universität wien

# DISSERTATION | DOCTORAL THESIS

Titel | Title

Practical Existence Theorems for Function Classes Arising from Applications

verfasst von | submitted by

Andrés Felipe Lerma Pineda

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Doktor der Naturwissenschaften (Dr.rer.nat.)

Wien | Vienna, 2025

| | |
|---|---|
| Studienkennzahl lt. Studienblatt \| Degree programme code as it appears on the student record sheet: | UA 796 605 405 |
| Dissertationsgebiet lt. Studienblatt \| Field of study as it appears on the student record sheet: | Mathematik |
| Betreut von \| Supervisor: | Assoz. Prof. Dr. Philipp Christian Petersen M.Sc. |

# Zusammenfassung

In den letzten Jahren wurde gezeigt, dass die Menge der Realisierungen von neuronalen Netzwerken (NNs) eine breite Klasse von Funktionen approximieren kann, was sie zu einer erfolgreichen Hypothesenklasse in verschiedenen Anwendungsbereichen macht. In dieser Arbeit untersuchen wir, wie auf neuronalen Netzwerken basierende Algorithmen zur Lösung verschiedener Problemstellungen eingesetzt werden können. Insbesondere analysieren wir theoretische Garantien für die Lösung inverser Probleme, Klassifikationsaufgaben und die Schätzung frequenzbegrenzter Funktionen.

In all unseren Problemstellungen interessieren wir uns für die Approximation von Lösungen anhand einer begrenzten Anzahl an Datenproben. Es wird angenommen, dass die Beziehung innerhalb der Daten deterministisch ist und durch eine sogenannte Ground-Truth-Funktion modelliert wird, die vom Algorithmus gelernt werden soll. Wir untersuchen die Approximationseigenschaffen neuronaler Netzwerke für Funktionen mit bestimmten Regularitätseigenschaften. Insbesondere betrachten wir Funktionen, die Lipschitz-stetig sind, $RBV^2$-Regularität aufweisen oder frequenzlokalisiert sind. Auch unstetige Ground-Truth-Funktionen werden berücksichtigt, da sie dennoch von theoretischen Garantien für kontinuierlich $RBV^2$-reguläre Funktionen profitieren können. Darüber hinaus untersuchen wir die Approximation solcher Ground-Truth-Funktionen durch Realisierungen neuronaler Netzwerke mit fester Architektur. Außerdem analysieren wir, wie man ein geeignetes neuronales Netz auswählt, das gut auf unbekannte Daten generalisiert. Jedes Problem wird unter einer spezifischen Verlustfunktion betrachtet.

Wir stoßen auf den sogenannte Fluch-der-Dimensionen-Problem, ein Begriff, der die zunehmende Komplexität bestimmter Algorithmen bei wachsender Dimension beschreibt. In verschiedenen Publikationen wurde beobachtet, dass Methoden auf Basis neuronaler Netzwerke diesen Fluch oft zu überwinden scheinen. In dieser Arbeit analysieren wir zwei der am häufigsten verwendeten Strategien zur Minderung des Fluchs der Dimensionalität. Konkret berücksichtigen wir in Kapitel 4 die Mannigfaltigkeitsannahme und untersuchen in Kapitel 3, wie Regularität im Radon-Raum ausgenutzt werden kann. Unter diesen Annahmen leiten wir Approximations- und Schätzraten her.

Im Gegensatz zum Rahmen der universellen Approximationssätze sind unsere Resultate im Rahmen des kürzlich eingeführten Practical Existence Theorem (PET) formuliert. Innerhalb dieses Rahmens liefern wir theoretische Garantien für die Existenz einer Klasse neuronaler Netzwerke, die jedes der betrachteten Probleme mit einer begrenzten Anzahl an Parametern löst. Zudem geben wir untere Schranken für die Stichprobenkomplexität und obere Schranken für den Generalisierungsfehler an. Die Auswahl eines geeigneten neuronalen Netzwerks erfolgt über ein Optimierungsverfahren, typischerweise durch Minimierung eines empirischen Risikofunktionals mit einer problemspezifischen Verlustfunktion.

# Abstract

In recent years, the set of realizations of neural networks (NNs) has been shown to approximate a wide range of functions, making it a successful hypothesis class across various fields. In this thesis, we investigate how neural-network-based algorithms can be implemented to solve a variety of problems. Specifically, we study theoretical guarantees for the solution of inverse problems, classification tasks, and the estimation of frequency-limited functions.

In all of our problems, we are interested in approximating solutions from a limited number of data samples. The relationship within the data is assumed to be deterministic and modeled by a ground-truth function to be learned by the algorithm. We study the approximation capabilities of neural networks (NNs) for functions satisfying certain types of regularity. In particular, we consider functions that are Lipschitz continuous, exhibit $RBV^2$-regularity, or are frequency-localized. We also include discontinuous ground-truth functions, which can still benefit from theoretical guarantees derived for $RBV^2$-regular functions. Besides, we study the problem of approximating such ground-truth functions using realizations of NNs with fixed architectures. Furthermore, we study how to select an appropriate NN that generalizes well to unseen data. Each problem is considered under a different loss function.

