. arXiv:1809.02362, Sep. 2018.

[51] 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 Journal on Mathematics of Data Science*, vol. 2, no. 3, pp. 631–657, 2020.

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

[53] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab, "DNN expression rate analysis of high-dimensional PDEs: Application to option pricing," *arXiv:1809.07669*, 2018.

[54] S. Ellacott, "Aspects of the numerical analysis of neural networks," *Acta Numer.*, vol. 3, pp. 145–202, 1994.

[55] A. Pinkus, "Approximation theory of the MLP model in neural networks," *Acta Numer.*, vol. 8, pp. 143–195, 1999.

[56] R. DeVore, B. Hanin, and G. Petrova, "Neural network approximation," *arXiv:2012.14501*, 2020.

[57] U. Shaham, A. Cloninger, and R. R. Coifman, "Provable approximation properties for deep neural networks," *Appl. Comput. Harmon. Anal.*, vol. 44, no. 3, pp. 537–557, May 2018. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1509.html#ShahamCC15

[58] R. A. DeVore and G. G. Lorentz, *Constructive Approximation*. Springer, 1993.

[59] R. A. DeVore, "Nonlinear approximation," *Acta Numerica*, vol. 7, pp. 51–150, 1998.

[60] P. Grohs, "Optimally sparse data representations," in *Harmonic and Applied Analysis*. Springer, 2015, pp. 199–248.

[61] E. Ott, *Chaos in Dynamical Systems*. Cambridge University Press, 2002.

[62] M. Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019.

[63] R. T. Prosser, "The ε-entropy and ε-capacity of certain time-varying channels," *Journal of Mathematical Analysis and Applications*, vol. 16, pp. 553–573, 1966.

[64] A. Kolmogorov and V. Tikhomirov, "ε-entropy and ε-capacity of sets in function spaces," *Uspekhi Mat. Nauk.*, vol. 14, no. 2, pp. 3–86, 1959.

[65] M. Ehler and F. Filbir, "Metric entropy, n-widths, and sampling of functions on manifolds," *Journal of Approximation Theory*, vol. 225, pp. 41 – 57, 2018.

[66] J. Schmidt-Hieber, "Deep ReLU network approximation of functions on a manifold," *arXiv:1908.00695*, 2019.

[67] H. Mhaskar, "A direct approach for function approximation on data defined manifolds," *Neural Networks*, vol. 132, pp. 253 – 268, 2020.

[68] V. Morgenshtern and H. Bölcskei, *Mathematical Foundations for Signal Processing, Communications, and Networking*, Boca Raton, FL, 2012, ch. A short course on frame theory, pp. 737–789.

[69] P. Grohs, S. Keiper, G. Kutyniok, and M. Schäfer, "α-molecules," *Appl. Comput. Harmon. Anal.*, vol. 41, no. 1, pp. 297–336, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.acha.2015.10.009

[70] I. Daubechies, *Ten Lectures on Wavelets*. SIAM, 1992.

[71] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C2 singularities," *Comm. Pure Appl. Math.*, vol. 57, pp. 219–266, 2002.

[72] K. Guo, G. Kutyniok, and D. Labate, "Sparse multidimensional representations using anisotropic dilation and shear operators," in *Wavelets and Splines (Athens, GA, 2005)*. Nashboro Press, Nashville, TN, 2006, pp. 189–201.

[73] P. Grohs and G. Kutyniok, "Parabolic molecules," *Found. Comput. Math.*, vol. 14, pp. 299–337, 2014.

[74] K. Gröchenig and S. Samarah, “Nonlinear approximation with local Fourier bases,” *Constructive Approximation*, vol. 16, no. 3, pp. 317–331, Jul. 2000.

[75] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, “Data compression and harmonic analysis,” *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2435–2476, 1998.

[76] P. Grohs, A. Klotz, and F. Voigtlaender, “Phase transitions in rate distortion theory and deep learning,” *arxiv:2008.01011*, 2020.

[77] A. Hinrichs, I. Piotrowska-Kurczewski, and M. Piotrowski, “On the degree of compactness of embeddings between weighted modulation spaces,” *J. Funct. Spaces Appl.*, vol. 6, pp. 303–317, 01 2008.

[78] P. Grohs, S. Keiper, G. Kutyniok, and M. Schäfer, “Cartoon approximation with α-curvelets,” *J. Fourier Anal. Appl.*, vol. 22, no. 6, pp. 1235–1293, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00041-015-9446-6

[79] D. L. Donoho, “Sparse components of images and optimal atomic decompositions,” *Constr. Approx.*, vol. 17, no. 3, pp. 353–382, 2001. [Online]. Available: http://dx.doi.org/10.1007/s003650010032

[80] J. Munkres, *Topology*, ser. Featured Titles for Topology. Prentice Hall, Incorporated, 2000.

[81] M. Unser, “Ten good reasons for using spline wavelets,” *Wavelet Applications in Signal and Image Processing V*, vol. 3169, pp. 422–431, 1997.

[82] S. Mallat, “Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$,” *Trans. Amer. Math. Soc.*, vol. 315, no. 1, pp. 69–87, Sep. 1989.

[83] C. K. Chui and J.-Z. Wang, “On compactly supported spline wavelets and a duality principle,” *Trans. Amer. Math. Soc.*, vol. 330, no. 2, pp. 903–915, Apr. 1992.

[84] C. L. Fefferman, “The uncertainty principle,” *Bull. Amer. Math. Soc. (N.S.)*, vol. 9, no. 2, pp. 129–206, 1983. [Online]. Available: https://doi.org/10.1090/S0273-0979-1983-15154-6

[85] H. G. Feichtinger, “On a new Segal algebra,” *Monatshefte für Mathematik*, vol. 92, pp. 269–289, 1981.

[86] A. Zygmund, *Trigonometric series*. Cambridge University Press, 2002.

[87] C. Frenzen, T. Sasao, and J. T. Butler, “On the number of segments needed in a piecewise linear approximation,” *Journal of Computational and Applied Mathematics*, vol. 234, no. 2, pp. 437–446, 2010.

III. Towards a regularity theory for ReLU networks – chain rule and global error estimates

Authors: Julius Berner, Dennis Elbrächter, Philipp Grohs, and Arnulf Jentzen

Status: Published in the proceedings of the 13th International conference on Sampling Theory and Applications (SampTA).

Arxiv: https://arxiv.org/abs/1905.04992

Contribution: Conceived, developed, and written in equal parts by Julius Berner and Dennis Elbrächter. Philipp Grohs and Arnulf Jentzen contributed in an advisory capacity.

Towards a regularity theory for ReLU networks – chain rule and global error estimates

Julius Berner*, Dennis Elbrächter*, Philipp Grohs‡, Arnulf Jentzen§
*Faculty of Mathematics, University of Vienna
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
‡Faculty of Mathematics and Research Platform DataScience@UniVienna, University of Vienna
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
§Department of Mathematics, ETH Zürich
Rämistrasse 101, 8092 Zürich, Switzerland

Abstract—**Although for neural networks with locally Lipschitz continuous activation functions the classical derivative exists almost everywhere, the standard chain rule is in general not applicable. We will consider a way of introducing a derivative for neural networks that admits a chain rule, which is both rigorous and easy to work with. In addition we will present a method of converting approximation results on bounded domains to global (pointwise) estimates. This can be used to extend known neural network approximation theory to include the study of regularity properties. Of particular interest is the application to neural networks with ReLU activation function, where it contributes to the understanding of the success of deep learning methods for high-dimensional partial differential equations.**

I. Introduction

It has been observed that deep neural networks exhibit the remarkable capability of overcoming the curse of dimensionality in a number of different scenarios. In particular, for certain types of high-dimensional partial differential equations (PDEs) there are promising empirical observations [1], [2], [3], [4], [5], [6], [7] backed by theoretical results for both the approximation error [8], [9], [10], [11] as well as the generalization error [12]. In this context it becomes relevant to not only show how well a given function of interest can be approximated by neural networks but also to extend the study to the derivative of this function. A number of recent publications [13], [14], [15] have investigated the required size of a network which is sufficient to approximate certain interesting (classes of) functions within a given accuracy. This is achieved, first, by considering the approximation of basic functions by very simple networks and, subsequently, by combining those networks in order to approximate more difficult structures. To extend this approach to include the regularity of the approximation, one requires some kind of chain rule for the composition of neural networks. For neural networks with differentiable activation function the standard chain rule is sufficient. It, however, fails when considering neural networks with an activation function, which is not everywhere differentiable. Although locally Lipschitz continuous functions are w.r.t the Lebesgue measure almost everywhere (a.e.) differentiable, the standard chain rule is not applicable, as, in general, it does not hold even in an 'almost everywhere' sense. We will introduce derivatives of neural networks in a way that admits a chain rule which is both rigorous as well as easy to work with. Chain rules for functions which are not everywhere differentiable have been considered in a more general setting in e.g. [16], [17]. We employ the specific structure of neural networks to get stronger results using simpler arguments. In particular it allows for a stability result, i.e. Lemma III.3, the application of which will be discussed in Section V. We would also like to mention a very recent work [18] about approximation in Sobolev norms, where they deal with the issue by using a general bound for the Sobolev norm of the composition of functions from the Sobolev space $W^{1,\infty}$. Note however that this approach leads to a certain factor depending on the dimensions of the domains of the functions, which can be avoided with our method. For ease of exposition, we formulate our results for neural networks with the ReLU activation function. We, however, consider in Section IV how such a chain rule can be obtained for any activation function which is locally Lipschitz continuous (with at most countably many points at which it is not differentiable). In Section V we briefly sketch how the results from Section III can be utilized to get approximation results for certain classes of functions. Subsequently, in Section VI, we present a general method of deriving global error estimates from such approximation results, which are naturally obtained for bounded domains. Ultimately, we discuss how our results can be used to extend known theory, enabling the further study of the approximation of PDE solutions by neural networks.

II. Setting

As in [14], we consider a neural network Φ to be a finite sequence of matrix-vector pairs, i.e.

$$\Phi = ((A_k, b_k))_{k=1}^{L}, \tag{1}$$

where $A_k \in \mathbb{R}^{N_k \times N_{k-1}}$ and $b_k \in \mathbb{R}^{N_k}$ for some depth $L \in \mathbb{N}$ and layer dimensions $N_0, N_1, \dots, N_L \in \mathbb{N}$. The realization of the neural network Φ is the function $\mathcal{R}\Phi\colon \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ given by

$$\mathcal{R}\Phi = W_L \circ \mathrm{ReLU} \circ W_{L-1} \circ \ldots \circ \mathrm{ReLU} \circ W_1, \tag{2}$$

where $W_k(x) = A_k x + b_k$ for every $x \in \mathbb{R}^{N_k}$ and where

$$\mathrm{ReLU}(x) := (\max\{0, x_1\}, \dots, \max\{0, x_N\}) \tag{3}$$

for every $x \in \mathbb{R}^N$. We distinguish between a neural network and its realization, since Φ uniquely induces $\mathcal{R}\Phi$, while in general there can be multiple non-trivially different neural networks with the same realization. The representation of a neural network as a structured set of weights as in (1) allows the introduction of notions of network sizes. While there are slight differences between various publications, commonly considered quantities are the depth (i.e. number of affine transformations), the connectivity (i.e. number of non-zero entries of the A_k and b_k), and the weight bound (i.e. maximum of the absolute values of the entries of the A_k and b_k). In [15] it has been shown that these three quantities determine the length of a bit string which is sufficient to encode the network with a prescribed quantization error. In the following let

$$\Phi = ((A_k, b_k))_{k=1}^{L}, \quad \Psi = ((\tilde{A}_k, \tilde{b}_k))_{k=1}^{\tilde{L}} \tag{4}$$

be neural networks with matching dimensions in the sense that $\mathcal{R}\Phi\colon \mathbb{R}^d \to \mathbb{R}^m$ and $\mathcal{R}\Psi\colon \mathbb{R}^m \to \mathbb{R}^n$. We then define their composition as

$$\begin{aligned}&\Psi \odot \Phi := \\ &\big(((A_k, b_k))_{k=1}^{L-1}, (\tilde{A}_1 A_L, \tilde{A}_1 b_L + \tilde{b}_1), ((\tilde{A}_k, \tilde{b}_k))_{k=2}^{\tilde{L}}\big).\end{aligned} \tag{5}$$

Direct computation shows

$$\mathcal{R}(\Psi \odot \Phi) = \mathcal{R}\Psi \circ \mathcal{R}\Phi. \tag{6}$$

Note that the realization $\mathcal{R}\Phi$ of a neural network Φ is continuous piecewise linear (CPL) as a composition of CPL functions. Consequently, it is Lipschitz continuous and the realization $\mathcal{R}\Phi$ is almost everywhere differentiable by Rademacher's theorem. In particular all three functions in (6) are a.e. differentiable. This, however, is not sufficient to get the derivative of $\mathcal{R}(\Psi \odot \Phi)$ from the derivatives of $\mathcal{R}\Psi$ and $\mathcal{R}\Phi$ by use of the classical chain rule. Consider the very simple counterexample of $u(x) := \mathrm{ReLU}(x)$ and $v(x) := 0$ and formally apply the chain rule, i.e.

$$(D(u \circ v))(x) = (Du)(v(x)) \cdot (Dv)(x). \tag{7}$$

Even though $(Du)(y)$ is well-defined for every $y \in \mathbb{R}\backslash\{0\}$, the expression $(Du)(v(x))$ is defined for no $x \in \mathbb{R}$. In general this problem occurs when the inner function maps a set of positive measure into a set where the derivative of the outer function does not exist. Now in this case, one can directly see that setting $(Du)(0)$ to any arbitrary value would cause (7) to provide the correct result since $(Dv)(x) = 0$.

III. ReLU network derivative

We proceed by defining the derivative of an arbitrary neural network in a way such that it not only coincides a.e. with the derivative of the realization, but also admits a chain rule. To this end let $H\colon \mathbb{R}^N \to \mathbb{R}^{N\times N}$ be the function given by

$$H(x) := \mathrm{diag}(\mathbb{1}_{(0,\infty)}(x_1), \dots, \mathbb{1}_{(0,\infty)}(x_N)) \tag{8}$$

for every $x = (x_1, \dots, x_N) \in \mathbb{R}^N$ and let $\mathcal{R}_K\Phi := \mathcal{R}((A_k, b_k))_{k=1}^{K}$. We then define the neural network derivative of Φ as the function $\mathcal{D}\Phi\colon \mathbb{R}^{N_0} \to \mathbb{R}^{N_L \times N_0}$ given by

$$\mathcal{D}\Phi := A_L \cdot H(\mathcal{R}_{L-1}\Phi) \cdot A_{L-1} \cdot \ldots \cdot H(\mathcal{R}_1\Phi) \cdot A_1. \tag{9}$$

Note that this definition is motivated by formally applying the chain rule with the convention that the derivative of $\max\{0, \cdot\}$ is zero at the origin. Now we need to verify that this is justified.

Theorem III.1. *It holds for almost every $x \in \mathbb{R}^d$ that*

$$(\mathcal{D}\Phi)(x) = (D(\mathcal{R}\Phi))(x). \tag{10}$$

Proof. Let $v\colon \mathbb{R}^d \to \mathbb{R}^N$ be a locally Lipschitz continuous function, define $w := \mathrm{ReLU} \circ v$, and

$$L_i := \{x \in \mathbb{R}^d \colon w_i(x) = 0\} = \{x \in \mathbb{R}^d \colon v_i(x) \leq 0\}. \tag{11}$$

We now use an observation about differentiability on level sets (see e.g. [19, Thm 3.3(i)]), which states that

$$(Dw_i)(x) = 0 \quad \text{for almost every } x \in L_i. \tag{12}$$

As $w_i(x) = v_i(x)$ for every $x \in \mathbb{R}^d \backslash L_i$, we get a.e.

$$Dw_i = \mathbb{1}_{\mathbb{R}^d\backslash L_i} \cdot Dv_i = \mathbb{1}_{(0,\infty)}(v_i) \cdot Dv_i \tag{13}$$

and consequently

$$D(\mathrm{ReLU} \circ v) = H(v) \cdot Dv. \tag{14}$$

The claim follows by induction over the layers $K = 1, \dots, L$ of Φ, using (14) with $v = \mathcal{R}_K\Phi$ for the induction step. □

Note that even for convex $\mathcal{R}\Phi$ the values of $\mathcal{D}\Phi$ on the nullset do not necessarily lie in the respective subdifferentials of $\mathcal{R}\Phi$, as can be seen in Figure 1. Although Theorem III.1 holds regardless of which value is chosen for the derivative of $\max\{0, \cdot\}$ at the origin, no choice will guarantee that all values of $\mathcal{D}\Phi$ lie in the respective subdifferentials of $\mathcal{R}\Phi$. Here we have set the derivative at the origin to zero, following the convention of software implementations for deep learning applications, e.g. TensorFlow and PyTorch. Using (5) and (9) one can verify by direct computation that $\mathcal{D}$ obeys the chain rule.