We encounter the so-called curse of dimensionality, a term that refers to the increasing complexity of certain algorithms as the input dimension grows. It has been observed in various publications that neural-network-based methods often appear to overcome this curse. In this thesis, we study two of the most commonly employed strategies to mitigate the curse of dimensionality. Specifically, in Chapter 4, we consider the manifold assumption, and in Chapter 3, we investigate how regularity in the Radon domain can be exploited. Under these assumptions, we derive approximation and estimation rates.

In contrast to the setting of universal approximation theorems, our results are formulated within the recently introduced framework of the Practical Existence Theorem (PET). Within this framework, we provide theoretical guarantees for the existence of a class of neural networks that solves each problem, using a limited number of parameters. We also establish lower bounds on sample complexities and upper bounds on generalization errors. The choice of an appropriate neural network is carried out via an optimization algorithm, typically by minimizing an empirical risk functional with a problem-specific loss function.

# Contents

# Chapter 1

# Introduction

In recent years, computers have become capable of performing tasks that are usually labeled with the adjective of *intelligent* [27]. Although the concept of *intelligence* has not been properly defined [46], it is usually connected to processes such as *learning* new information or *reasoning* from a set of statements [168]. Building new computational algorithms with this property is the main goal of the field of *artificial intelligence (AI)* (see [141, Section 1.1]). This area of research has produced state-of-the art techniques in a vast number of disciplines for the solution of tasks in fields such as natural language processing [163, 13], computer vision [74], decision-making [47] and pattern recognition [110].

Within the area of AI, the subfield of *machine learning (ML)* stands for its astonishing achievements [72], with multiple applications that permeate our daily life, ranging from spam detection [40], stock price prediction [94] and face recognition [147]. In contrast to other areas of AI, ML techniques focus on analyzing and extracting information from a set of data, typically referred to as a sample set [19]. After obtaining information from the sample set, these *data-driven algorithms* output a model which can be used to make new predictions for new data. We refer to the process above as *learning* which can be improved depending on the quality and quantity of the data [28].

In this thesis, we investigate the performance of a particular class of ML algorithms for the solution of diverse problems. Specifically, we restrict ourselves to the subfield of *deep learning (DL)*. One of the fundamental notions of this area is the concept of *artificial neuron* which was originally intended to model the functioning of a biological neuron [98]. Artificial neurons can be arranged in different ways to give rise to the so-called *neural networks (NNs)* (see [127, Chapter 2]). Broadly speaking, an NN is a sequence of matrices and vectors as well as a nonlinear function that induces a function between Euclidean spaces [132], usually referred to as the *realization* of the NN. The entries of the matrices and vectors of each NN are called the *parameters* of the NN. The set of realizations of NNs constitute the foundation of the so-called *neural-network-based* algorithms, which use the set of realizations of NNs as a *hypothesis set* to approximate the underlying relation of the data [7]. Through this thesis, we follow this approach.

One of the first fundamental results of the theory of NNs establishes that the set of realizations of NNs is sufficiently expressive to approximate every continuous function defined on

a compact set in the uniform norm to any given accuracy. This fact is commonly formulated in the form of the so-called *universal approximation theorems*, initially proposed by Hornik [63] and Cybenko [41], and further expanded in later publications (see for example [36], [70]). Such a property makes the set of NNs into a suitable hypothesis set for the relation between quantities in a vast number of phenomena from diverse fields of studies such as partial differential equations [152, 135, 59], dynamical systems [26, 95], or inverse problems [10].

Although the underlying relation of the data may not be deterministic, we assume in this thesis that it can be described by a function between two suitable sets. We call this function the *ground-truth* function to be learned by the algorithm. Universal approximation results guarantee that we can always design a neural-network-based algorithm to learn a continuous ground-truth function. However, such theorems do not provide further insight into how the number of parameters in the neural network should be selected. We address this issue in all the methods proposed in this thesis.

Another common objection to the universal approximation theorems comes from the lack of clarity on how to select the NN. In practice, we use an optimization algorithm to minimize an objective function that depends on the dataset [6]. One parameter that we would like to optimize is the number of samples required for the algorithm to learn a ground-truth function with a certain accuracy. We call this number the *sample complexity* of the problem. In many applications, the number of data points can be severely restricted. In the problem of computerized tomography, for example, collecting a large amount of samples requires long exposure time to some kind of radiation [66]. Thus, since the collection of the data may cause irreversible damage, such an exposure time needs to be minimized (and therefore the number of samples to be collected). Thus, we wish to find lower sample complexities that guarantee that our algorithm performs well on new data. This is the problem of *generalization* of the algorithm. We address the question of sample complexity and the problem of generalization in each chapter of this thesis. In particular, we present theoretical guarantees for three specific problems, which will be described in the upcoming sections.

In contrast to universal approximation theorems, the so-called practical existence theorems (PTE) have emerged as a new framework for the study of neural-network-based methods. This new paradigm was initially formulated in [3] and further developed in [1]. Depending on the problem to be solved, a PET guarantees the existence of a class of admissible NNs from which we can select a convenient NN that approximates its solution via an optimization algorithm [1]. Moreover, PETs provide upper bounds for the number of samples needed to find a solution to the problem. Several techniques have been at the core of PETs such as the approximation of orthogonal polynomials for basis expansions in Hilbert spaces [43, 44, 119], physics-informed NNs [25], or convolutional auto-encoders for reduced-order modeling [52].