Corollary III.2. *It holds for every $x \in \mathbb{R}^d$ that*

$$(\mathcal{D}(\Psi \odot \Phi))(x) = (\mathcal{D}\Psi)(\mathcal{R}\Phi(x)) \cdot (\mathcal{D}\Phi)(x). \tag{15}$$

Note that (15) is well-defined as $\mathcal{D}\Psi$ exists everywhere, although it only coincides with $D(\mathcal{R}\Psi)$ almost everywhere. Theorem III.1 however guarantees that we still have a.e.

$$\mathcal{D}(\Psi \odot \Phi) = D(\mathcal{R}(\Psi \odot \Phi)) = D(\mathcal{R}\Psi \circ \mathcal{R}\Phi). \tag{16}$$

Next we provide a technical result dealing with the stability of our chain rule, which will prove to be useful in Section V.

Lemma III.3. *It holds for almost every $x \in \mathbb{R}^d$ that*

$$\lim_{y \to \mathcal{R}\Phi(x)} \big[(\mathcal{D}\Psi)(y) - (\mathcal{D}\Psi)(\mathcal{R}\Phi(x))\big] \cdot (\mathcal{D}\Phi)(x) = 0. \tag{17}$$

Proof. We first show for every locally Lipschitz continuous function $u\colon \mathbb{R}^m \to \mathbb{R}^N$ and for almost every $x \in \mathbb{R}^d$ that

$$\lim_{y \to \mathcal{R}\Phi(x)} [H(u(y)) - H(u(\mathcal{R}\Phi(x)))] \cdot (D(u \circ \mathcal{R}\Phi))(x) = 0. \tag{18}$$

If $u_i(\mathcal{R}\Phi(x)) \neq 0$ we have

$$\lim_{y \to \mathcal{R}\Phi(x)} \mathbb{1}_{(0,\infty)}(u_i(y)) = \mathbb{1}_{(0,\infty)}(u_i(\mathcal{R}\Phi(x))) \tag{19}$$

as u_i is continuous and $\mathbb{1}_{(0,\infty)}$ is continuous on $\mathbb{R}\backslash\{0\}$. Furthermore, [19, Thm 3.3(i)] implies that

$$(D(u_i \circ \mathcal{R}\Phi))(x) = 0 \tag{20}$$

for almost every $x \in \mathbb{R}^d$ with $u_i(\mathcal{R}\Phi(x)) = 0$. Since a finite union of nullsets is again a nullset, this proves the claim (18). The lemma follows by induction over the layers $K = 1, \dots, \tilde{L}$ of Ψ and applying (18) with $u = \mathcal{R}_K\Psi$. □

IV. General Activation Functions

As mentioned in the introduction, it is possible to replace the ReLU activation function in (2) by some locally Lipschitz continuous, component-wise applied function $\varrho\colon \mathbb{R} \to \mathbb{R}$ with an at most countably large set S of points where ϱ is not differentiable. Specifically, one can define the neural network derivative (with activation function ϱ) as in (9) with $\mathbb{1}_{(0,\infty)}(x_i)$ in (8) replaced by

$$(\bar{D}\varrho)(x_i) := \begin{cases} 0, & x_i \in S \\ (D\varrho)(x_i), & \text{else} \end{cases}. \tag{21}$$

The chain rule can, again, be checked by direct computation and it is straightforward to adapt Theorem III.1 to this more general setting by considering the level sets

$$\{x \in \mathbb{R}^d \colon w_i(x) = s\}, \quad s \in S. \tag{22}$$

If additionally $\bar{D}\varrho$ is continuous on $\mathbb{R} \setminus S$, the proof of Lemma III.3 translates without any modifications.

V. Utilization in Approximation Theory

These results can now be employed to bound the L^∞-norm of $\mathcal{D}(\Psi \circ \Phi) - D(u \circ v)$, given corresponding estimates for the approximation of u and v by Ψ and Φ, respectively. Here, one has to take some care when bounding the term

$$\|[\mathcal{D}\Psi \circ \mathcal{R}\Phi - Du \circ \mathcal{R}\Phi]\, \mathcal{D}\Phi\|_{L^\infty} \tag{23}$$

by

$$\|\mathcal{D}\Psi - Du\|_{L^\infty} \|\mathcal{D}\Phi\|_{L^\infty}. \tag{24}$$

Again it can happen that $\mathcal{R}\Phi$ maps a set of positive measure into a nullset where the estimate for the approximation of Du by $\mathcal{D}\Psi$ in the *essential* supremum norm is not valid. However, using the stability result in Lemma III.3 one can for almost every $x \in \mathbb{R}^d$ shift to a sufficiently close point $y \approx \mathcal{R}\Phi(x)$ where the estimate holds. In [13] Yarotsky explicitly constructs networks whose realization is a linear interpolation[1] of the squaring function (see Fig. 1 for illustration), which directly gives an estimate on the approximation rate for the derivatives. These simple networks can then be combined to get networks approximating multiplication, polynomials and eventually, by means of e.g. local Taylor approximation, functions f whose first $n \geq 1$ (weak) derivatives are bounded. This leads to estimates of the form

$$\|f - \mathcal{R}\Phi_{\varepsilon,B}\|_{L^\infty(I_B)} \leq \varepsilon, \tag{25}$$

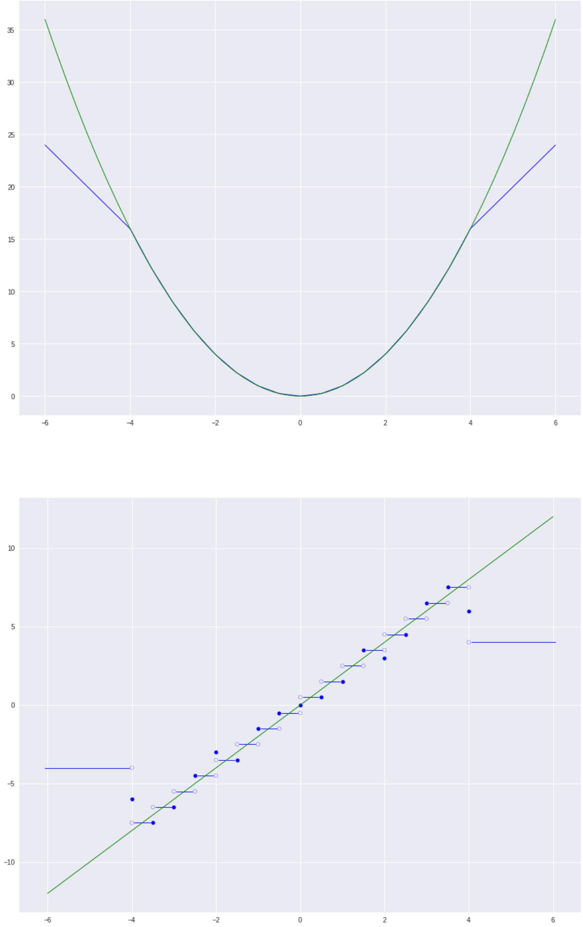


Fig. 1. Approximation of the function $x \mapsto x^2$ and its derivative on the interval $[-4, 4]$ by a neural network Φ with depth 6, connectivity 52 and weight bound 4. Note that not all values of $\mathcal{D}\Phi$ at the points of non-differentiablity of $\mathcal{R}\Phi$ lie between the values at either side, i.e. in the subdifferential.

with $I_B = [-B, B]^d$, including estimates for the scaling of the size of the network $\Phi_{\varepsilon,B}$ w.r.t. B and ε. As these constructions are based on composing simpler functions with known estimates one can now employ Theorem III.1 and Corollary III.2 to show that the derivatives of those networks also approximate the derivative of the function, i.e.

$$\|Df - \mathcal{D}\Phi_{\varepsilon,B}\|_{L^\infty(I_B)} \leq c\varepsilon^r. \tag{26}$$

Such constructive approaches can further be found in [8], in [14] for β-cartoon-like functions, in [20] for $(\boldsymbol{b}, \varepsilon)$-holomorphic maps, and in [15] for high-frequent sinusoidal functions.

VI. Global Error Estimates

The error estimates above are usually only sensible for bounded domains, as the realization of a neural network is always CPL with a finite number of pieces. We briefly discuss a general way of transforming them into global pointwise error estimates, which can be useful in the context of PDEs (see e.g.

[1] The interpolation points are uniformly distributed over the domain of approximation and their number grows exponentially with the size of the networks.

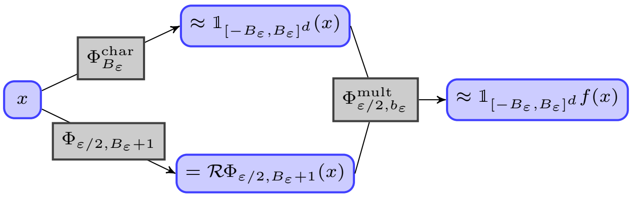

x
$\Phi^{\text{char}}_{B_\varepsilon}$
$\approx \mathbb{1}_{[-B_\varepsilon,B_\varepsilon]^d}(x)$
$\Phi_{\varepsilon/2,B_\varepsilon+1}$
$= \mathcal{R}\Phi_{\varepsilon/2,B_\varepsilon+1}(x)$
$\Phi^{\text{mult}}_{\varepsilon/2,b_\varepsilon}$
$\approx \mathbb{1}_{[-B_\varepsilon,B_\varepsilon]^d} f(x)$

Fig. 2. The neural networks Φ_ε approximating f globally.

[9], [10]). In the following assume that we have a function f with an at most polynomially growing derivative, i.e.

$$\|(Df)(x)\|_2 \le \boldsymbol{c}(1 + \|x\|_2^\kappa). \tag{27}$$

Denote by Φ^{char}_B a neural network which represents the d-dimensional approximate characteristic function of I_B, i.e. $\mathcal{R}\Phi^{\text{char}}_B(x) \in [0, 1]$ and

$$\begin{aligned} \mathcal{R}\Phi^{\text{char}}_B(x) &= 1, \quad x \in I_B, \\ \mathcal{R}\Phi^{\text{char}}_B(x) &= 0, \quad x \notin I_{B+1}. \end{aligned} \tag{28}$$

See [15, Proof of Thm. VIII.3] for such a construction. Further let $\Phi^{\text{mult}}_{\varepsilon,b}$ be the neural network approximating the multiplication function on $[-b, b]^2$ with error ε (see e.g. [20, Prop. 3.1]).

Now we define the global approximation networks Φ_ε as the composition of $\Phi^{\text{mult}}_{\varepsilon/2,b_\varepsilon}$ with the parallelization of $\Phi^{\text{char}}_{B_\varepsilon}$ and $\Phi_{\varepsilon/2,B_\varepsilon+1}$ for suitable

$$B_\varepsilon \in \mathcal{O}(\varepsilon^{-1}) \quad \text{and} \quad b_\varepsilon \in \mathcal{O}(\varepsilon^{-\kappa-1}). \tag{29}$$

See Figure 2 for an illustration and e.g. [14, Def. 2.7] for a formal definition of parallelization. Considering the errors on I_B, $I_{B+1} \backslash I_B$ and $\mathbb{R}^d \backslash I_{B+1}$ leads to global estimates, i.e. for every $x \in \mathbb{R}^d$

$$|f(x) - \mathcal{R}\Phi_\varepsilon(x)| \le \varepsilon(1 + \|x\|_2^{\kappa+2}) \tag{30}$$

and, by use of the chain rule III.2, for almost every $x \in \mathbb{R}^d$

$$\|(Df)(x) - (\mathcal{D}\Phi_\varepsilon)(x)\|_2 \le C\varepsilon^r(1 + \|x\|_2^{\kappa+2}). \tag{31}$$

Due to the logarithmic size scaling of the multiplication network, the size of Φ_ε can be bounded by the size of $\Phi_{\varepsilon/2,B_\varepsilon+1}$ plus an additional term in $\mathcal{O}(d + \kappa \log \varepsilon^{-1})$.

VII. Application to PDEs

Analyzing the regularity properties of neural networks was motivated by the recent successful application of deep learning methods to PDEs [2], [3], [4], [5], [6], [7], [11]. Initiated by empirical experiments [1] it has been proven that neural networks are capable of overcoming the curse of dimensionality for solving so-called Kolmogorov PDEs [12]. More precisely, the solution to the empirical risk minimization problem over a class of neural networks approximates the solution of the PDE up to error ε with high probability and with size of the networks and number of samples scaling only polynomially in the dimension d and ε^{-1}. The above requires a suitable learning problem and a sufficiently good approximation of the solution function by neural networks. For Kolmogorov PDEs, this boils down to calculating global Lipschitz coefficients and error estimates for neural networks approximating the initial condition and coefficient functions (see e.g. [9], [10]). Employing estimates of the form (26) one can bound the derivative on I_B, i.e.

$$L_B := \|\mathcal{D}\Phi_{\varepsilon,B}\|_{L^\infty(I_B)} \le \|Df\|_{L^\infty(I_B)} + c\varepsilon^r. \tag{32}$$

Using mollification and the mean value theorem we can establish local Lipschitz estimates, i.e. for all $x, y \in (-B, B)^d$ that

$$|\mathcal{R}\Phi_{\varepsilon,B}(x) - \mathcal{R}\Phi_{\varepsilon,B}(y)| \le L_B\|x - y\|_2, \tag{33}$$

and corresponding linear growth bounds

$$|\mathcal{R}\Phi_{\varepsilon,B}(x)| \le \big(|\mathcal{R}\Phi_{\varepsilon,B}(0)| + L_B\big)(1 + \|x\|_2). \tag{34}$$

Similarly, one can use (31) to obtain estimates of the form

$$|\mathcal{R}\Phi_\varepsilon(x) - \mathcal{R}\Phi_\varepsilon(y)| \le \boldsymbol{C}(1 + \|x\|_2^{\kappa+2} + \|y\|_2^{\kappa+2})\|x - y\|_2 \tag{35}$$

for all $x, y \in \mathbb{R}^d$ (which are demanded in [10, Theorem 1.1]). Moreover, note that the capability to produce approximation results which include error estimates for the derivative is of significant independent interest. Various numerical methods (for instance Galerkin methods) rely on bounding the error in some Sobolev norm $\|\cdot\|_{W^{1,p}}$, which requires estimates of the derivative differences. We believe that the possibility to obtain regularity estimates significantly contributes to the mathematical theory of neural networks and allows for further advances in the numerical approximation of high dimensional partial differential equations.

VIII. Relation to Backpropagation in Training

The approach discussed here could further be applied to the training of neural networks by (stochastic) gradient descent. Note, however, that this is a slightly different setting. From the approximation theory perspective we were interested in the derivative of $x \mapsto \mathcal{R}\Phi(x)$, while in training one requires the derivative of $\Phi \mapsto \mathcal{R}\Phi(x^*)$ for some fixed sample x^*. In particular this function is no longer CPL but rather continuous piecewise polynomial. While this would necessitate some technical modifications, we believe that it should be possible to employ the method used here in order to show that the gradient of $\Phi \mapsto \mathcal{R}\Phi(x^*)$ coincides a.e. with what is computed by backpropagation using the convention of setting the derivative of $\max\{0, \cdot\}$ to zero at the origin (as well as similar conventions for e.g. max-pooling).

Acknowledgment

The research of JB and DE was supported by the Austrian Science Fund (FWF) under grants I3403-N32 and P 30148.

References

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

[2] W. E, J. Han, and A. Jentzen, “Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations,” *Communications in Mathematics and Statistics*, vol. 5, no. 4, pp. 349–380, 2017.

[3] J. Han, A. Jentzen, and W. E, “Solving high-dimensional partial differential equations using deep learning,” *arXiv:1707.02568*, 2017.

[4] J. Sirignano and K. Spiliopoulos, “DGM: A deep learning algorithm for solving partial differential equations,” *arXiv:1708.07469*, 2017.

[5] M. Fujii, A. Takahashi, and M. Takahashi, “Asymptotic Expansion as Prior Knowledge in Deep Learning Method for high dimensional BSDEs,” *arXiv:1710.07030*, 2017.

[6] Y. Khoo, J. Lu, and L. Ying, “Solving parametric PDE problems with artificial neural networks,” *arXiv:1707.03351*, 2017.

[7] W. E and B. Yu, “The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems,” *arXiv:1710.00211*, 2017.

[8] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab, “DNN Expression Rate Analysis of high-dimensional PDEs: Application to Option Pricing,” *arXiv:1809.07669*, 2018.

[9] 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:1809.02362*, 2018.

[10] 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:1809.07321*, 2018.

[11] M. Hutzenthaler, A. Jentzen, T. Kruse, and T. A. Nguyen, “A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations,” *arXiv:1707.02568*, 2019.

[12] 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:1809.03062*, 2018.

[13] D. Yarotsky, “Optimal approximation of continuous functions by very deep ReLU networks,” *arXiv:1802.03620*, 2018.

[14] P. Petersen and F. Voigtlaender, “Optimal approximation of piecewise smooth functions using deep ReLU neural networks,” *arXiv:1709.05289*, 2017.

[15] P. Grohs, D. Perekrestenko, D. Elbrächter, and H. Bölcskei, “Deep Neural Network Approximation Theory,” *arxiv:1901.02220*, 2019.

[16] F. Murat and C. Trombetti, “A chain rule formula for the composition of a vector-valued function by a piecewise smooth function,” *Bollettino dell’Unione Matematica Italiana*, vol. 6, no. 3, pp. 581–595, 2003.

[17] L. Ambrosio and G. Dal Maso, “A general chain rule for distributional derivatives,” *Proceedings of the American Mathematical Society*, vol. 108, no. 3, pp. 691–702, 1990.

[18] I. Gühring, G. Kutyniok, and P. Petersen, “Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms,” *arXiv:1902.07896*, 2019.

[19] L. C. Evans and R. F. Gariepy, *Measure Theory and Fine Properties of Functions, Revised Edition*, ser. Textbooks in Mathematics. CRC Press, 2015.

[20] C. Schwab and J. Zech, “Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ,” *Analysis and Applications, Singapore*, vol. 17, no. 1, pp. 19–55, 2019.

IV. How degenerate is the parametrization of neural networks with the ReLU activation function?

Authors: Julius Berner, Dennis Elbrächter, and Philipp Grohs

Status: Published in Advances in Neural Information Processing Systems 32 (NeurIPS 2019)

Arxiv: https://arxiv.org/abs/1905.09803

Contribution: Conceived, developed, and written in equal parts by Julius Berner and Dennis Elbrächter. Philipp Grohs contributed the initial idea and helpful advice.

How degenerate is the parametrization of neural networks with the ReLU activation function?

Julius Berner
Faculty of Mathematics, University of Vienna
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
julius.berner@univie.ac.at

Dennis Elbrächter
Faculty of Mathematics, University of Vienna
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
dennis.elbraechter@univie.ac.at

Philipp Grohs
Faculty of Mathematics and Research Platform DataScience@UniVienna, University of Vienna
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
philipp.grohs@univie.ac.at

Abstract

Neural network training is usually accomplished by solving a non-convex optimization problem using stochastic gradient descent. Although one optimizes over the networks parameters, the main loss function generally only depends on the realization of the neural network, i.e. the function it computes. Studying the optimization problem over the space of realizations opens up new ways to understand neural network training. In particular, usual loss functions like mean squared error and categorical cross entropy are convex on spaces of neural network realizations, which themselves are non-convex. Approximation capabilities of neural networks can be used to deal with the latter non-convexity, which allows us to establish that for sufficiently large networks local minima of a regularized optimization problem on the realization space are almost optimal. Note, however, that each realization has many different, possibly degenerate, parametrizations. In particular, a local minimum in the parametrization space needs not correspond to a local minimum in the realization space. To establish such a connection, inverse stability of the realization map is required, meaning that proximity of realizations must imply proximity of corresponding parametrizations. We present pathologies which prevent inverse stability in general, and, for shallow networks, proceed to establish a restricted space of parametrizations on which we have inverse stability w.r.t. to a Sobolev norm. Furthermore, we show that by optimizing over such restricted sets, it is still possible to learn any function which can be learned by optimization over unrestricted sets.

1 Introduction and Motivation

In recent years much effort has been invested into explaining and understanding the overwhelming success of deep learning based methods. On the theoretical side, impressive approximation capabilities of neural networks have been established [9, 10, 16, 20, 32, 33, 37, 39]. No less important are recent results on the generalization of neural networks, which deal with the question of how

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

well networks, trained on limited samples, perform on unseen data [2, 3, 5–7, 17, 29]. Last but not least, the optimization error, which quantifies how well a neural network can be trained by applying stochastic gradient descent to an optimization problem, has been analyzed in different scenarios [1, 11, 13, 22, 24, 25, 27, 38]. While there are many interesting approaches to the latter question, they tend to require very strong assumptions (e.g. (almost) linearity, convexity, or extreme over-parametrization). Thus a satisfying explanation for the success of stochastic gradient descent for a non-smooth, non-convex problem remains elusive.

In the present paper we intend to pave the way for a functional perspective on the optimization problem. This allows for new mathematical approaches towards understanding the training of neural networks, some of which are demonstrated in Section 1.2. To this end we examine degenerate parametrizations with undesirable properties in Section 2. These can be roughly classified as

C.1 unbalanced magnitudes of the parameters

C.2 weight vectors with the same direction

C.3 weight vectors with directly opposite directions.

Under conditions designed to avoid these degeneracies, Theorem 3.1 establishes inverse stability for shallow networks with ReLU activation function. This is accomplished by a refined analysis of the behavior of ReLU networks near a discontinuity of their derivative. Proposition 1.2 shows how inverse stability connects the loss surface of the parametrized minimization problem to the loss surface of the realization space problem. In Theorem 1.3 we showcase a novel result on almost optimality of local minima of the parametrized problem obtained by analyzing the realization space problem. Note that this approach of analyzing the loss surface is conceptually different from previous approaches as in [11, 18, 23, 30, 31, 36].

1.1 Inverse Stability of Neural Networks

We will focus on neural networks with the ReLU activation function $\rho(x) := x_+$, and adapt the mathematically convenient notation from [33], which distinguishes between the *parametrization* of a neural network and its *realization*. Let us define the set $\mathcal{A}_L$ of all network *architectures* with depth $L \in \mathbb{N}$, input dimension $d \in \mathbb{N}$, and output dimension $D \in \mathbb{N}$ by

$$\mathcal{A}_L := \{(N_0, \dots, N_L) \in \mathbb{N}^{L+1} \colon N_0 = d, N_L = D\}. \tag{1}$$

The architecture $N \in \mathcal{A}_L$ simply specifies the number of neurons N_l in each of the L layers. We can then define the space $\mathcal{P}_N$ of *parametrizations* with architecture $N \in \mathcal{A}_L$ as

$$\mathcal{P}_N := \prod_{\ell=1}^{L} \left(\mathbb{R}^{N_\ell \times N_{\ell-1}} \times \mathbb{R}^{N_\ell}\right), \tag{2}$$

the set $\mathcal{P} = \bigcup_{N \in \mathcal{A}_L} \mathcal{P}_N$ of all parametrizations with architecture in $\mathcal{A}_L$, and the *realization* map

$$\begin{aligned} \mathcal{R} \colon \mathcal{P} &\to C(\mathbb{R}^d, \mathbb{R}^D) \\ \Theta = ((A_\ell, b_\ell))_{\ell=1}^L &\mapsto \mathcal{R}(\Theta) := W_L \circ \rho \circ W_{L-1} \dots \rho \circ W_1, \end{aligned} \tag{3}$$

where $W_\ell(x) := A_\ell x + b_\ell$ and ρ is applied component-wise. We refer to A_ℓ and b_ℓ as the weights and biases in the ℓ-th layer.

Note that a parametrization $\Theta \in \Omega \subseteq \mathcal{P}$ uniquely induces a realization $\mathcal{R}(\Theta)$ in the realization space $\mathcal{R}(\Omega)$, while in general there can be multiple non-trivially different parametrizations with the same realization. To put it in mathematical terms, the realization map is not injective. Consider the basic counterexample

$$\Theta = \big((A_1, b_1), \dots, (A_{L-1}, b_{L-1}), (0, 0)\big) \quad \text{and} \quad \Gamma = \big((B_1, c_1), \dots, (B_{L-1}, c_{L-1}), (0, 0)\big) \tag{4}$$

from [34] where regardless of A_ℓ, B_ℓ, b_ℓ and c_ℓ both realizations coincide with $\mathcal{R}(\Theta) = \mathcal{R}(\Gamma) = 0$. However, it it is well-known that the realization map is locally Lipschitz continuous, meaning that close[1] parametrizations in $\mathcal{P}_N$ induce realizations which are close in the uniform norm on compact

[1] On the finite dimensional vector space $\mathcal{P}_N$ all norms are equivalent and we take w.l.o.g. the maximum norm $\|\Theta\|_\infty$, i.e. the maximum of the absolute values of the entries of the A_ℓ and b_ℓ.

sets, see e.g. [2, Lemma 14.6], [7, Theorem 4.2], and [34, Proposition 5.1].
We will shed light upon the inverse question. Given realizations $\mathcal{R}(\Gamma)$ and $\mathcal{R}(\Theta)$ that are close, do the parametrizations Γ and Θ have to be close? In an abstract setting we measure the proximity of realizations in the norm $\|\cdot\|$ of a Banach space $\mathcal{B}$ with $\mathcal{R}(\mathcal{P}) \subseteq \mathcal{B}$, while concrete Banach spaces of interest will be specified later. In view of the above counterexample we will, at the very least, need to allow for the reparametrization of one of the networks, i.e. we arrive at the following question.

> Given $\mathcal{R}(\Gamma)$ and $\mathcal{R}(\Theta)$ that are close, does there exist a parametrization Φ with $\mathcal{R}(\Phi) = \mathcal{R}(\Theta)$ such that Γ and Φ are close?

As we will see in Section 2, this question is fundamentally connected to understanding the redundancies and degeneracies of the way that neural networks are parametrized. By suitable regularization, i.e. considering a subspace $\Omega \subseteq \mathcal{P}_N$ of parametrizations, we can avoid these pathologies and establish a positive answer to the question above. For such a property the term *inverse stability* was introduced in [34], which constitutes the only other research conducted in this area, as far as we are aware.

Definition 1.1 (Inverse stability)**.** *Let $s, \alpha > 0$, $N \in \mathcal{A}_L$, and $\Omega \subseteq \mathcal{P}_N$. We say that the realization map is (s, α) inverse stable on Ω w.r.t. $\|\cdot\|$, if for all $\Gamma \in \Omega$ and $g \in \mathcal{R}(\Omega)$ there exists $\Phi \in \Omega$ with*

$$\mathcal{R}(\Phi) = g \quad \textit{and} \quad \|\Phi - \Gamma\|_\infty \leq s\|g - \mathcal{R}(\Gamma)\|^\alpha. \tag{5}$$

In Section 2 we will see why inverse stability fails w.r.t. the uniform norm. Therefore, we consider a norm which takes into account not only the maximum error of the function values but also of the gradients. In mathematical terms, we make use of the Sobolev norm $\|\cdot\|_{W^{1,\infty}(U)}$ (on some domain $U \subseteq \mathbb{R}^d$) defined for every (locally) Lipschitz continuous function $g\colon \mathbb{R}^d \to \mathbb{R}^D$ by $\|g\|_{W^{1,\infty}(U)} := \max\{\|g\|_{L^\infty(U)}, |g|_{W^{1,\infty}(U)}\}$ with the Sobolev semi-norm $|\cdot|_{W^{1,\infty}(U)}$ given by

$$|g|_{W^{1,\infty}(U)} := \|Dg\|_{L^\infty(U)} = \operatorname*{ess\,sup}_{x\in U} \|Dg(x)\|_\infty. \tag{6}$$

See [15] for further information on Sobolev norms, and [8] for further information on the derivative of ReLU networks.

1.2 Implications of inverse stability for neural network optimization

We proceed by demonstrating how inverse stability opens up new perspectives on the optimization problem which arises in neural network training. Specifically, consider a loss function $\mathcal{L}\colon C(\mathbb{R}^d, \mathbb{R}^D) \to [0, \infty)$ on the space of continuous functions. For illustration, we take the commonly used mean squared error (MSE) which, for training data $((x^i, y^i))_{i=1}^n \in (\mathbb{R}^d \times \mathbb{R}^D)^n$, is given by

$$\mathcal{L}(g) = \tfrac{1}{n} \sum_{i=1}^n \|g(x^i) - y^i\|_2^2, \quad \text{for } g \in C(\mathbb{R}^d, \mathbb{R}^D). \tag{7}$$

Typically, the optimization problem is solved over some subspace of parametrizations $\Omega \subseteq \mathcal{P}_N$, i.e.

$$\min_{\Gamma\in\Omega} \mathcal{L}(\mathcal{R}(\Gamma)) = \min_{\Gamma\in\Omega} \tfrac{1}{n} \sum_{i=1}^n \|\mathcal{R}(\Gamma)(x^i) - y^i\|_2^2. \tag{8}$$

From an abstract point of view, by writing $g = \mathcal{R}(\Gamma) \in \mathcal{R}(\Omega)$, this is equivalent to the corresponding optimization problem over the space of realizations $\mathcal{R}(\Omega)$, i.e.

$$\min_{g\in\mathcal{R}(\Omega)} \mathcal{L}(g) = \min_{g\in\mathcal{R}(\Omega)} \tfrac{1}{n} \sum_{i=1}^n \|g(x^i) - y^i\|_2^2. \tag{9}$$

However, the loss landscape of the optimization problem (8) is only properly connected to the loss landscape of the optimization problem (9) if the realization map is inverse stable on Ω. Otherwise a realization $g \in \mathcal{R}(\mathcal{P}_N)$ can be arbitrarily close to a global minimum in the realization space but every parametrization Φ with $\mathcal{R}(\Phi) = g$ is far away from the corresponding global minimum in the parametrization space. Moreover, local minima of (8) in the parametrization space must correspond to local minima of (9) in the realization space if and only if we have inverse stability.

Proposition 1.2 (Parametrization minimum ⇒ realization minimum)**.** *Let $N \in \mathcal{A}_L$, $\Omega \subseteq \mathcal{P}_N$ and let the realization map be (s, α) inverse stable on Ω w.r.t. $\|\cdot\|$. Let $\Gamma_* \in \Omega$ be a local minimum of $\mathcal{L} \circ \mathcal{R}$ on Ω with radius $r > 0$, i.e. for all $\Phi \in \Omega$ with $\|\Phi - \Gamma_*\|_\infty \le r$ it holds that*

$$\mathcal{L}(\mathcal{R}(\Gamma_*)) \le \mathcal{L}(\mathcal{R}(\Phi)). \tag{10}$$

Then $\mathcal{R}(\Gamma_*)$ is a local minimum of $\mathcal{L}$ on $\mathcal{R}(\Omega)$ with radius $(\frac{r}{s})^{1/\alpha}$, i.e. for all $g \in \mathcal{R}(\Omega)$ with $\|g - \mathcal{R}(\Gamma_*)\| \le (\frac{r}{s})^{1/\alpha}$ it holds that

$$\mathcal{L}(\mathcal{R}(\Gamma_*)) \le \mathcal{L}(g). \tag{11}$$

See Appendix A.1.2 for a proof and Example A.1 for a counterexample in the case that inverse stability is not given. Note that in (9) we consider a problem with convex loss function but non-convex feasible set, see [34, Section 3.2]. This opens up new avenues of investigation using tools from functional analysis and allows utilizing recent results [19, 34] exploring the topological properties of neural network realization spaces.

As a concrete demonstration we provide with Theorem A.2 a strong result obtained on the realization space, which estimates the quality of a local minimum based on its radius and the approximation capabilities of the chosen architecture for a class of functions S. Specifically let $C > 0$, let $\Lambda \colon \mathcal{B} \to [0, \infty)$ be a quasi-convex regularizer, and define

$$S := \{f \in \mathcal{B} \colon \Lambda(f) \le C\}. \tag{12}$$

We denote the sets of regularized parametrizations by

$$\Omega_N := \{\Phi \in \mathcal{P}_N \colon \Lambda(\mathcal{R}(\Phi)) \le C\} \tag{13}$$

and assume that the loss function $\mathcal{L}$ is convex and c-Lipschitz continuous on S. Note that virtually all relevant loss functions are convex and locally Lipschitz continuous on $C(\mathbb{R}^d, \mathbb{R}^D)$. Employing Proposition 1.2, inverse stability can then be used to derive the following result for the practically relevant parametrized problem, showing that for sufficiently large architectures local minima of a regularized neural network optimization problem are almost optimal.

Theorem 1.3 (Almost optimality of local parameter minima)**.** *Assume that S is compact in the $\|\cdot\|$-closure of $\mathcal{R}(\mathcal{P})$ and that for every $N \in \mathcal{A}_L$ the realization map is (s, α) inverse stable on Ω_N w.r.t. $\|\cdot\|$. Then for all $\varepsilon, r > 0$ there exists $n(\varepsilon, r) \in \mathcal{A}_L$ such that for every $N \in \mathcal{A}_L$ with $N_1 \ge n_1(\varepsilon, r), \dots, N_{L-1} \ge n_{L-1}(\varepsilon, r)$ the following holds:*
Every local minimum Γ_* with radius at least r of $\min_{\Gamma \in \Omega_N} \mathcal{L}(\mathcal{R}(\Gamma))$ satisfies

$$\mathcal{L}(\mathcal{R}(\Gamma_*)) \le \min_{\Gamma \in \Omega_N} \mathcal{L}(\mathcal{R}(\Gamma)) + \varepsilon. \tag{14}$$

See Appendix A.1.2 for a proof and note that here it is important to have an inverse stability result, where the parameters (s, α) do not depend on the size of the architecture, which we achieve for $L = 2$ and $\mathcal{B} = W^{1,\infty}$. Suitable Λ would be Besov norms which constitute a common regularizer in image and signal processing. Moreover, note that the required size of the architecture in Theorem 1.3 can be quantified, if one has approximation rates for S. In particular, this approach allows the use of approximation results in order to explain the success of neural network optimization and enables a combined study of these two aspects, which, to the best of our knowledge, has not been done before. Unlike in recent literature, our result needs no assumptions on the sample set (incorporated in the loss function, see (7)), in particular we do not require "overparametrization" with respect to the sample size. Here the required size of the architecture only depends on the complexity of S, i.e. the class of functions one wants to approximate, the radius of the local minima of interest, the Lipschitz constant of the loss function, and the parameters of the inverse stability.