The PET setting provides a summarizing framework addressing all of the issues discussed above. In this thesis, we present a study of different neural-network-based algorithms for a variety of problems that leverages this newly introduced setting. Specifically, we prove for each case, that there is an NN that can be learned from a data set by minimizing a functional depending on this data set and provide lower bounds for the sample complexity

and the generalization error. The results that we discuss in this thesis can be found in the papers [92, 93, 115], but we have included here minor modifications, further explanations, and additional clarifications. We explain each problem to be considered in the following sections.

In the remainder of this chapter, we provide a brief overview of the fundamental topics that can be found in this thesis. In Section 1.1, we introduce the concept of NN and present some of the most relevant literature on the subject of approximation and estimation functions by neural-network-based methods. Section 1.2 describes the problem of classification that we consider in this thesis and discuss some of the classical approaches to tackle it. In Section 1.3, we present an informal discussion on inverse problems and recall some of the most relevant references for the study of their solutions via deep-learning-based methods. Finally, in Section 1.4, we set out informally the problem of reconstructing frequency-localized functions, emphasize its importance, and review some connecting literature.

## 1.1 Function Approximation and Estimation

To learn a ground-truth function $f : X \to Y$, we consider our data to be represented in the form of pairs or a 2-tuple $(x, y)$ where $x \in X$, $y = f(x) \in Y$ and $X, Y$ are two suitable spaces. In particular, we approximate functions belonging to a given set $\mathcal{G} \subset \mathcal{F}$, where $\mathcal{F}$ is a Banach space, where the approximating elements are in a different class $\mathcal{H} \subset \mathcal{F}$. In our case, the set $\mathcal{H}$ corresponds to a set of realizations of NNs with fixed architecture. In particular, we are interested in whether, for each function $f \in \mathcal{G}$ and for each $\epsilon > 0$ there is a function $g \in \mathcal{H}$ such that $\|f - g\| < \epsilon$. Note that this class $\mathcal{G}$ may be the whole Banach space $\mathcal{F}$. Moreover, we want to learn the function $f$ when only a limited number of data points $(x_i, f(x_i))_{i=1}^m$ is known, that is, we aim to find a function $h \in \mathcal{H}$ such that the difference between $f$ and $h$ at these values $x_i$ of its domain $i \in [m]$ is small. Moreover, we want the selected function $h$ to generalize well, i.e, the error between the two functions $h$ and $f$ is small on a different choice of values $\tilde{x}_i$ on its domain.

Preliminary studies have shown that NNs are effective in learning functions with different levels of regularity, for example, under the assumption of Lipschitz continuity [75], Hölder continuity [143], smoothness [133] or analyticity [101]. In Chapter 2, we discuss some of the most relevant achievements in this area, which will be used to demonstrate our results in the following chapters. We would like to mention that our work has made some theoretical contributions for the case of continuous functions as well. Specifically, in Chapter 5, we present an alternative result for the approximation of square-integrable continuous functions by realization of NNs.

Most of the aforementioned results lead to an excessive number of NN parameters when the input dimension is large. This exponential dependency is an undesirable feature that is known as the *curse of dimensionality* (see [82]). Numerical experiments have shown that in practice, one may not require to set apriori such a large number of parameters to approximate a function with high accuracy [84, 151]. This observation is usually referred to as the gap between theory and practice [3]. Inspired by this intriguing discrepancy, functions defined

on low-dimensional sets within Euclidean spaces have lately received greater attention, and the problem of approximating such functions via NNs has been extensively investigated. Under the choice of smooth activation functions, the existence of approximating realizations of NNs for functions defined on manifolds is proved in [37, 102]. The approximation rates by neural networks for $C^k$-functions, $k \in \mathbb{N}$ defined on manifolds are analyzed in [38, 145]. Analogous results for $\alpha$-Hölder continuous functions, $\alpha > 0$, can be found in [32, 80, 142, 109]. Under this low-dimensional setting, we demonstrate in this thesis that for each Lipschitz continuous function defined on a low-dimensional manifold immersed in a highly-dimensional Euclidean space which satisfies some technical assumptions, there is a robust-to-noise NN, the realization of which approximates the given function, for an arbitrary uniform accuracy $\epsilon > 0$. Moreover, we show that the number of parameters of this NN can be controlled as well. Our main contributions to this issue can be found in Chapter 4.

Other less known function properties can be exploited to infer approximation and estimation rates in neural-network-based methods. For example, dimension-independent approximation and estimation rates have been established for functions with high regularity in their Fourier domain [14] or Radon domain [123]. These results have been used to obtain approximation and estimation rates for certain types of discontinuous functions (see [128, 130]). We revisited these ideas in Chapter 3, where we study in detail how the $RBV^2$-regularity of a function determines the approximation and estimation rates when the set of realizations of NNs is used as hypothesis set.

## 1.2 Classification Tasks

Some of the most prominent examples of the power of NNs emerge from image classification tasks [90, 84, 83]. Motivated by this fact, we study in this thesis the ability of deep-learning-based methods to perform such classification tasks.