In the following we restrict ourselves to two-layer ReLU networks without biases, where we present a proof for $(4, 1/2)$ inverse stability w.r.t. the Sobolev semi-norm on a suitably regularized space of parametrizations. Both the regularizations as well as the stronger norm (compared to the uniform norm) will shown to be necessary in Section 2. We now present, in an informal way, a collection of our main results. A short proof making the connection to the formal results can be found in Appendix A.1.2.

Corollary 1.4 (Inverse stability and implications - colloquial)**.** *Suppose we are given data $((x^i, y^i))_{i=1}^n \in (\mathbb{R}^d \times \mathbb{R}^D)^n$ and want to solve a typical minimization problem for ReLU networks with shallow architecture $N = (d, N_1, D)$, i.e.*

$$\min_{\Gamma \in \mathcal{P}_N} \frac{1}{n} \sum_{i=1}^{n} \|\mathcal{R}(\Gamma)(x^i) - y^i\|_2^2. \tag{15}$$

First we augment the architecture to $\tilde{N} = (d+2, N_1+1, D)$, while omitting the biases, and augment the samples to $\tilde{x}^i = (x_1^i, \dots, x_d^i, 1, -1)$. Moreover, we assume that the parametrizations

$$\Phi = \big(([a_1|\dots|a_{N_1+1}]^T, 0), ([c_1|\dots|c_{N_1+1}], 0)\big) \in \Omega \subseteq \mathcal{P}_{\tilde{N}} \tag{16}$$

are regularized such that

C.1 the network is balanced, i.e. $\|a_i\|_\infty = \|c_i\|_\infty$,

C.2 no non-zero weight vectors in the first layer are redundant, i.e. $a_i \not\parallel a_j$,

C.3 the last two coordinates of each weight vector a_i are strictly positive.

Then for the new minimization problem

$$\min_{\Phi \in \Omega} \tfrac{1}{n} \sum_{i=1}^{n} \|\mathcal{R}(\Phi)(\tilde{x}^i) - y^i\|_2^2 \tag{17}$$

the following holds:

1. *If Φ_* is a local minimum of (17) with radius r, then $\mathcal{R}(\Phi_*)$ is a local minimum of $\min_{g\in\mathcal{R}(\Omega)} \frac{1}{n}\sum_{i=1}^n \|g(\tilde{x}^i) - y^i\|_2^2$ with radius at least $\frac{r^2}{16}$ w.r.t. $|\cdot|_{W^{1,\infty}}$.*
2. *The global minimum of (17) is at least as good as the global minimum of (15), i.e.*

$$\min_{\Phi \in \Omega} \tfrac{1}{n} \sum_{i=1}^{n} \|\mathcal{R}(\Phi)(\tilde{x}^i) - y^i\|_2^2 \le \min_{\Gamma \in \mathcal{P}_N} \tfrac{1}{n} \sum_{i=1}^{n} \|\mathcal{R}(\Gamma)(x^i) - y^i\|_2^2. \tag{18}$$

3. *By further regularizing (17) in the sense of Theorem 1.3, we can estimate the quality of its local minima.*

This argument is not limited to the MSE loss function but works for any loss function based on evaluating the realization. The omission of bias weights is standard in neural network optimization literature [11, 13, 22, 24]. While this severely limits the functions that can be realized with a given architecture, it is sufficient to augment the problem by one dimension in order to recover the full range of functions that can be learned [1]. Here we augment by two dimensions, so that the third regularization condition C.3 can be fulfilled without loosing range. Moreover, note that, for simplicity of presentation, the regularization assumptions stated above are stricter than necessary and possible relaxations are discussed in Section 3.

2 Obstacles to inverse stability - degeneracies of ReLU parametrizations

In the remainder of this paper we focus on shallow ReLU networks without biases and define the corresponding space of parametrizations with architecture $N = (d, m, D)$ as $\mathcal{N}_N := \mathbb{R}^{m\times d} \times \mathbb{R}^{D\times m}$. The realization map[2] $\mathcal{R}$ is, for every $\Theta = (A, C) = \big([a_1|\dots|a_m]^T, [c_1|\dots|c_m]\big) \in \mathcal{N}_N$, given by

$$\mathbb{R}^d \ni x \mapsto \mathcal{R}(\Theta)(x) = C\rho(Ax) = \sum_{i=1}^{m} c_i \rho(\langle a_i, x\rangle). \tag{19}$$

Note that each function $x \mapsto c_i\rho(\langle a_i, x\rangle)$ represents a so-called ridge function which is zero on the half-space $\{x \in \mathbb{R}^d \colon \langle a_i, x\rangle \le 0\}$ and linear with constant derivative $c_i a_i^T \in \mathbb{R}^D \times \mathbb{R}^d$ on the other half-space. Thus, the a_i are the normal vectors of the separating hyperplanes $\{x \in \mathbb{R}^d \colon \langle a_i, x\rangle = 0\}$ and consequently we refer to the weight vectors a_i also as the directions of Θ. Moreover, for $\Theta \in \mathcal{N}_N$ it holds that $\mathcal{R}(\Theta)(0) = 0$ and, as long as the domain of interest $U \subseteq \mathbb{R}^d$ contains the origin, the Sobolev norm $\|\cdot\|_{W^{1,\infty}(U)}$ is equivalent to its semi-norm, since

$$\|\mathcal{R}(\Theta)\|_{L^\infty(U)} \le \sqrt{d}\,\operatorname{diam}(U)|\mathcal{R}(\Theta)|_{W^{1,\infty}}, \tag{20}$$

[2] This is a slight abuse of notation, justified by the the fact that $\mathcal{R}$ acts the same on $\mathcal{P}_N$ with zero biases b_1, b_2 and weights $A_1 = A$ and $A_2 = C$.

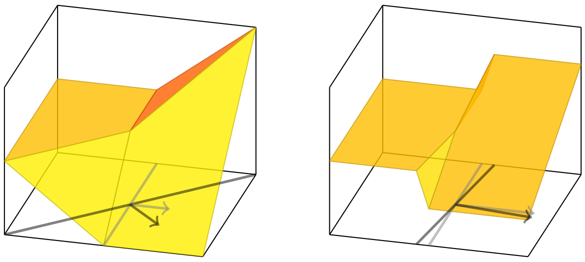


Figure 1: The figure shows g_k for $k = 1, 2$.

see also inequalities of Poincaré-Friedrichs type [14, Subsection 5.8.1]. Therefore, in the rest of the paper we will only consider the Sobolev semi-norm[3]

$$|\mathcal{R}(\Theta)|_{W^{1,\infty}(U)} = \operatorname*{ess\,sup}_{x\in U} \Big\| \sum_{i\in[m]\colon \langle a_i,x\rangle>0} c_i a_i^T \Big\|_\infty . \tag{21}$$

In (21) one can see that in our setting $|\cdot|_{W^{1,\infty}(U)}$ is independent of U (as long as U contains a neighbourhood of the origin) and will thus be abbreviated by $|\cdot|_{W^{1,\infty}}$.

2.1 Failure of inverse stability w.r.t. uniform norm

All proofs for this section can be found in Appendix A.2.2. We start by showing that inverse stability fails w.r.t. the uniform norm. This example is adapted from [34, Theorem 5.2] and represents, to the best of our knowledge, the only degeneracy which has already been observed before.

Example 2.1 (Failure due to exploding gradient)**.** *Let* $\Gamma := (0,0) \in \mathcal{N}_{(2,2,1)}$ *and* $g_k \in \mathcal{R}(\mathcal{N}_{(2,2,1)})$ *be given by (see Figure 1)*

$$g_k(x) := k\rho(\langle (k,0), x\rangle) - k\rho(\langle (k, -\tfrac{1}{k^2}), x\rangle), \quad k \in \mathbb{N}. \tag{22}$$

Then for every sequence $(\Phi_k)_{k\in\mathbb{N}} \subseteq \mathcal{N}_{(2,2,1)}$ *with* $\mathcal{R}(\Phi_k) = g_k$ *it holds that*

$$\lim_{k\to\infty} \|\mathcal{R}(\Phi_k) - \mathcal{R}(\Gamma)\|_{L^\infty((-1,1)^2)} = 0 \quad \textit{and} \quad \lim_{k\to\infty} \|\Phi_k - \Gamma\|_\infty = \infty. \tag{23}$$

In particular, note that inverse stability fails here even for a non-degenerate parametrization of the zero function $\Gamma = (0,0)$. However, for this type of counterexample the magnitude of the gradient of $\mathcal{R}(\Phi_k)$ needs to go to infinity, which is our motivation for looking at inverse stability w.r.t. $|\cdot|_{W^{1,\infty}}$.

2.2 Failure of inverse stability w.r.t. Sobolev norm

In this section we present four degenerate cases where inverse stability fails w.r.t. $|\cdot|_{W^{1,\infty}}$. This collection of counterexamples is complete in the sense that we can establish inverse stability under assumptions which are designed to exclude these four pathologies.

Example 2.2 (Failure due to complete unbalancedness)**.** *Let* $r > 0$, $\Gamma := \big((r,0),0\big) \in \mathcal{N}_{(2,1,1)}$ *and* $g_k \in \mathcal{R}(\mathcal{N}_{(2,1,1)})$ *be given by (see Figure 2)*

$$g_k(x) = \tfrac{1}{k}\rho(\langle (0,1), x\rangle), \quad k \in \mathbb{N}. \tag{24}$$

Then for every $k \in \mathbb{N}$ *and* $\Phi_k \in \mathcal{N}_{(2,1,1)}$ *with* $\mathcal{R}(\Phi_k) = g_k$ *it holds that*

$$|\mathcal{R}(\Phi_k) - \mathcal{R}(\Gamma)|_{W^{1,\infty}} = \tfrac{1}{k} \quad \textit{and} \quad \|\Phi_k - \Gamma\|_\infty \geq r. \tag{25}$$

This is a very simple example of a degenerate parametrization of the zero function, since $\mathcal{R}(\Gamma) = 0$ regardless of choice of r. The issue here is that we can have a weight pair, i.e. $((r,0),0)$, where the product is independent of the value of one of the parameters. Note that in Example A.4 one can see a slightly more subtle version of this pathology by considering $\Gamma_k := \big((k,0), \frac{1}{k^2}\big) \in \mathcal{N}_{(2,1,1)}$ instead. In that case one could still get an inverse stability estimate for each fixed k; the parameters of inverse

[3] For $m \in \mathbb{N}$ we abbreviate $[m] := \{1, \ldots, m\}$.

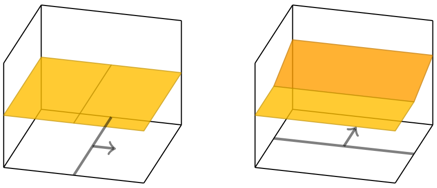


Figure 2: Shows $\mathcal{R}(\Gamma)$ $(r = 0.5)$ and g_3.

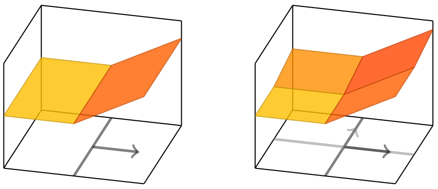


Figure 3: Shows $\mathcal{R}(\Gamma)$ and g_2.

stability (s, α) would however deteriorate with increasing k. In particular this demonstrates the need for some sort of balancedness of the parametrization, i.e. control over $\|c_i\|_\infty$ and $\|a_i\|_\infty$ individually relative to $\|c_i\|_\infty \|a_i\|_\infty$.
Inverse stability is also prevented by redundant directions as the following example illustrates.

Example 2.3 (Failure due to redundant directions)**.** *Let*

$$\Gamma := \left(\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, (1, 1) \right) \in \mathcal{N}_{(2,2,1)} \tag{26}$$

and $g_k \in \mathcal{R}(\mathcal{N}_{(2,2,1)})$ be given by (see Figure 3)

$$g_k(x) := 2\rho(\langle (1, 0), x \rangle) + \tfrac{1}{k}\rho(\langle (0, 1), x \rangle), \quad k \in \mathbb{N}. \tag{27}$$

Then for every $k \in \mathbb{N}$ and $\Phi_k \in \mathcal{N}_{(2,2,1)}$ with $\mathcal{R}(\Phi_k) = g_k$ it holds that

$$|\mathcal{R}(\Phi_k) - \mathcal{R}(\Gamma)|_{W^{1,\infty}} = \tfrac{1}{k} \quad \textit{and} \quad \|\Phi_k - \Gamma\|_\infty \geq 1. \tag{28}$$

The next example shows that not only redundant weight vectors can cause issues, but also weight vectors of opposite direction, as they would allow for a (balanced) degenerate parametrization of the zero function.

Example 2.4 (Failure due to opposite weight vectors 1)**.** *Let $a_i \in \mathbb{R}^d$, $i \in [m]$, be pairwise linearly independent with $\|a_i\|_\infty = 1$ and $\sum_{i=1}^m a_i = 0$. We define*

$$\Gamma := \left([a_1| \dots |a_m| - a_1| \dots | - a_m]^T, (1, \dots, 1, -1, \dots, -1) \right) \in \mathcal{N}_{(d,2m,1)}. \tag{29}$$

Now let $v \in \mathbb{R}^d$ with $\|v\|_\infty = 1$ be linearly independent to each a_i, $i \in [m]$, and let $g_k \in \mathcal{R}(\mathcal{N}_{(d,2m,1)})$ be given by (see Figure 4)

$$g_k(x) = \tfrac{1}{k}\rho(\langle v, x \rangle), \quad k \in \mathbb{N}. \tag{30}$$

Then there exists a constant $C > 0$ such that for every $k \in \mathbb{N}$ and every $\Phi_k \in \mathcal{N}_{(d,2m,1)}$ with $\mathcal{R}(\Phi_k) = g_k$ it holds that

$$|\mathcal{R}(\Phi_k) - \mathcal{R}(\Gamma)|_{W^{1,\infty}} = \tfrac{1}{k} \quad \textit{and} \quad \|\Phi_k - \Gamma\|_\infty \geq C. \tag{31}$$

Thus we will need an assumption which prevents each individual Γ in our restricted set from having pairwise linearly dependent weight vectors, i.e. coinciding hyperplanes of non-differentiability. This, however, does not suffice as is demonstrated by the next example, which shows that the relation between the hyperplanes of the two realizations matters.

Example 2.5 (Failure due to opposite weight vectors 2)**.** *We define the weight vectors*

$$a_1^k = (k, k, \tfrac{1}{k}), \quad a_2^k = (-k, k, \tfrac{1}{k}), \quad a_3^k = (0, -\sqrt{2}k, \tfrac{1}{\sqrt{2}k}), \quad c^k = (k, k, \sqrt{2}k) \tag{32}$$

and consider the parametrizations (see Figure 5)

$$\Gamma_k := \left(\left[-a_1^k \right| - a_2^k \left| - a_3^k \right]^T, c^k \right) \in \mathcal{N}_{(3,3,1)}, \quad \Theta_k := \left(\left[a_1^k | a_2^k | a_3^k \right]^T, c^k \right) \in \mathcal{N}_{(3,3,1)}. \tag{33}$$

Then for every $k \in \mathbb{N}$ and every $\Phi_k \in \mathcal{N}_{(3,3,1)}$ with $\mathcal{R}(\Phi_k) = \mathcal{R}(\Theta_k)$ it holds that

$$|\mathcal{R}(\Phi_k) - \mathcal{R}(\Gamma_k)|_{W^{1,\infty}} = 3 \quad \textit{and} \quad \|\Phi_k - \Gamma_k\|_\infty \geq k. \tag{34}$$

Note that Γ and Θ need to have multiple exactly opposite weight vectors which add to something small (compared to the size of the individual vectors), but not zero, since otherwise reparametrization would be possible (see Lemma A.5).

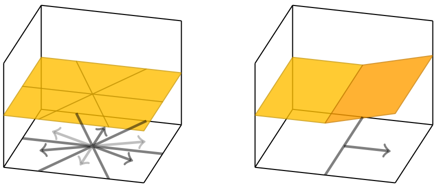


Figure 4: Shows $\mathcal{R}(\Gamma)$ and g_3 ($a_1 = (1, -\frac{1}{2})$, $a_2 = (-1, -\frac{1}{2})$, $a_3 = (0, 1)$, $v = (1, 0)$).

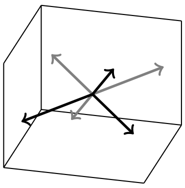


Figure 5: Shows the weight vectors of Θ_2 (grey) and Γ_2 (black).

3 Inverse stability for two-layer ReLU Networks

We now establish an inverse stability result using assumptions designed to exclude the pathologies from the previous section. First we present a rather technical theorem for output dimension one which considers a parametrization Γ in the unrestricted parametrization space $\mathcal{N}_N$ and a function g in the the corresponding function space $\mathcal{R}(\mathcal{N}_N)$. The aim is to use assumptions which are as weak as possible, while allowing us to find a parametrization Φ of g, whose distance to Γ can be bounded relative to $|g - \mathcal{R}(\Gamma)|_{W^{1,\infty}}$. We then continue by defining a restricted parametrization space $\mathcal{N}_N^*$, for which we get uniform inverse stability (meaning that we get the same estimate for every $\Gamma \in \mathcal{N}_N^*$).

Theorem 3.1 (Inverse stability at $\Gamma \in \mathcal{N}_N$)**.** *Let $d, m \in \mathbb{N}$, $N := (d, m, 1)$, $\beta \in [0, \infty)$, let $\Gamma = \left(\left[a_1^\Gamma|\dots|a_m^\Gamma\right]^T, c^\Gamma\right) \in \mathcal{N}_N$, $g \in \mathcal{R}(\mathcal{N}_N)$, and let $I^\Gamma := \{i \in [m] \colon a_i^\Gamma \neq 0\}$.*
Assume that the following conditions are satisfied:

C.1 It holds for all $i \in [m]$ with $\|c_i^\Gamma a_i^\Gamma\|_\infty \leq 2|g - \mathcal{R}(\Gamma)|_{W^{1,\infty}}$ that $|c_i^\Gamma|, \|a_i^\Gamma\|_\infty \leq \beta$.

C.2 It holds for all $i, j \in I^\Gamma$ with $i \neq j$ that $\frac{a_j^\Gamma}{\|a_j^\Gamma\|_\infty} \neq \frac{a_i^\Gamma}{\|a_i^\Gamma\|_\infty}$.

C.3 There exists a parametrization $\Theta = \left(\left[a_1^\Theta|\dots|a_m^\Theta\right]^T, c^\Theta\right) \in \mathcal{N}_N$ such that $\mathcal{R}(\Theta) = g$ and

(a) it holds for all $i, j \in I^\Gamma$ with $i \neq j$ that $\frac{a_j^\Gamma}{\|a_j^\Gamma\|_\infty} \neq -\frac{a_i^\Gamma}{\|a_i^\Gamma\|_\infty}$ and for all $i, j \in I^\Theta$ with $i \neq j$ that $\frac{a_j^\Theta}{\|a_j^\Theta\|_\infty} \neq -\frac{a_i^\Theta}{\|a_i^\Theta\|_\infty}$,

(b) it holds for all $i \in I^\Gamma$, $j \in I^\Theta$ that $\frac{a_i^\Gamma}{\|a_i^\Gamma\|_\infty} \neq -\frac{a_j^\Theta}{\|a_j^\Theta\|_\infty}$

where $I^\Theta := \{i \in [m] \colon a_i^\Theta \neq 0\}$.

Then there exists a parametrization $\Phi \in \mathcal{N}_N$ with

$$\mathcal{R}(\Phi) = g \quad \text{and} \quad \|\Phi - \Gamma\|_\infty \leq \beta + 2|g - \mathcal{R}(\Gamma)|_{W^{1,\infty}}^{\frac{1}{2}}. \tag{35}$$

The proof can be found in Appendix A.3.2. Note that each of the conditions in the theorem above corresponds directly to one of the pathologies in Section 2.2. Condition C.1, which deals with unbalancedness, only imposes an restriction on the weight pairs whose product is small compared to the distance of $\mathcal{R}(\Gamma)$ and g. As can be guessed from Example 2.2 and seen in the proof of Theorem 3.1, such a balancedness assumption is in fact only needed to deal with degenerate cases, where $\mathcal{R}(\Gamma)$ and g have parts with mismatching directions of negligible magnitude. Otherwise a matching reparametrization is always possible. Note that a balanced Γ (i.e. $|c_i^\Gamma| = \|a_i^\Gamma\|_\infty$) satisfies Condition C.1 with $\beta = (2|g - \mathcal{R}(\Gamma)|_{W^{1,\infty}})^{1/2}$.
It is also possible to relax the balancedness assumption by only requiring $|c_i^\Gamma|$ and $\|\Gamma_i\|_\infty$ to be close to $\|c_i^\Gamma a_i^\Gamma\|_\infty^{1/2}$, which would still give a similar estimate but with a worse exponent. In order to see that requiring balancedness does not restrict the space of realizations, observe that the ReLU is positively homogeneous (i.e. $\rho(\lambda x) = \lambda\rho(x)$ for all $\lambda \geq 0$, $x \in \mathbb{R}$). Thus balancedness can always be achieved simply by rescaling.
Condition C.2 requires Γ to have no redundant directions, the necessity of which is demonstrated by Example 2.3. Note that prohibiting redundant directions does not restrict the space of realizations,

see (87) in the appendix for details. From a practical point of view, enforcing this condition could be achieved by a regularization term using a barrier function. Alternatively on could employ a non-standard approach of combining such redundant neurons by changing one of them according to (87) and either setting the other one to zero or removing it entirely[4].

From a theoretical perspective the first two conditions are rather mild, in the sense that they only restrict the space of parametrizations and not the corresponding space of realizations. Specifically we can define the restricted parametrization space

$$\mathcal{N}'_{(d,m,D)} := \{\Gamma \in \mathcal{N}_{(d,m,D)} \colon \|c_i^\Gamma\|_\infty = \|a_i^\Gamma\|_\infty \text{ for all } i \in [m] \text{ and } \Gamma \text{ satisfies C.2}\} \tag{36}$$

for which we have $\mathcal{R}(\mathcal{N}'_N) = \mathcal{R}(\mathcal{N}_N)$. Note that the above definition as well as the following definition and theorem are for networks with arbitrary output dimensions, as the balancedness condition makes this extension rather straightforward.

In order to satisfy Conditions C.3a and C.3b we need to restrict the parametrization space in a way which also restricts the corresponding space of realizations. One possibility to do so is the following approach, which also incorporates the previous restrictions as well as the transition to networks without biases.

Definition 3.2 (Restricted parametrization space)**.** *Let $N = (d, m, D) \in \mathbb{N}^3$. We define*

$$\mathcal{N}^*_N := \left\{\Gamma \in \mathcal{N}'_N \colon (a_i^\Gamma)_{d-1}, (a_i^\Gamma)_d > 0 \text{ for all } i \in [m]\right\}. \tag{37}$$

While we no longer have $\mathcal{R}(\mathcal{N}^*_N) = \mathcal{R}(\mathcal{N}_N)$, Lemma A.6 shows that for every $\Theta \in \mathcal{P}_{(d,m,D)}$ there exists $\Gamma \in \mathcal{N}^*_{(d+2,m+1,D)}$ such that for all $x \in \mathbb{R}^d$ it holds that

$$\mathcal{R}(\Gamma)(x_1, \dots, x_d, 1, -1) = \mathcal{R}(\Theta)(x_1, \dots, x_d). \tag{38}$$

In particular, this means that for any optimization problem over an unrestricted parametrization space $\mathcal{P}_{(d,m,D)}$, there is a corresponding optimization problem over the parametrization space $\mathcal{N}^*_{(d+2,m+1,D)}$ whose solution is at least as good (see Corollary 1.4). Our main result now states that for such a restricted parametrization space we have uniform $(4, 1/2)$ inverse stability w.r.t. $|\cdot|_{W^{1,\infty}}$, a proof of which can be found in Appendix A.3.2.

Theorem 3.3 (Inverse stability on $\mathcal{N}^*_N$)**.** *Let $N \in \mathbb{N}^3$. For all $\Gamma \in \mathcal{N}^*_N$ and $g \in \mathcal{R}(\mathcal{N}^*_N)$ there exists a parametrization $\Phi \in \mathcal{N}^*_N$ with*

$$\mathcal{R}(\Phi) = g \quad \textit{and} \quad \|\Phi - \Gamma\|_\infty \leq 4|g - \mathcal{R}(\Gamma)|^{\frac{1}{2}}_{W^{1,\infty}}. \tag{39}$$

4 Outlook

This contribution investigates the potential insights which may be gained from studying the optimization problem over the space of realizations, as well as the difficulties encountered when trying to connect it to the parametrized problem. While Theorem 1.3 and Theorem 3.3 offer some compelling preliminary answers, there are multiple ways in which they can be extended.

To obtain our inverse stability result for shallow ReLU networks we studied sums of ridge functions. Extending this result to deep ReLU networks requires understanding their behaviour under composition. In particular, we have ridge functions which vanish on some half space, i.e. colloquially speaking each neuron may "discard half the information" it receives from the previous layer. This introduces a new type of degeneracy, which one will have to deal with.

Another interesting direction is an extension to inverse stability w.r.t. some weaker norm like $\|\cdot\|_{L^\infty}$ or a fractional Sobolev norm under stronger restrictions on the space of parametrizations (see Lemma A.7 for a simple approach using very strong restrictions).

Lastly, note that Theorem 1.3 is not specific to the ReLU activation function and thus also incentivizes the study of inverse stability for any other activation function.

From an applied point of view, Conditions C.1-C.3 motivate the implementation of corresponding regularization (i.e. penalizing unbalancedness and redundancy in the sense of parallel weight vectors) in state-of-the-art networks, in order to explore whether preventing inverse stability leads to improved performance in practice. Note that there already are results using, e.g. *cosine similarity*, as regularizer to prevent parallel weight vectors [4, 35] as well as approaches, called *Sobolev Training*, reporting better generalization and data-efficiency by employing a Sobolev norm based loss [12].

[4] This could be of interest in the design of dynamic network architectures [26, 28, 40] and is also closely related to the co-adaption of neurons, to counteract which, dropout was invented [21].

Acknowledgment

The research of JB and DE was supported by the Austrian Science Fund (FWF) under grants I3403-N32 and P 30148. The authors would like to thank Pavol Harár for helpful comments.

References

[1] Z. Allen-Zhu, Y. Li, and Z. Song. A Convergence Theory for Deep Learning via Over-Parameterization. *arXiv:1811.03962*, 2018.

[2] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009.

[3] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pages 254–263, 2018.

[4] N. Bansal, X. Chen, and Z. Wang. Can we gain more from orthogonality regularizations in training deep networks? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4261–4271. Curran Associates, Inc., 2018.

[5] P. L. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. *arXiv:1706.08498*, 2017.

[6] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *arXiv:1703.02930*, 2017.

[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:1809.03062*, 2018.

[8] J. Berner, D. Elbrächter, P. Grohs, and A. Jentzen. Towards a regularity theory for ReLU networks–chain rule and global error estimates. *arXiv:1905.04992*, 2019.

[9] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen. Optimal approximation with sparsely connected deep neural networks. *arXiv:1705.01714*, 2017.

[10] M. Burger and A. Neubauer. Error Bounds for Approximation with Neural Networks . *Journal of Approximation Theory*, 112(2):235–250, 2001.

[11] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

[12] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu. Sobolev training for neural networks. In *Advances in Neural Information Processing Systems*, pages 4278–4287, 2017.

[13] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai. Gradient Descent Finds Global Minima of Deep Neural Networks. *arXiv:1811.03804*, 2018.

[14] L. C. Evans. *Partial Differential Equations (second edition)*. Graduate studies in mathematics. American Mathematical Society, 2010.

[15] L. C. Evans and R. F. Gariepy. *Measure Theory and Fine Properties of Functions, Revised Edition*. Textbooks in Mathematics. CRC Press, 2015.

[16] K.-I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, 1989.

[17] N. Golowich, A. Rakhlin, and O. Shamir. Size-independent sample complexity of neural networks. *arXiv:1712.06541*, 2017.

[18] I. J. Goodfellow, O. Vinyals, and A. M. Saxe. Qualitatively characterizing neural network optimization problems. *arXiv:1412.6544*, 2014.

[19] R. Gribonval, G. Kutyniok, M. Nielsen, and F. Voigtlaender. Approximation spaces of deep neural networks. *arXiv: 1905.01208*, 2019.

[20] I. Gühring, G. Kutyniok, and P. Petersen. Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms. *arXiv:1902.07896*, 2019.

[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.

[22] K. Kawaguchi. Deep learning without poor local minima. In *Advances in neural information processing systems*, pages 586–594, 2016.

[23] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399, 2018.

[24] Y. Li and Y. Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pages 8157–8166, 2018.

[25] Y. Li and Y. Yuan. Convergence analysis of two-layer neural networks with ReLU activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

[26] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. *arXiv:1806.09055*, 2018.

[27] S. Mei, A. Montanari, and P.-M. Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[28] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat. Chapter 15 - evolving deep neural networks. In R. Kozma, C. Alippi, Y. Choe, and F. C. Morabito, editors, *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pages 293 – 312. Academic Press, 2019.

[29] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[30] Q. Nguyen and M. Hein. The loss surface of deep and wide neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 2603–2612. JMLR.org, 2017.

[31] J. Pennington and Y. Bahri. Geometry of neural network loss surfaces via random matrix theory. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 2798–2806. JMLR.org, 2017.

[32] D. Perekrestenko, P. Grohs, D. Elbrächter, and H. Bölcskei. The universal approximation power of finite-width deep ReLU networks. *arXiv:1806.01528*, 2018.

[33] P. Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *arXiv:1709.05289*, 2017.

[34] P. Petersen, M. Raslan, and F. Voigtlaender. Topological properties of the set of functions generated by neural networks of fixed size. *arXiv:1806.08459*, 2018.

[35] P. Rodríguez, J. Gonzalez, G. Cucurull, J. M. Gonfaus, and X. Roca. Regularizing cnns with locally constrained decorrelations. *arXiv:1611.01967*, 2016.

[36] I. Safran and O. Shamir. On the quality of the initial basin in overspecified neural networks. In *International Conference on Machine Learning*, pages 774–782, 2016.

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

[38] O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pages 71–79, 2013.

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

[40] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

A Appendix - Proofs and Additional Material

A.1 Section 1

A.1.1 Additional Material

Example A.1 (Without inverse stability: parameter minimum $\not\Rightarrow$ realization minimum)**.** *Consider the two domains*

$$D_1 := \{(x_1,x_2)\in(-1,1)^2\colon x_2>|x_1|\}, \quad D_2 := \{(x_1,x_2)\in(-1,1)^2\colon x_1>|x_2|\}. \tag{40}$$

For simplicity of presentation, assume we are given two samples $x^1\in D_1$, $x^2\in D_2$ with labels $y^1=0$, $y^2=1$. The corresponding MSE is

$$\mathcal{L}(g)=\tfrac{1}{2}\big((g(x^1))^2+(g(x^2)-1)^2\big) \tag{41}$$

for every $g\in C(\mathbb{R}^2,\mathbb{R})$. Let the zero realization be parametrized by[5]

$$\Gamma_*=(0,(-1,0))\in\mathcal{N}_{(2,1,1)} \tag{42}$$

with loss $\mathcal{L}(\mathcal{R}(\Gamma_*))=\frac{1}{2}$. Note that changing each weight by less than $\frac{1}{2}$ does not decrease the loss, as this rotates the vector $(-1,0)$ by at most $45°$. Thus Γ_* is a local minimum in the parametrization space. However, the sequence of realizations given by

$$g_k(x)=\tfrac{1}{k}\rho(x_1-x_2)=\mathcal{R}((1,-1),\tfrac{1}{k}) \tag{43}$$

satisfies that

$$\|g_k-\mathcal{R}(\Gamma_*)\|_{W^{1,\infty}((-1,1)^2)}=\|g_k\|_{W^{1,\infty}((-1,1)^2)}\le\tfrac{1}{k} \tag{44}$$

and

$$\mathcal{L}(g_k)=\tfrac{1}{2}(g_k(x^2)-1)^2<\tfrac{1}{2}=\mathcal{L}(\mathcal{R}(\Gamma_*)), \tag{45}$$

see Figure 6. Accordingly, $\mathcal{R}(\Gamma_*)$ is not *a local minimum in the realization space even w.r.t. the Sobolev norm. The problem occurs, since inverse stability fails due to unbalancedness of Γ_*.*

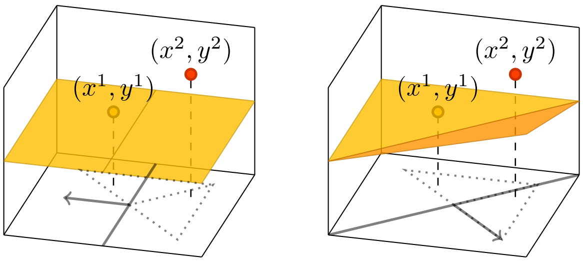


Figure 6: The figure shows the samples $((x^i,y^i))_{i=1,2}$, the realization $\mathcal{R}(\Gamma_*)$ of the local parameter minimum (left) and g_3 (right).