The ground-truth function for several classification tasks can be modeled using a discontinuous function called the *classifier*. In simple terms, a classifier is a function that assigns a numerical *label* to each element or point in a given domain [128]. For simplicity, we assume the domain of these functions to be a subset of a Euclidean space. Each coordinate of a certain point represents a different *feature* of the element, which we assume here to be a real number. In addition, we assume the set of labels to be finite which leads us to a finite partition of the domain into different classes, numbered in accordance with the label set.

Several deep-learning algorithms for the identification of the ground-truth function have already been implemented with astonishing results. One of the first assumptions imposed on the model considered the case of only two partition sets $C_+$ and $C_-$ having positive set distance [169]. Under such conditions, the authors obtained some approximation and estimation bounds by shallow NNs for the classification problem. In [67, 68, 132], it is proved that the regularity of the boundary partition set determines the approximation and estimation rates for classifiers using neural networks. Under the assumption of differentiability up to an order $k$, these rates are subject to the curse of dimensionality. Indeed, this idea has inspired our own work, and some of the ideas of [30] are similar to those we discussed here.

Following this approach, in [30], a new condition is introduced. Therein, the boundaries of each partition class are assumed to be locally parametrized by Barron functions. This new assumption leads to rates that are not subject to the curse of dimensionality.

We continue this line of thought by studying classification problems related to a different Banach space. This space is called the space of $RBV^2-$functions, which arises in the solution of an optimization problem [122]. In Chapter 3, we present a brief introduction of this function space and list some of its properties. This space contains highly smooth functions as well as functions with very low regularity. In particular, it contains the space of Sobolev functions, where the regularity order is sufficiently large. The approximation and estimation rates for this function space using neural-network-based methods that do not grow exponentially on the dimension have been derived in [123]. We complement these results and extend them for the case of classifiers with decision boundaries in the space of $RBV^2$ as they play an important role in our results on classification. Our main result is summarized in the form of the Practical Existence Theorem 3.3.2. Most of the approximation and estimation results can be found in the paper:

Paper Reference

**A. Lerma, P. Petersen, S. Frieder, T. Lukasiewicz**, *Dimension-independent learning rates for high-dimensional classification problems*, 2024 , **in Arxiv**.

Our theory is built upon previously established rates for the problem of approximating $RBV^2$-functions by the realization of NNs. P. Petersen contributed to conceptualization and supervision; S. Frieder and T. Lukasiewicz were involved in the numerical experimentation; A. Lerma led the technical writing and carried out most of the experimentation. In particular, we focus on the results of [123] and [149]. We noticed that some of the statements in these publications were not rigorously demonstrated. However, recent developments in [149] filled these gaps, allowing better formal demonstrations. In this thesis, we provide alternative proofs for such results when required; particularly, we present a modified version of [123, Theorem 8] where the existence of an NN approximating each $RBV^2$-function is stated. An additional reason to do so is that the question of upper bounds for the weights and biases of the approximating NN is not addressed in these publications. We solve this issue as well, as this is a crucial step in the auxiliary results for our PET on classification.

## 1.3 Inverse Problems

Finding the solution to an inverse problem is an important goal in many fields of mathematics and applied sciences. Broadly speaking, each scenario where the cause of a process needs to be identified can be labeled as an inverse problem. These causes correspond to physical quantities or parameters to be identified in a process and can be considered to be functions defined between Euclidean spaces. Due to limitations in time, cost, or measurability, these causes are often impossible to obtain in practical settings [10]. Frequently, only partial knowledge of the effects, in our case in the form of samples, is available. The data samples

are usually corrupted by noise, which makes the inverse problem even more challenging to solve (see [79, Chapter 1]).

There is no universal definition of what an inverse problem is [10, Section 2.1]. However, there are convenient formulations in terms of operators between two topological spaces. Solving an inverse problem corresponds to reconstructing the solution of a functional equation. The operator involved in an inverse problem is usually referred to as the *forward operator*, which we denote by $F$. The computation of the solution to an inverse problem can be performed via the inverse operator of $F$ in case this inverse operator exists. For many interesting forward operators, a closed form of the inverse operator $F^{-1}$ can be impractical, if not impossible, to obtain (assuming its existence) [10]. Indeed, a well-established fact in the inverse problem community is that many problems lack *stability* when the data are known only with a certain degree of uncertainty (see [79]). As a remedy to the instability of several inverse problems, regularization techniques have been proposed. A detailed survey of regularization techniques can be found in [79, Chapter 2].

Inspired by the overwhelming success in estimating ground-truth functions from data, deep-learning-based methods have been applied for the reconstruction of inverse problems solutions in various domains [10, 66]. Some initial approaches investigated the ability of NNs to solve inverse problems in imaging via data-driven algorithms, which is widely discussed in [117]. Although initial studies were focused mostly on the numerical capabilities of NNs, this line of research is currently better established, and many observed phenomena are already well-understood. Most of the widely used data-driven techniques for solving inverse problems fall into one of the next four categories and are designed in accordance with some previous knowledge on the forward operator $F$: known from the beginning [56, 158], during training [42, 23, 164], partially known [9, 176], or never known [175]. One limitation of several neural-network-based methods for the solution of inverse problems comes from their instability, as demonstrated in [8, 65]. In [54], a comprehensive analysis explaining why these methods are intrinsically unstable is presented, and the difficulties in finding remedies for these instabilities are also discussed.

Building on preliminary results, we continue to study neural-network-based methods for the solution to inverse problems. In Chapter 4, we present a practical existence theorem for the solution of inverse problems under the assumption that the operator domain is a low-dimensional manifold within a high-dimensional Euclidean space. Our main contribution is to prove that the constructed neural network is stable to noise, as required by the class of inverse problems. Our contribution on this topic has already been presented in our paper

Paper Reference

**A. Lerma, P. Petersen**, *Deep neural networks can stably solve high-dimensional, noisy, nonlinear inverse problems* , **in Analysis and Applications 21.01** (2023): 49-91.

P. Petersen contributed to conceptualization and supervision; A. Lerma led the technical writing and carried out most of the experimentation. Our approach allows for the recon-

struction of the inverse problem solution as long as the domain of the forward operator is a low-dimensional manifold that we call the *intrinsic domain*. In contrast, the constructed NN is defined in Euclidean spaces, which we refer to as the *ambient domain*. This NN is proved to be *robust-to-noise* under certain conditions on the noise distribution, which is in fact our main contribution to this topic. As inverse problems are commonly defined on infinite-dimensional spaces, we also discuss how our approach can be implemented in practice. Similarly to other chapters in this thesis, we study the problem of estimating a solution to an inverse problem from a limited number of samples and formulate our results in the form of a PET. We include some numerical simulations to assess the performance of our method for some inverse problems of interest.

## 1.4 Frequency-localized Functions

Reconstruction of signals from samples is a frequent problem in different fields, for which there is a continuous need to develop more accurate algorithms. One standard approach to tackle this problem is to assume that the signal belongs to the set of *frequency-localized* functions, i.e., the functions where most of its energy is concentrated in a compact set of its Fourier domain. A particular subset of the space of frequency-localized functions is the class of *band-limited functions*. For the latter class of functions, there is a vast number of important applications ranging from signal estimation [150], radar imaging [33], radio transmission [29], and compressed sensing [161].

In this thesis, we study the approximation of multi-dimensional frequency-localized functions via deep-learning-based methods. The main ingredients for the PET have been presented in the following manuscript:

Paper Reference

**M. Neuman, A. Lerma, J. Bramburger, S. Brugiapaglia**, *Reconstruction of frequency-localized functions from pointwise samples via least squares and deep learning*, 2025, **in Arxiv**.

M. Neumann developed the main arguments, led most of the final writing, and supervised the project. J. Bramburger and S. Brugliapaglia contributed to the technical writing. A. Lerma was responsible for the development of several technical results and for the numerical experimentation.

We present the results of the aforementioned manuscript in Chapter 5 which constitute a detailed study for the reconstruction of frequency-localized functions by NNs. Following the context of this thesis, our findings are formulated here in the form of a PET. Furthermore, we compare the performance of our deep-learning-based methods with the performance of one of the most widespread methods in approximation of functions: the least squares method.

Several of our results build upon the properties of the Slepian basis [153, 154, 156], which is an orthogonal basis for the space of $L^2$-functions defined on $\mathbb{R}$. This basis presents maximal energy concentration within the interval $[-1, 1]$ [105, 150, 167, 170], making these functions

suitable for the estimation of frequency-localized and bandlimited functions from samples. In contrast to Chebyshev- or Lagrange-based interpolation methods, the spatial resolution of the Slepian basis is more uniform, which allows us to support longer stable time steps. This feature gives the Slepian basis a distinct advantage in applications such as weather prediction [22].

To obtain an upper bound for the sample complexity, we leverage previously established results from the field of *least-squares approximation*(see [2, Chapter 5]). In particular, we consider the approximation of smooth functions via NNs. To this end, we approximate the Slepian basis with linear expansions which are approximated by NNs as well and can be explicitly constructed. Thus, we need to run an algorithm that determines the parameters of the last layer. This idea has previously been studied in the field of *transfer learning* [121], where some layers of a previously trained model are fine-tuned on new data to solve a similar task. In particular, the method of fine-tuning (or retraining) the last NN layer has been shown to improve the accuracy of neural networks when dealing with spurious features in the data [78, 157].

# Chapter 2

# The Mathematical Theory of Neural Networks

In this chapter, we introduce all the fundamental notions necessary for the following chapters. In particular, we formally define what a *neural network (NN)* is. Before we do this, we take a detour to introduce the notion of *perceptron* which historically appeared first and paved the way for the later development of NNs [99]. A *perceptron* has a simple structure: it is defined for all elements in $\mathbb{R}^d$, $d \in \mathbb{N}$, and is associated with a vector $\mathbf{w} \in \mathbb{R}^d$ and a scalar $b \in \mathbb{R}$. The concept of perceptron was originally designed to model the functioning of human neurons [139].