Theorem A.2 (Quality of local realization minima)**.** *Assume that*

$$\sup_{f\in S}\ \inf_{\Phi\in\Omega_N}\|\mathcal{R}(\Phi)-f\|<\eta \quad \textit{(approximability)}. \tag{46}$$

Let g_* be a local minimum with radius $r'\ge 2\eta$ of the optimization problem $\min_{g\in\mathcal{R}(\Omega_N)}\mathcal{L}(g)$. Then it holds for every $g\in\mathcal{R}(\Omega_N)$ (in particular for every global minimizer) that

$$\mathcal{L}(g_*)\le\mathcal{L}(g)+\tfrac{2c}{r'}\|g_*-g\|\eta. \tag{47}$$

Proof. Define $\lambda:=\frac{r'}{2\|g-g_*\|}$ and $f:=(1-\lambda)g_*+\lambda g\in S$. Due to (46) there is $\Phi\in\Omega_N$ such that $\|\mathcal{R}(\Phi)-f\|\le\eta$ and by the assumptions on g_* and $\mathcal{L}$ it holds that

$$\mathcal{L}(g_*)\le\mathcal{L}(\mathcal{R}(\Phi))\le\mathcal{L}(f)+c\eta\le(1-\lambda)\mathcal{L}(g_*)+\lambda\mathcal{L}(g)+c\eta.$$

This completes the proof. See Figure 7 for illustration. □

[5] See notation in the beginning of Section 2.

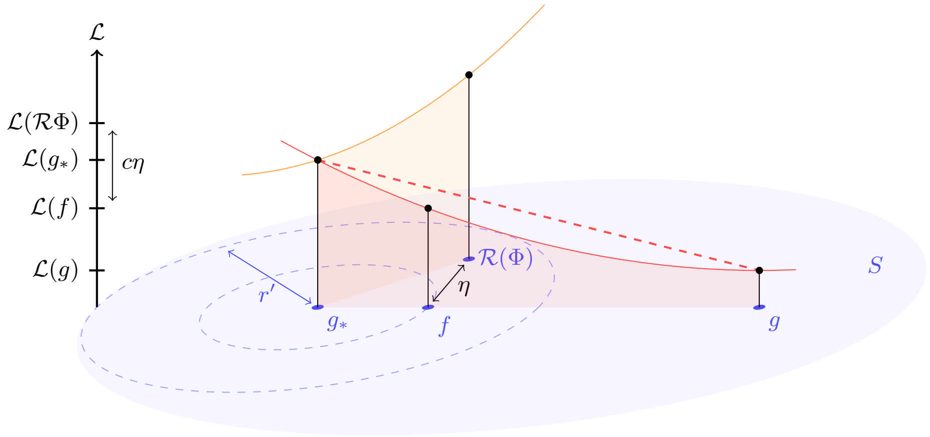


Figure 7: The figure illustrates the proof idea of Theorem A.2. Note that decreasing η, c, $\|g_* - g\|$ or increasing r' leads to a better local minimum due to the convexity of the loss function (red).

A.1.2 Proofs

Proof of Proposition 1.2. By Definition 1.1 we know that for every $g \in \mathcal{R}(\Omega)$ with $\|g - \mathcal{R}(\Gamma_*)\| \leq (\frac{r}{s})^{1/\alpha}$ there exists $\Phi \in \Omega$ with

$$\mathcal{R}(\Phi) = g \quad \text{and} \quad \|\Phi - \Gamma_*\|_\infty \leq s\|g - \mathcal{R}(\Gamma_*)\|^\alpha \leq r. \tag{48}$$

Therefore by assumption it holds that

$$\mathcal{L}(\mathcal{R}(\Gamma_*)) \leq \mathcal{L}(\mathcal{R}(\Phi)) = \mathcal{L}(g). \tag{49}$$

which proves the claim. □

Proof of Theorem 1.3. Let $\varepsilon, r > 0$, define $r' := (\frac{r}{s})^{1/\alpha}$ and $\eta := \min\{(\frac{2c}{r'} \operatorname{diam}(S))^{-1}\varepsilon, \frac{r'}{2}\}$. Then compactness of S implies the existence of an architecture $n(\varepsilon, r) \in \mathcal{A}_L$ such that for every $N \in \mathcal{A}_L$ with $N_1 \geq n_1(\varepsilon, r), \dots, N_{L-1} \geq n_{L-1}(\varepsilon, r)$ the approximability assumption (46) is satisfied. Let now Γ_* be a local minimum with radius at least r of $\min_{\Gamma \in \Omega_N} \mathcal{L}(\mathcal{R}(\Gamma))$. As we assume uniform (s, α) inverse stability, Proposition 1.2 implies that $\mathcal{R}(\Gamma_*)$ is a local minimum of the optimization problem $\min_{g \in \mathcal{R}(\Omega_N)} \mathcal{L}(g)$ with radius at least $r' = (\frac{r}{s})^{1/\alpha} \geq 2\eta$. Theorem A.2 establishes the claim. □

Proof of Corollary 1.4. We simply combine the main observations from our paper. First, note that the assumptions imply that the restricted parametrization space Ω, which we are optimizing over, is the space $\mathcal{N}^*_{(d+2, N_1+1, D)}$ from Definition 3.2. Secondly, Theorem 3.3 implies that the realization map is $(4, 1/2)$ inverse stable on Ω. Thus, Proposition 1.2 directly proves Claim 1. For the proof of Claim 2 we make use of Lemma A.6. It implies that for every $\Theta \in \mathcal{P}_{(d, N_1, D)}$ there exists $\Gamma \in \Omega$ such that it holds that

$$\frac{1}{n} \sum_{i=1}^{n} \|\mathcal{R}(\Gamma)(\tilde{x}^i) - y^i\|^2 = \frac{1}{n} \sum_{i=1}^{n} \|\mathcal{R}(\Theta)(x^i) - y^i\|^2, \tag{50}$$

which proves the claim. □

A.2 Section 2

A.2.1 Additional Material

Lemma A.3 (Reparametrization in case of linearly independent weight vectors)**.** *Let*

$$\Theta = (A^\Theta, C^\Theta) = \left([a_1^\Theta | \dots | a_m^\Theta]^T, [c_1^\Theta | \dots | c_m^\Theta]\right) \in \mathcal{N}_{(d,m,D)} \tag{51}$$

with linearly independent weight vectors $(a_i^\Theta)_{i=1}^m$ *and* $\min_{i\in[m]} \|c_i^\Theta\|_\infty > 0$ *and let*

$$\Phi = (A^\Phi, B^\Phi) = \big([a_1^\Phi|\dots|a_m^\Phi]^T, [c_1^\Phi|\dots|c_m^\Phi]\big) \in \mathcal{N}_{(d,m,D)} \tag{52}$$

with $\mathcal{R}(\Phi) = \mathcal{R}(\Theta)$*. Then there exists a permutation* $\pi\colon [m] \to [m]$ *such that for every* $i \in [m]$ *there exist* $\lambda_i \in (0,\infty)$ *with*

$$a_i^\Phi = \lambda_i a_{\pi(i)}^\Theta \quad \text{and} \quad c_i^\Phi = \tfrac{1}{\lambda_i} c_{\pi(i)}^\Theta. \tag{53}$$

This means that, up to reordering and rebalancing, Θ *is the unique parametrization of* $\mathcal{R}(\Theta)$*.*

Proof. First we define for every $s \in \{0,1\}^m$ the corresponding open orthant

$$O^s := \{x \in \mathbb{R}^m\colon x_1(2s_1-1) > 0, \dots, x_m(2s_m-1) > 0\} \subseteq \mathbb{R}^m. \tag{54}$$

By assumption A^Θ has rank m, i.e. is surjective, and therefore the preimages of the orthants

$$H^s := \{x \in \mathbb{R}^d\colon A^\Theta x \in O^s\} \subseteq \mathbb{R}^d, \quad s \in \{0,1\}^m, \tag{55}$$

are disjoint, non-empty open sets. Note that on each H^s the realization $\mathcal{R}(\Theta)$ is linear with

$$\mathcal{R}(\Theta)(x) = C^\Theta \operatorname{diag}(s) A^\Theta x \quad \text{and} \quad D\mathcal{R}(\Theta)(x) = C^\Theta \operatorname{diag}(s) A^\Theta. \tag{56}$$

Since A^Θ has full row rank, it has a right inverse. Thus we have for $s, t \in \{0,1\}^m$ that

$$C^\Theta \operatorname{diag}(s) A^\Theta = C^\Theta \operatorname{diag}(t) A^\Theta \implies C^\Theta \operatorname{diag}(s) = C^\Theta \operatorname{diag}(t). \tag{57}$$

Note that $C^\Theta \operatorname{diag}(s) = C^\Theta \operatorname{diag}(t)$ can only hold if $s = t$ due to the assumptions that $\|c_i^\Theta\|_\infty \neq 0$ for all $i \in [m]$. Thus the above establishes that for $s, t \in \{0,1\}^m$ it holds that

$$C^\Theta \operatorname{diag}(s) A^\Theta = C^\Theta \operatorname{diag}(t) A^\Theta \quad \text{if and only if} \quad s = t, \tag{58}$$

i.e. $\mathcal{R}(\Theta)$ has different derivatives on its 2^m linear regions. In order for $\mathcal{R}(\Phi)$ to have matching linear regions and matching derivatives on each one of them, there must exist a permutation matrix $P \in \{0,1\}^{m\times m}$ such that for every $s \in \{0,1\}^m$

$$PA^\Phi x \in O^s \quad \text{for every } x \in H^s. \tag{59}$$

Thus, there exist $(\lambda_i)_{i=1}^m \in (0,\infty)^m$ such that

$$A^\Phi = \operatorname{diag}(\lambda_1, \dots, \lambda_m) P^T A^\Theta. \tag{60}$$

The assumption that $D\mathcal{R}(\Theta) = D\mathcal{R}(\Psi)$, together with (56) for $s = (1, \dots, 1)$, implies that

$$C^\Phi = C^\Theta P \operatorname{diag}(\tfrac{1}{\lambda_1}, \dots, \tfrac{1}{\lambda_m}), \tag{61}$$

which proves the claim. □

Example A.4 (Failure due to unbalancedness)**.** *Let*

$$\Gamma_k := \big((k,0), \tfrac{1}{k^2}\big) \in \mathcal{N}_{(2,1,1)}, \quad k \in \mathbb{N}, \tag{62}$$

and $g_k \in \mathcal{R}(\mathcal{N}_{(2,1,1)})$ *be given by*

$$g_k(x) = \tfrac{1}{k}\rho(\langle(0,1), x\rangle), \quad k \in \mathbb{N}. \tag{63}$$

The only way to parametrize g_k *is* $g_k(x) = \mathcal{R}(\Phi_k)(x) = c\rho(\langle(0,a), x\rangle)$ *with* $a, c > 0$ *(see Lemma A.3), and we have*

$$|\mathcal{R}(\Phi_k) - \mathcal{R}(\Gamma_k)|_{W^{1,\infty}} \leq \tfrac{1}{k} \quad \text{and} \quad \|\Phi_k - \Gamma_k\|_\infty \geq k. \tag{64}$$

Lemma A.5. *Let* $d, m \in \mathbb{N}$ *and* $a_i \in \mathbb{R}^d$*,* $i \in [m]$*, such that* $\sum_{i\in[m]} a_i = 0$*. Then it holds for all* $x \in \mathbb{R}^d$ *that*

$$\sum_{i\in[m]} \rho(\langle a_i, x\rangle) = \sum_{i\in[m]} \rho(\langle -a_i, x\rangle). \tag{65}$$

Proof. By assumption we have for all $x \in \mathbb{R}^d$ that $\sum_{i\in[m]}\langle a_i, x\rangle = 0$. This implies for all $x \in \mathbb{R}^d$ that

$$\sum_{i\in[m]\colon \langle a_i,x\rangle\geq 0} \langle a_i, x\rangle - \sum_{i\in[m]} \langle a_i, x\rangle = \sum_{i\in[m]\colon \langle a_i,x\rangle\leq 0} -\langle a_i, x\rangle, \tag{66}$$

which proves the claim. □

A.2.2 Proofs

Proof of Example 2.1. We have for every $k \in \mathbb{N}$ that

$$\|g_k\|_{L^\infty((-1,1)^2)} \leq \tfrac{1}{k} \quad \text{and} \quad |g_k|_{W^{1,\infty}} = k^2. \tag{67}$$

Assume that there exists sequence of networks $(\Phi_k)_{k\in\mathbb{N}} \subseteq \mathcal{N}_{(2,2,1)}$ with $\mathcal{R}(\Phi_k) = g_k$ and with uniformly bounded parameters, i.e. $\sup_{k\in\mathbb{N}} \|\Phi_k\|_\infty < \infty$. Note that there exists a constant C (depending only on the network architecture) such that the realizations $\mathcal{R}(\Phi_k)$ are Lipschitz continuous with

$$\operatorname{Lip}(\mathcal{R}(\Phi_k)) \leq C\|\Phi_k\|_\infty^2$$

(see [34, Prop. 5.1]). It follows that $|\mathcal{R}(\Phi_k)|_{W^{1,\infty}} \leq \operatorname{Lip}(\mathcal{R}(\Phi_k))$ is uniformly bounded which contradicts (67). □

Proof of Example 2.2. The only way to parametrize g_k is $g_k(x) = \mathcal{R}(\Phi_k)(x) = c\rho(\langle(0,a), x\rangle)$ with $a, c > 0$ (see also Lemma A.3), which proves the claim. □

Proof of Example 2.3. Any parametrization of g_k must be of the form $\Phi_k := (A, c) \in \mathbb{R}^{2\times 2} \times \mathbb{R}^{1\times 2}$ with

$$A = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \quad \text{or} \quad A = \begin{bmatrix} 0 & a_2 \\ a_1 & 0 \end{bmatrix} \tag{68}$$

(see Lemma A.3). Thus it holds that $\|\Phi_k - \Gamma\|_\infty \geq \|(1,0) - (0,a_2)\|_\infty \geq 1$ and the proof is completed by direct calculation. □

Proof of Example 2.4. Let Φ_k be an arbitrary parametrization of g_k given by

$$\Phi_k = \big([\tilde{a}_1|\tilde{a}_2|\dots|\tilde{a}_{2m}]^T, \tilde{c}\big) \in \mathcal{N}_{(d,2m,1)} \tag{69}$$

As g_k has two linear regions separated by the hyperplane with normal vector v, there exists $j \in [2m]$ and $\lambda \in \mathbb{R} \setminus \{0\}$ such that

$$\tilde{a}_j = \lambda v. \tag{70}$$

The distance of any weight vector $\pm a_i$ of Γ to the line $\{\lambda v \colon \lambda \in \mathbb{R}\}$ can be lower bounded by

$$\|\pm a_i - \lambda v\|_\infty^2 \geq \tfrac{1}{d}\|\pm a_i - \lambda v\|_2^2 \geq \tfrac{1}{d^2}\left[\|a_i\|_2^2\|v\|_2^2 - \langle a_i, v\rangle^2\right], \quad i \in [m], \lambda \in \mathbb{R}. \tag{71}$$

The Cauchy-Schwarz inequality and the linear independence of v to each a_i, $i \in [m]$, establishes that $C := \frac{1}{d^2}\min_{i\in[m]}\left[\|a_i\|_2^2\|v\|_2^2 - \langle a_i, v\rangle^2\right] > 0$. Together with the fact that $\mathcal{R}(\Gamma) = 0$, this completes the proof. □

Proof of Example 2.5. Since $x = \rho(x) - \rho(-x)$ for every $x \in \mathbb{R}$, the difference of the realizations is linear, i.e.

$$\mathcal{R}(\Theta_k) - \mathcal{R}(\Gamma_k) = \langle c_1^k a_1^k + c_2^k a_2^k + c_3^k a_3^k, x\rangle = \langle(0,0,3), x\rangle \tag{72}$$

and thus the difference of the gradients is constant, i.e.

$$|\mathcal{R}(\Theta_k) - \mathcal{R}(\Gamma_k)|_{W^{1,\infty}} = 3, \quad k \in \mathbb{N}. \tag{73}$$

However, regardless of the balancing and reordering of the weight vectors a_i^k, $i \in [3]$, we have that

$$\|\Theta_k - \Gamma_k\|_\infty \geq k. \tag{74}$$

By Lemma A.3, up to balancing and reordering, there does not exist any other parametrization of Θ_k with the same realization. □

A.3 Section 3

A.3.1 Additional Material

Lemma A.6. *Let $d, m, D \in \mathbb{N}$ and $\Theta \in \mathcal{P}_{(d,m,D)}$. Then there exists $\Gamma \in \mathcal{N}^*_{(d+2,m+1,D)}$ such that for all $x \in \mathbb{R}^d$ it holds that*

$$\mathcal{R}(\Gamma)(x_1, \dots, x_d, 1, -1) = \mathcal{R}(\Theta)(x). \tag{75}$$