**Definition 2.0.1.** *[7, Discussed in Section 1.2] Let $d \in \mathbb{N}$. Let $\mathbb{1}_{\mathbb{R}^+} : \mathbb{R} \to \mathbb{R}$ be the* Heaviside function*, which is the indicator function of the set $(0, \infty)$. A function of the form*

$$\mathbb{R}^d \ni \mathbf{x} \to \mathbb{1}_{\mathbb{R}^+}(\mathbf{w}^T\mathbf{x} + b),$$

*where $b \in \mathbb{R}$ and $\mathbf{w} \in \mathbb{R}^d$ is called a* perceptron*.*

Due to the presence of the Heaviside function, the perceptron outputs either the value one or zero, depending on the sign of the quantity $\mathbf{w}^T\mathbf{x} + b$. We say that the perceptron *fires* when $\mathbf{w}^T\mathbf{x} + b > 0$, that is, when the output is one. The Heaviside function is an example of a broader class of functions known as *activation functions*, which introduce a nonlinearity into the perceptron. We can generalize this notion by allowing the use of other activation functions.

**Definition 2.0.2.** *[7, Discussed in Section 1.2] Let $d \in \mathbb{N}$ and $\varrho : \mathbb{R} \to \mathbb{R}$ be a function. Let $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. A* neuron *is a function $\Phi$ of the form*

$$\mathbb{R}^d \ni \mathbf{x} \to \Phi(\mathbf{x}) = \varrho(\mathbf{w}^T\mathbf{x} + b),$$

*where $b \in \mathbb{R}$ is called the bias, $\mathbf{w} \in \mathbb{R}^d$ is the weight vector and $\varrho$ is called the activation function.*

We can combine different neurons to form what we call a neural network. Moreover, we can arrange different neurons so that the output is high-dimensional, as we see below. The way we combine these neurons and the operations we allow among them gives rise to different types of neural networks [139]. In this work, our main focus is on *feed-forward neural networks* which we simply call *neural networks (NN).*

**Definition 2.0.3** ([130, Definition 1.2])**.** *Let $d, L \in \mathbb{N}$. A* neural network (NN) with input dimension $d$ and $L$ layers *is a sequence of matrix-vector tuples*

$$\Phi = \big((A_1, b_1), (A_2, b_2), \dots, (A_L, b_L)\big),$$

*where $N_0 \coloneqq d$ and $N_1, \dots, N_L \in \mathbb{N}$, and where $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{N_\ell}$ for $\ell = 1, ..., L$.*

*For an NN $\Phi$ and an activation function $\varrho : \mathbb{R} \to \mathbb{R}$, we define the associated* realization of $\Phi$ *as*

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

*where the output $x_L \in \mathbb{R}^{N_L}$ results from*

$$\begin{aligned} x_0 &\coloneqq x, \\ x_\ell &\coloneqq \varrho(A_\ell\, x_{\ell-1} + b_\ell) \quad \text{for } \ell = 1, \dots, L-1, \\ x_L &\coloneqq A_L\, x_{L-1} + b_L. \end{aligned} \tag{2.0.1}$$

*Here $\varrho$ is understood to act component-wise on vector-valued inputs, i.e., for $y = (y^1, \dots, y^m) \in \mathbb{R}^m$, $\varrho(y) \coloneqq (\varrho(y^1), \dots, \varrho(y^m))$. We call $N(\Phi) \coloneqq d + \sum_{j=1}^{L} N_j$ the* number of neurons of $\Phi$*, $L(\Phi) \coloneqq L$ the* number of layers *or* depth*, $W_j(\Phi) \coloneqq \|A_j\|_0 + \|b_j\|_0$ the* number of weights in the $j$-th layer*, and $W(\Phi) \coloneqq \sum_{j=1}^{L} W_j(\Phi)$ the* number of weights of $\Phi$*, also referred to as the* size *of $\Phi$. The number of weights in the first layer is also denoted by $W_{\mathrm{fi}}(\Phi)$, the number of weights in the last layer by $W_{\mathrm{la}}(\Phi)$. We refer to $N_L$ as the* dimension of the output layer of $\Phi$*. Lastly, we refer to $(d, N_1, \dots, N_L)$ as the* architecture of $\Phi$*.*

In Figure 2.1, we illustrate a feed-forward NN with number of layers $L = 4$, input dimension $d = 3$ and output dimension $N_L = 1$. The only activation function considered in this thesis is the well-known ReLU, denoted as $\varrho(x) = max\{x, 0\}$. Due to the simplicity of its formula, among many other properties, this activation function has been used in several publications [108]. However, other activation functions worth mentioning are the sigmoid [85] and the Tanh [96], which are illustrated in Figure 2.2. The choice of the activation function depends on the characteristics of the problem to be solved, as analyzed in [60].

In this thesis, we investigate how the set of realizations of NNs performs as a hypothesis set to identify the ground-truth function in various applications. In our proposed approaches, we fix the number of layers $L$ as well as the weights $W$ and the maximal number of neurons $N$ per layer to carry out our study. We now introduce the set of NNs under this set of conditions.

**Definition 2.0.4.** *[129] Let $d, q \in \mathbb{N}_{\geq 2}$, $N, W, L \in \mathbb{N}$, and $B > 0$. The* set of realizations of ReLU NNs with $L$ layers, at most $N$ neurons per layer, at most $W$ non-zero weights and weights with magnitudes bounded by $B$ *is defined as*

$$\mathcal{NN}(d, q, L, N, W, B) \coloneqq \{f : \mathbb{R}^d \to \mathbb{R}^q : \exists \text{ an NN } \Phi^f \text{ s.t. } \mathrm{R}(\Phi^f)(x) = f(x)\, x \in [0,1]^d\}.$$

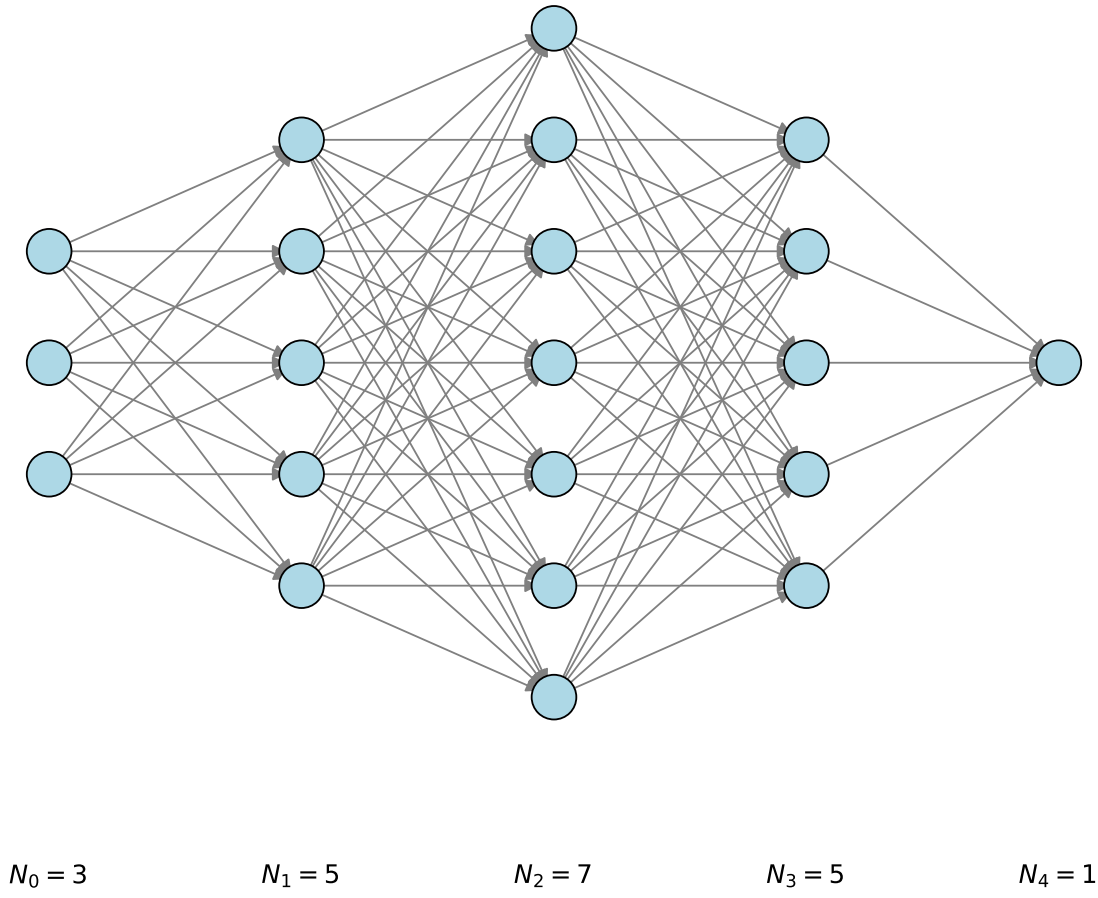


Figure 2.1: Illustration of a 4-layer feed-forward NN.

*Moreover, we set*

$$\mathcal{NN}_*(d, 1, L, N, W, B) \coloneqq \big\{ f \in \mathcal{NN}(d, 1, L, N, W, B) \quad : \quad 0 \leq f(x) \leq 1 \quad \textit{for all } x \in [0,1]^d \big\}.$$

In particular, we consider in Chapter 3, NNs with only two-layers which are referred to as *shallow NN*. We can express the realization of a shallow NN $\Phi$ as

$$\mathrm{R}(\Phi)(\mathbf{x}) = \sum_{i=1}^{m} a_i \varrho(\mathbf{w}_i^T \mathbf{x} + b_i),$$

where $m \in \mathbb{N}$ is the number of neurons in the hidden layer, $a_i, b_i \in \mathbb{R}$ and $\mathbf{w}_i \in \mathbb{R}^d$. Three-layer NNs are considered for classification problems in Chapter 3 and the realizations of NNs with deeper architectures are considered as hypothesis sets in the Chapters 4 and 5. The PETs that we discuss in the following chapters base their proofs on the existence of NN sets with a limited number of parameters, although not explicitly mentioned in their formulation.