Proof. Since $\Theta \in \mathcal{P}_{(d,m,D)}$ it can be written as

$$\Theta = \Big(\big(A, b\big), \big(c, e\big)\Big) = \big(([a_1|\dots|a_m]^T, b), ([c_1|\dots|c_m], e)\big) \tag{76}$$

with

$$\mathcal{R}(\Theta)(x) = \sum_{i=1}^{m} c_i \rho(\langle a_i, x\rangle + b_i) + e, \quad x \in \mathbb{R}^d, \tag{77}$$

where $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, $C \in \mathbb{R}^{D \times m}$, and $e \in \mathbb{R}^D$. We define for $i \in [m]$

$$b_i^+ := \begin{cases} b_i + 1 & : b_i \geq 0 \\ 1 & : b_i < 0 \end{cases}, \quad \text{and} \quad b_i^- := \begin{cases} 1 & : b_i \geq 0 \\ -b_i + 1 & : b_i < 0 \end{cases} \tag{78}$$

and observe that $b_i^+ > 0$, $b_i^- > 0$, and $b_i^+ - b_i^- = b_i$. For $i \in [m]$ let

$$c_i^* := \begin{cases} c_i & : \|c_i\|_\infty \neq 0 \\ (1, \dots, 1) & : \|c_i\|_\infty = 0 \end{cases} \tag{79}$$

and

$$a_i^* := \begin{cases} (a_{i,1}, \dots, a_{i,d}, b_i^+, b_i^-) & : \|c_i\|_\infty \neq 0 \\ (0, \dots, 0, 1, 1) & : \|c_i\|_\infty = 0 \end{cases}. \tag{80}$$

Note that we have

$$\mathcal{R}(\Theta)(x) = \sum_{i=1}^{m} c_i^* \rho(\langle a_i^*, (x_1, \dots, x_d, 1, -1)\rangle) + e, \quad x \in \mathbb{R}^d. \tag{81}$$

To include the second bias e let

$$c_{m+1}^* := \begin{cases} e & : e \neq 0 \\ (1, \dots, 1) & : e = 0 \end{cases}, \quad \text{and} \quad a_{m+1}^* := \begin{cases} (0, \dots, 0, 2, 1) & : e \neq 0 \\ (0, \dots, 0, 1, 1) & : e = 0 \end{cases}. \tag{82}$$

In order to balance the network, let $a_i^\Gamma = a_i^* (\frac{\|c_i^*\|_\infty}{\|a_i^*\|_\infty})^{1/2}$ and $c_i^\Gamma = c_i^* (\frac{\|a_i^*\|_\infty}{\|c_i^*\|_\infty})^{1/2}$ for every $i \in [m+1]$. Then the claim follows by direct computation. □

A.3.2 Proofs

Proof of Theorem 3.1. Without loss of generality[6], we can assume for all $i \in [m]$ that $a_i^\Theta = 0$ if and only if $c_i^\Theta = 0$. We now need to show that there always exists a way to reparametrize $\mathcal{R}(\Theta)$ such that the architecture remains the same and (35) is satisfied. For simplicity of notation we will write $r := |g - \mathcal{R}(\Gamma)|_{W^{1,\infty}}$ throughout the proof. Let $f_i^\Gamma : \mathbb{R}^d \to \mathbb{R}$ resp. $f_i^\Theta : \mathbb{R}^d \to \mathbb{R}$ be the part that is contributed by the i-th neuron, i.e.

$$\mathcal{R}(\Gamma) = \sum_{i=1}^{m} f_i^\Gamma \quad \text{with} \quad f_i^\Gamma(x) := c_i^\Gamma \rho(\langle a_i^\Gamma, x\rangle), \tag{83}$$

$$g = \mathcal{R}(\Theta) = \sum_{i=1}^{m} f_i^\Theta \quad \text{with} \quad f_i^\Theta(x) := c_i^\Theta \rho(\langle a_i^\Theta, x\rangle). \tag{84}$$

[6] In case one of them is zero, the other one can be set to zero without changing the realization.

Further let

$$\begin{aligned} H^+_{\Gamma,i} &:= \{x \in \mathbb{R}^d \colon \langle a^\Gamma_i, x\rangle > 0\}, \\ H^0_{\Gamma,i} &:= \{x \in \mathbb{R}^d \colon \langle a^\Gamma_i, x\rangle = 0\}, \\ H^-_{\Gamma,i} &:= \{x \in \mathbb{R}^d \colon \langle a^\Gamma_i, x\rangle < 0\}. \end{aligned} \tag{85}$$

By conditions C.2 and C.3a we have for all $i, j \in I^\Gamma$ that

$$i \neq j \implies H^0_{\Gamma,i} \neq H^0_{\Gamma,j}. \tag{86}$$

Further note that we can reparametrize $\mathcal{R}(\Theta)$ such that the same holds there. To this end observe that

$$c\rho(\langle a, x\rangle) + c'\rho(\langle a', x\rangle) = (c + c'\tfrac{\|a'\|_\infty}{\|a\|_\infty})\rho(\langle a, x\rangle), \tag{87}$$

given that a' is a positive multiple of a. Specifically, let $(J_k)_{k=1}^K$ be a partition of I^Θ (i.e. $J_k \neq \emptyset$, $\cup_{k=1}^K J_k = I^\Theta$ and $J_k \cap J_{k'} = \emptyset$ if $k \neq k'$), such that for all $k \in [K]$ it holds that

$$i, j \in J_k \implies \frac{a^\Theta_j}{\|a^\Theta_j\|_\infty} = \frac{a^\Theta_i}{\|a^\Theta_i\|_\infty}. \tag{88}$$

We denote by j_k the smallest element in J_k and make the following replacements, for all $i \in I^\Theta$, without changing the realization of Θ:

$$a^\Theta_i \mapsto a^\Theta_i, c^\Theta_i \mapsto \sum_{j \in J_k} c^\Theta_j \tfrac{\|a^\Theta_j\|_\infty}{\|a^\Theta_{j_k}\|_\infty}, \quad \text{if } i \in J_k \text{ and } i = j_k, \tag{89}$$

$$a^\Theta_i \mapsto 0, c^\Theta_i \mapsto 0, \qquad \text{if } i \in J_k \text{ and } i \neq j_k. \tag{90}$$

Note that we also update the set $I^\Theta := \{i \in [m] \colon a^\Theta_i \neq 0\}$ accordingly. Let now

$$\begin{aligned} H^+_{\Theta,i} &:= \{x \in \mathbb{R}^d \colon \langle a^\Theta_i, x\rangle > 0\}, \\ H^0_{\Theta,i} &:= \{x \in \mathbb{R}^d \colon \langle a^\Theta_i, x\rangle = 0\}, \\ H^-_{\Theta,i} &:= \{x \in \mathbb{R}^d \colon \langle a^\Theta_i, x\rangle > 0\}. \end{aligned} \tag{91}$$

By construction and condition C.3a, we have for all $i, j \in I^\Theta$ that

$$i \neq j \implies H^0_{\Theta,i} \neq H^0_{\Theta,j}. \tag{92}$$

Note that we now have a parametrization Θ of g, where all weight vectors a^Θ_i are either zero (in which case the corresponding c^Θ_i are also zero) or pairwise linearly independent to each other nonzero weight vector.

Next, for $s \in \{0,1\}^m$, let

$$\begin{aligned} H^s_\Gamma &:= \bigcap_{i \in [m] \colon s_i = 1} H^+_{\Gamma,i} \cap \bigcap_{i \in [m] \colon s_i = 0} H^-_{\Gamma,i}, \\ H^s_\Theta &:= \bigcap_{i \in [m] \colon s_i = 1} H^+_{\Theta,i} \cap \bigcap_{i \in [m] \colon s_i = 0} H^-_{\Theta,i}, \end{aligned} \tag{93}$$

and

$$S^\Gamma := \{s \in \{0,1\}^m \colon H^s_\Gamma \neq \emptyset\}, \quad S^\Theta := \{s \in \{0,1\}^m \colon H^s_\Theta \neq \emptyset\}. \tag{94}$$

The H^s_Γ, $s \in S^\Gamma$, and H^s_Θ, $s \in S^\Theta$, are the interiors of the different linear regions of $\mathcal{R}(\Gamma)$ and $\mathcal{R}(\Theta)$ respectively. Next observe that the derivatives of f^Γ_i, f^Θ_i are (a.e.) given by

$$Df^\Gamma_i(x) = \mathbf{1}_{H^+_{\Gamma,i}}(x)\, c^\Gamma_i a^\Gamma_i, \quad Df^\Theta_i(x) = \mathbf{1}_{H^+_{\Theta,i}}(x)\, c^\Theta_i a^\Theta_i. \tag{95}$$

Note that for every $x \in H^s_\Gamma$, $y \in H^s_\Theta$ we have

$$\begin{aligned} D\mathcal{R}(\Gamma)(x) &= \sum_{i \in [m]} Df^\Gamma_i(x) = \sum_{i \in [m]} s_i c^\Gamma_i a^\Gamma_i =: \Sigma^\Gamma_s, \\ D\mathcal{R}(\Theta)(y) &= \sum_{i \in [m]} Df^\Theta_i(y) = \sum_{i \in [m]} s_i c^\Theta_i a^\Theta_i =: \Sigma^\Theta_s. \end{aligned} \tag{96}$$

Next we use that for $s \in S^\Gamma$, $t \in S^\Theta$ we have $|\Sigma_s^\Gamma - \Sigma_t^\Theta| \le r$ if $H_s^\Gamma \cap H_t^\Theta \neq \emptyset$, and compare adjacent linear regions of $\mathcal{R}(\Gamma) - \mathcal{R}(\Theta)$. Let now $i \in I^\Gamma$ and consider the following cases:
Case 1: We have $H_{\Gamma,i}^0 \neq H_{\Theta,j}^0$ for all $j \in I^\Theta$. This means that the Df_k^Θ, $k \in [m]$, and the Df_k^Γ, $k \in [m]\backslash\{i\}$, are the same on both sides near the hyperplane $H_{\Gamma,i}^0$, while the value of Df_i^Γ is 0 on one side and $c_i^\Gamma a_i^\Gamma$ on the other. Specifically, there exist $s^+, s^- \in S^\Gamma$ and $s^* \in S^\Theta$ such that $s_i^+ = 1$, $s_i^- = 0$, $s_j^+ = s_j^-$ for all $j \in [m]\backslash\{i\}$, and $H_\Gamma^{s^+} \cap H_\Theta^{s^*} \neq \emptyset$, $H_\Gamma^{s^-} \cap H_\Theta^{s^*} \neq \emptyset$, which implies

$$\|c_i^\Gamma a_i^\Gamma\|_\infty = \|(\Sigma_{s^+}^\Gamma - \Sigma_{s^*}^\Theta) - (\Sigma_{s^-}^\Gamma - \Sigma_{s^*}^\Theta)\|_\infty \le 2r. \tag{97}$$

Case 2: There exists $j \in I^\Theta$ such that $H_{\Gamma,i}^0 = H_{\Theta,j}^0$. Note that (86) ensures that $H_{\Gamma,i}^0 \neq H_{\Gamma,k}^0$ for $k \in [m] \setminus \{i\}$ and (92) ensures that $H_{\Theta,j}^0 \neq H_{\Gamma,k}^0$ for $k \in [m] \setminus \{j\}$. Moreover, Condition C.3b implies $H_{\Gamma,i}^+ = H_{\Theta,j}^+$. This means that the Df_k^Θ, $k \in [m]\backslash\{j\}$, and the Df_k^Γ, $k \in [m]\backslash\{i\}$, are the same on both sides near the hyperplane $H_{\Gamma,i}^0 = H_{\Theta,j}^0$, while the values of Df_i^Γ and Df_j^Θ change. Specifically there exist $s^+, s^- \in S^\Gamma$ and $t^+, t^- \in S^\Theta$ such that $s_i^+ = 1$, $s_i^- = 0$, $s_k^+ = s_k^-$ for all $k \in [m]\backslash\{i\}$, $t_j^+ = 1$, $t_j^- = 0$, $t_k^+ = t_k^-$ for all $k \in [m]\backslash\{j\}$ and $H_{s^+}^\Gamma \cap H_{t^+}^\Theta \neq \emptyset$, $H_{s^-}^\Gamma \cap H_{t^-}^\Theta \neq \emptyset$, which implies

$$\|c_i^\Gamma a_i^\Gamma - c_j^\Theta a_j^\Theta\|_\infty = \|(\Sigma_{s^+}^\Gamma - \Sigma_{t^+}^\Theta) - (\Sigma_{s^-}^\Gamma - \Sigma_{t^-}^\Theta)\|_\infty \le 2r. \tag{98}$$

Analogously we get for $i \in I^\Theta$ that $H_{\Theta,i}^0 \neq H_{\Gamma,j}^0$ for all $j \in I^\Gamma$ implies $\|c_i^\Theta a_i^\Theta\|_\infty \le 2r$. Next let

$$I_1 := \{i \in [m] \colon H_{\Gamma,i}^0 \neq H_{\Theta,j}^0 \text{ for all } j \in I^\Theta\} \cup \{i \in [m] \colon a_i^\Gamma = 0\} \tag{99}$$

and

$$I_2 := [m] \setminus I_1 = \{i \in [m] \colon \exists\, j \in I^\Theta \text{ such that } H_{\Gamma,i}^+ = H_{\Theta,j}^+\}. \tag{100}$$

Colloquially speaking, this shows that for every f_i^Γ with $i \in I_2$ there is a f_j^Θ with exactly matching half-spaces, i.e. $H_{\Gamma,i}^+ = H_{\Theta,j}^+$, and approximately matching gradients (Case 2). Moreover, all unmatched f_i^Γ and f_j^Θ must have a small gradient (Case 1).
Specifically, the above establishes that there exists a permutation $\pi \colon [m] \to [m]$ such that for every $i \in I_1$ it holds that

$$\|c_i^\Gamma a_i^\Gamma\|_\infty, \|c_{\pi(i)}^\Theta a_{\pi(i)}^\Theta\|_\infty \le 2r, \tag{101}$$

and for every $i \in I_2$ that

$$\|c_i^\Gamma a_i^\Gamma - c_{\pi(i)}^\Theta a_{\pi(i)}^\Theta\|_\infty \le 2r. \tag{102}$$

We make the following replacements, for all $i \in [m]$, without changing the realization of Θ:

$$a_i^\Theta \to a_{\pi(i)}^\Theta, \quad c_i^\Theta \to c_{\pi(i)}^\Theta. \tag{103}$$

In order to balance the weights of Θ for I_1, we further make the following replacements, for all $i \in I_1$ with $a_i^\Theta \neq 0$, without changing the realization of Θ:

$$a_i^\Theta \to \big(\tfrac{|c_i^\Theta|}{\|a_i^\Theta\|_\infty}\big)^{1/2}\, a_i^\Theta, \quad c_i^\Theta \to \big(\tfrac{\|a_i^\Theta\|_\infty}{|c_i^\Theta|}\big)^{1/2}\, c_i^\Theta. \tag{104}$$

This implies for every $i \in I_1$ that

$$|c_i^\Theta|, \|a_i^\Theta\|_\infty \le (2r)^{1/2}. \tag{105}$$

Moreover, due to Condition C.1, we get for every $i \in I_1$ that

$$|c_i^\Gamma|, \|a_i^\Gamma\|_\infty \le \beta. \tag{106}$$

Thus we get for every $i \in I_1$ that

$$|c_i^\Theta - c_i^\Gamma|, \|a_i^\Theta - a_i^\Gamma\|_\infty \le \beta + (2r)^{1/2}. \tag{107}$$

Next we (approximately) match the balancing of (c_i^Θ, a_i^Θ) to the balancing of (c_i^Γ, a_i^Γ) for $i \in I_2$, in order to derive estimates on $|c_i^\Theta - c_i^\Gamma|$ and $\|a_i^\Theta - a_i^\Gamma\|_\infty$ from (102). Specifically, we make the following replacements, for all $i \in I_2$, without changing the realization of Θ:

$$a_i^\Theta \to (\tfrac{|c_i^\Theta|}{\|a_i^\Theta\|_\infty})^{1/2}\, a_i^\Theta, \quad c_i^\Theta \to (\tfrac{\|a_i^\Theta\|_\infty}{|c_i^\Theta|})^{1/2}\, c_i^\Theta, \qquad \text{if } \|c_i^\Gamma a_i^\Gamma\|_\infty \leq 2r, \tag{108}$$

$$a_i^\Theta \to \frac{c_i^\Theta}{c_i^\Gamma} a_i^\Theta, \quad c_i^\Theta \to c_i^\Gamma, \qquad \text{if } \|c_i^\Gamma a_i^\Gamma\|_\infty > 2r, |c_i^\Gamma| > \|a_i^\Gamma\|_\infty, \tag{109}$$

$$a_i^\Theta \to a_i^\Gamma, \quad c_i^\Theta \to \frac{\|a_i^\Theta\|_\infty}{\|a_i^\Gamma\|_\infty} c_i^\Theta, \qquad \text{if } \|c_i^\Gamma a_i^\Gamma\|_\infty > 2r, |c_i^\Gamma| < \|a_i^\Gamma\|_\infty, \tag{110}$$

$$a_i^\Theta \to (\tfrac{|c_i^\Theta|}{\|a_i^\Theta\|_\infty})^{1/2}\, a_i^\Theta, \quad c_i^\Theta \to (\tfrac{\|a_i^\Theta\|_\infty}{|c_i^\Theta|})^{1/2}\, c_i^\Theta, \quad \text{if } \|c_i^\Gamma a_i^\Gamma\|_\infty > 2r, |c_i^\Gamma| = \|a_i^\Gamma\|_\infty. \tag{111}$$

Let now $i \in I_2$ and consider the following cases:
Case A: We have $\|c_i^\Gamma a_i^\Gamma\|_\infty \leq 2r$ which, together with (102), implies $\|c_i^\Theta a_i^\Theta\|_\infty \leq 4r$. Due to (108) and Condition C.1 it follows that

$$|c_i^\Theta - c_i^\Gamma|, \|a_i^\Theta - a_i^\Gamma\|_\infty \leq \beta + 2r^{1/2}. \tag{112}$$

Case B.1: We have $\|c_i^\Gamma a_i^\Gamma\|_\infty > 2r$ and $|c_i^\Gamma| > \|a_i^\Gamma\|_\infty$ which ensures $|c_i^\Gamma| > \|c_i^\Gamma a_i^\Gamma\|_\infty^{1/2}$. Due to (109) we get $c_i^\Theta = c_i^\Gamma$ and it follows that

$$\|a_i^\Theta - a_i^\Gamma\|_\infty = \frac{1}{|c_i^\Gamma|}\|c_i^\Theta a_i^\Theta - c_i^\Gamma a_i^\Gamma\|_\infty \leq \frac{2r}{\|c_i^\Gamma a_i^\Gamma\|_\infty^{1/2}} \leq (2r)^{1/2}. \tag{113}$$

Case B.2: We have $\|c_i^\Gamma a_i^\Gamma\|_\infty > 2r$ and $|c_i^\Gamma| < \|a_i^\Gamma\|_\infty$ which ensures $\|a_i^\Gamma\| > \|c_i^\Gamma a_i^\Gamma\|_\infty^{1/2}$. Due to (110) we get $a_i^\Theta = a_i^\Gamma$ and it follows that

$$|c_i^\Theta - c_i^\Gamma| = \frac{1}{\|a_i^\Gamma\|_\infty}\|c_i^\Theta a_i^\Theta - c_i^\Gamma a_i^\Gamma\|_\infty \leq \frac{2r}{\|c_i^\Gamma a_i^\Gamma\|_\infty^{1/2}} \leq (2r)^{1/2}. \tag{114}$$

Case B.3: We have $\|c_i^\Gamma a_i^\Gamma\|_\infty > 2r$ and $|c_i^\Gamma| = \|a_i^\Gamma\|_\infty$. Note that $\|c_i^\Gamma a_i^\Gamma\|_\infty > 2r$ and (102) ensure that $\operatorname{sgn}(c_i^\Theta) = \operatorname{sgn}(c_i^\Gamma)$, and that for $x, y > 0$ it holds that $|x - y| \leq |x^2 - y^2|^{1/2}$. Combining this with the definition of I_2, the reverse triangle inequality, and (111) implies that

$$\|a_i^\Theta - a_i^\Gamma\|_\infty \leq (2r)^{1/2} \quad \text{and} \quad |c_i^\Theta - c_i^\Gamma| \leq (2r)^{1/2}. \tag{115}$$

Combining (107), (112), (113), (114), and (115) establishes that

$$\|\Theta - \Gamma\|_\infty \leq \beta + 2r^{\frac{1}{2}}, \tag{116}$$

which completes the proof. □

Proof of Theorem 3.3. Let $\Theta \in \mathcal{N}_N^*$ be a parametrization of g, i.e. $\mathcal{R}(\Theta) = g$. We write

$$\Gamma = \left(\begin{bmatrix} a_1^\Gamma \\ \hline \vdots \\ \hline a_m^\Gamma \end{bmatrix}, [c_1^\Gamma | \dots | c_m^\Gamma] \right), \quad \Theta = \left(\begin{bmatrix} a_1^\Theta \\ \hline \vdots \\ \hline a_m^\Theta \end{bmatrix}, [c_1^\Theta | \dots | c_m^\Theta] \right) \in \mathcal{N}_{(d,m,D)}^* \tag{117}$$

and $r := |g - \mathcal{R}(\Gamma)|_{W^{1,\infty}}$. For convenience of notation we consider the weight vectors a_i^Γ, a_i^Θ here as row vectors in order to write the derivatives of the ridge functions as $c_i^\Gamma a_i^\Gamma$, $c_i^\Theta a_i^\Theta \in \mathbb{R}^{D \times d}$ without transposing.
We will now adjust the approach used in the proof of Theorem 3.1 to work for multi-dimensional outputs in the case of balanced networks. By definition of $\mathcal{N}_N^*$, the $(a_i^\Theta)_{i=1}^m$ are pairwise linearly independent and we can skip the first reparametrization step in (89) and (90).
The following "hyperplane-jumping" argument, which was used to get the estimates (97) and (98), works analogously since Conditions C.2 and C.3 are fulfilled by definition of $\mathcal{N}_N^*$. This establishes the existence of a permutation $\pi\colon [m] \to [m]$ and sets $I_1, I_2 \subseteq [m]$, as defined as in (99) and (100), such that for every $i \in I_1$ it holds that

$$\|c_i^\Gamma a_i^\Gamma\|_\infty, \|c_{\pi(i)}^\Theta a_{\pi(i)}^\Theta\|_\infty \leq 2r, \tag{118}$$

and for every $i \in I_2$ that

$$\|c_i^\Gamma a_i^\Gamma - c_{\pi(i)}^\Theta a_{\pi(i)}^\Theta\|_\infty \leq 2r. \tag{119}$$

As in (103), we make the following replacements, for all $i \in [m]$, without changing the realization of Θ:

$$a_i^\Theta \to a_{\pi(i)}^\Theta, \quad c_i^\Theta \to c_{\pi(i)}^\Theta. \tag{120}$$

Note that the weights of Θ are already balanced, i.e. we have for every $i \in [m]$ that

$$\|c_i^\Theta\|_\infty = \|a_i^\Theta\|_\infty = \|c_i^\Theta\|_\infty^{1/2}\|a_i^\Theta\|_\infty^{1/2} = \|c_i^\Theta a_i^\Theta\|_\infty^{1/2}. \tag{121}$$

Thus, we can skip the reparametrization step in (104) and get directly for every $i \in I_1$ that

$$\|c_i^\Theta - c_i^\Gamma\|_\infty \leq \|c_i^\Theta\|_\infty + \|c_i^\Gamma\|_\infty = \|c_i^\Theta a_i^\Theta\|_\infty^{1/2} + \|c_i^\Gamma a_i^\Gamma\|_\infty^{1/2} \leq 2(2r)^{1/2} \tag{122}$$

and analogously $\|a_i^\Theta - a_i^\Gamma\|_\infty \leq 2(2r)^{1/2}$.
For $i \in I_2$ we need to slightly deviate from the proof of Theorem 3.1. We can skip the reparametrization step in (108)-(111) due to balancedness and need to distinguish three cases:
Case A.1: We have $\|c_i^\Gamma a_i^\Gamma\|_\infty \leq 2r$ which, together with (119), implies $\|c_i^\Theta a_i^\Theta\|_\infty \leq 4r$. Due to balancedness it follows that

$$\|c_i^\Theta - c_i^\Gamma\|_\infty, \|a_i^\Theta - a_i^\Gamma\|_\infty \leq 4r^{1/2}. \tag{123}$$

Case A.2: We have $\|c_i^\Theta a_i^\Theta\|_\infty \leq 2r$ which, together with (119), implies $\|c_i^\Gamma a_i^\Gamma\|_\infty \leq 4r$. Again it follows that

$$\|c_i^\Theta - c_i^\Gamma\|_\infty, \|a_i^\Theta - a_i^\Gamma\|_\infty \leq 4r^{1/2}. \tag{124}$$

Case B: We have $\|c_i^\Theta a_i^\Theta\|_\infty > 2r$ and $\|c_i^\Gamma a_i^\Gamma\|_\infty > 2r$. Due to the definition of I_2 there exists $e_i \in \mathbb{R}^d$, $\lambda_i^\Gamma, \lambda_i^\Theta \in (0, \infty)$ with $\|e_i\|_\infty = 1$, $a_i^\Theta = \lambda_i^\Theta e_i$, and $a_i^\Gamma = \lambda_i^\Gamma e_i$. As in (115) we obtain that

$$\begin{aligned} \|a_i^\Theta - a_i^\Gamma\|_\infty &= \|e_i\|_\infty |\lambda_i^\Theta - \lambda_i^\Gamma| \leq |(\lambda_i^\Theta)^2 - (\lambda_i^\Gamma)^2|^{1/2} \\ &= |\|c_i^\Theta\|_\infty \|a_i^\Theta\|_\infty - \|c_i^\Gamma\|_\infty \|a_i^\Gamma\|_\infty|^{1/2} \\ &\leq \|c_i^\Theta a_i^\Theta - c_i^\Gamma a_i^\Gamma\|_\infty^{1/2} \leq (2r)^{1/2}. \end{aligned} \tag{125}$$

Let now w.l.o.g. $\|a_i^\Gamma\|_\infty \geq \|a_i^\Theta\|_\infty$ (otherwise we switch their roles in the following) which implies that $\lambda_i^\Gamma = \Delta_i + \lambda_i^\Theta$ with $\Delta_i = \lambda_i^\Gamma - \lambda_i^\Theta \geq 0$. Then it holds that

$$\begin{aligned} \|c_i^\Theta - c_i^\Gamma\|_\infty &= \frac{\|c_i^\Theta a_i^\Gamma - c_i^\Gamma a_i^\Gamma\|_\infty}{\|a_i^\Gamma\|_\infty} \leq \frac{\|c_i^\Theta a_i^\Gamma - c_i^\Theta a_i^\Theta\|_\infty + \|c_i^\Theta a_i^\Theta - c_i^\Gamma a_i^\Gamma\|_\infty}{\|a_i^\Gamma\|_\infty} \\ &\leq \frac{\|c_i^\Theta\|_\infty |\lambda_i^\Gamma - \lambda_i^\Theta| + 2r}{\lambda_i^\Gamma} = \frac{\lambda_i^\Theta \Delta_i + 2r}{\Delta_i + \lambda_i^\Theta} \\ &= \frac{(2r)^{1/2}(\Delta_i + \lambda_i^\Theta) - (\lambda_i^\Theta - (2r)^{1/2})((2r)^{1/2} - \Delta_i)}{\Delta_i + \lambda_i^\Theta} \leq (2r)^{1/2}. \end{aligned} \tag{126}$$

The last step holds due to (125) and the balancedness of Θ which ensure that

$$\lambda_i^\Theta = \|c_i^\Theta a_i^\Theta\|_\infty^{1/2} > (2r)^{1/2} \geq |\lambda_i^\Theta - \lambda_i^\Gamma| = \Delta_i. \tag{127}$$

This completes the proof. □

A.4 Section 4

A.4.1 Additional Material

Lemma A.7 (Inverse stability for fixed weight vectors)**.** *Let $N = (d, m, D) \in \mathbb{N}^3$, let $A = [a_1|\dots|a_m]^T \in \mathbb{R}^{m \times d}$ with*

$$\frac{a_i}{\|a_i\|_\infty} \neq \frac{a_j}{\|a_j\|_\infty} \quad \textit{and} \quad (a_i)_{d-1}, (a_i)_d > 0 \tag{128}$$

for all $i \in [m], j \in [m] \setminus \{i\}$, and define

$$\mathcal{N}_N^A := \left\{ \Gamma \in \mathcal{N}_N \colon a_i^\Gamma = \lambda_i a_i \text{ with } \lambda_i \in (0,\infty) \text{ and } \|c_i^\Gamma\|_\infty = \|a_i^\Gamma\|_\infty \text{ for all } i \in [m] \right\}. \tag{129}$$

Then for every $B \in (0,\infty)$ there is $C_B \in (0,\infty)$ such that we have uniform $(C_B, 1/2)$ inverse stability w.r.t. $\|\cdot\|_{L^\infty((-B,B)^d)}$. That is, for all $\Gamma \in \mathcal{N}_N^A$ and $g \in \mathcal{R}(\mathcal{N}_N^A)$ there exists a parametrization $\Phi \in \mathcal{N}_N^A$ with

$$\mathcal{R}(\Phi) = g \quad \text{and} \quad \|\Phi - \Gamma\|_\infty \leq C_B \|g - \mathcal{R}(\Gamma)\|_{L^\infty((-B,B)^d)}^{\frac{1}{2}}. \tag{130}$$

Proof. Note that the non-zero angle between the hyperplanes given by the weight vectors $(a_i)_{i=1}^m$ establishes that the minimal perimeter inside each linear region intersected with $(-B,B)^d$ is lower bounded. As the realization is linear on each region, this implies the existence of a constant $C'_B \in (0,\infty)$, such that for every $\Theta \in \mathcal{N}_N^A$ it holds that

$$|\mathcal{R}(\Theta)|_{W^{1,\infty}} \leq C'_B \|\mathcal{R}(\Theta)\|_{L^\infty((-B,B)^d)}. \tag{131}$$

Now note that for $\mathcal{N}_N^A$ we can get the same uniform $(4, 1/2)$ inverse stability result w.r.t. $|\cdot|_{W^{1,\infty}}$ as in Theorem 3.3 by choosing π to be the identity in (118). Together with (131) this implies the claim. □

Appendix

Curriculum Vitæ **180**

Curriculum Vitæ

Dennis Elbrächter

email: dennis.elbraechter@univie.ac.at

Education

Since 11/2017	**University of Vienna**, Phd student Advisor: Prof. Philipp Grohs Topic: 'Approximation capabilities of ReLU neural networks'
10/2014 – 9/2017	**TU Munich**, Master Mathematik Thesis: 'Recovery of atomic measures from Short Time Fourier Transform measurements'
10/2010 – 9/2014	**TU Munich**, Bachelor Mathematik Thesis: 'An uncertainty relation for joint measurements'

Employment & Internships

11/2017 – 04/2021	**University of Vienna**, Research Assistant
9/2019 – 2/2020	**University of Vienna**, Teaching Assistant Exercises for 'Lineare Algebra für PhysikerInnen'
10/2016 – 2/2017	**TU Munich**, Teaching Assistant Exercises for 'Elements of Harmonic Analysis'
11/2013 – 1/2014	**Zuse Institute Berlin**, Intern Worked on the LP solver SoPlex

Selected Publications

[1] D. Elbrächter, D. Perekrestenko, P. Grohs, and H. Bölcskei, "Deep neural network approximation theory," *Accepted for publication in IEEE Transactions on Information Theory.* Available: https://arxiv.org/abs/1901.02220.

[2] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab, "DNN expression rate analysis of high-dimensional PDEs: Application to option pricing," *Accepted for publication in the special issue "Deep Networks in Approximation Theory" of Constructive Approximation.* Available: https://arxiv.org/abs/1809.07669.

[3] J. Berner, D. Elbrächter, and P. Grohs, “How degenerate is the parametrization of neural networks with the ReLU activation function?,” in *Advances in Neural Information Processing Systems 32*, pp. 7790–7801, 2019.

[4] J. Berner, D. Elbrächter, P. Grohs, and A. Jentzen, “Towards a regularity theory for ReLU networks – chain rule and global error estimates,” in *13th International conference on Sampling Theory and Applications (SampTA)*, pp. 1–5, 2019.

[5] D. Alfke, W. Baines, J. Blechschmidt, M. J. del Razo Sarmina, A. Drory, D. Elbrächter, N. Farchmin, M. Gambara, S. Glas, P. Grohs, P. Hinz, D. Kivaranovic, C. Kümmerle, G. Kutyniok, S. Lunz, J. Macdonald, R. Malthaner, G. Naisat, A. Neufeld, P. C. Petersen, R. Reisenhofer, J.-D. Sheng, L. Thesing, P. Trunschke, J. von Lindheim, D. Weber, and M. Weber, “The oracle of DLphi,” 2019. Available: https://arxiv.org/abs/1901.05744.

Presentations

12/2019 **2019 Conference on Neural Information Processing Systems (NeurIPS)** - Vancouver, Canada
Poster: ’How degenerate is the parametrization of neural networks with the ReLU activation function?’

10/2019 **School on Mathematical and Computational Aspects of Machine Learning, Centro di Ricerca Matematica Ennio De Giorgi** - Pisa, Italy
Oral presentation: ’Inverse Stability of ReLU neural networks’

7/2019 **Signal Processing with Adaptive Sparse Structured Representations (SPARS)** - Toulouse, France
Oral presentation: ’Universal sparsity of deep ReLU networks’

6/2019 **RTG Summer Lectures 2019** - Chicago, USA
Oral presentation: ’Universal sparsity of deep ReLU networks’

6/2019 **NYU, Center for Data Science** - New York, USA
Oral presentation: ’Universal sparsity of deep ReLU networks’

5/2019 **Student workshop on Approximation Theory and Applications** - Graz, Austria
Oral presentation: ’Universal sparsity of deep neural networks’

3/2018 89**th Annual Meeting of the International Association of Applied Mathematics and Mechanics** - Munich, Germany
Oral presentation: ’Using deep neural networks to approximate the solution of the Black-Scholes equation’