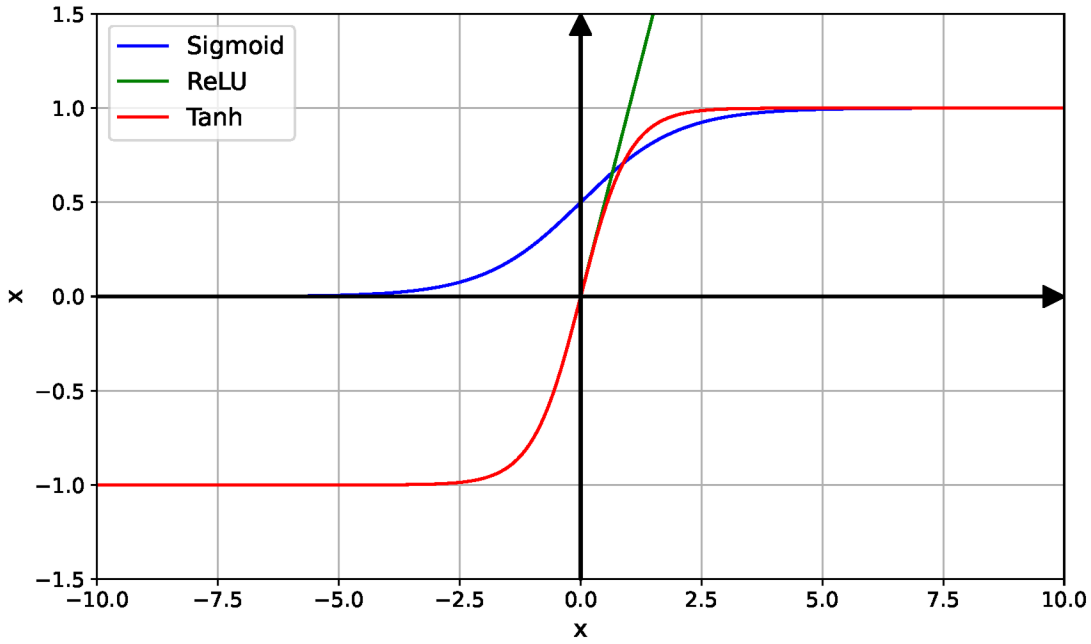


Figure 2.2: Some of the most used activation functions: Sigmoid, ReLU, and Tanh.

Two or more NNs can be combined to construct new NNs. By doing so, we provide the set of realizations of NNs with an additional algebraic structure which helps us in the constructions for new NNs in the upcoming sections. We now present the operations defined over the sets of neural networks that we use in most of our results.

**Definition 2.0.5.** *[131, Definition 2.2] Let $L_1, L_2 \in \mathbb{N}$. Let*

$$\Phi^1 = ((A_1^1, b_1^1), \ldots, (A_{L_1}^1, b_{L_1}^1))$$

*and*

$$\Phi^2 = ((A_1^2, b_1^2), \ldots, (A_{L_2}^2, b_{L_2}^2))$$

*be two NNs with the respective number of layers $L_1$ and $L_2$ such that $N_0^1 = N_{L_2}^2$, i.e., the input layer of $\Phi^1$ has the same dimension as that of the output layer of $\Phi^2$. Then $\Phi^1 \odot \Phi^2$ is an NN with $L_1 + L_2 - 1$ layers such that*

$$\Phi^1 \odot \Phi^2 := ((A_1^2, b_1^2), \ldots, (A_{L_2-1}^2, b_{L_2-1}^2), (A_1^1 A_{L_2}^2, A_1^1 b_{L_2}^2 + b_1^1), (A_2^1, b_2^1), \ldots, (A_{L_1}^1, b_{L_1}^1)),$$

*called the concatenation of $\Phi^1$ and $\Phi^2$. Moreover, $\mathrm{R}(\Phi^1 \odot \Phi^2) = \mathrm{R}(\Phi^1) \circ \mathrm{R}(\Phi^2)$.*

The operation of concatenation allows us to construct a new NN from a pair of NN, the realization of which is the composition of the respective realizations. Now, we introduce an operation that allows us to define addition on the set of NNs.

**Definition 2.0.6.** *[131, Definition 2.7] Let $L, d \in \mathbb{N}$. Let*

$$\Phi^1 = ((A_1^1, b_1^1), \dots, (A_L^1, b_L^1))$$

*and*

$$\Phi^2 = ((A_1^2, b_1^2), \dots, (A_L^2, b_L^2))$$

*be two NNs with $L$ layers, each having $d$-dimensional input. Then $\mathrm{P}(\Phi^1, \Phi^2)$ is an NN with $L$ layers such that*

$$\mathrm{P}(\Phi^1, \Phi^2) := ((\tilde{A}_1, \tilde{b}_1), \dots, (\tilde{A}_2, \tilde{b}_2))$$

*where*

$$\widetilde{A}_1 := \begin{pmatrix} A_1^1 \\ A_1^2 \end{pmatrix}, \quad \widetilde{b}_1 := \begin{pmatrix} b_1^1 \\ b_1^2 \end{pmatrix}, \quad \text{and} \quad \widetilde{A}_l := \begin{pmatrix} A_l^1 & 0 \\ 0 & A_l^2 \end{pmatrix}, \quad \widetilde{b}_l := \begin{pmatrix} b_l^1 \\ b_l^2 \end{pmatrix}, \quad \text{for} \quad l = 1, \dots, L,$$

*called the parallelization of $\Phi^1$ and $\Phi^2$. Moreover, $\mathrm{R}(\mathrm{P}(\Phi^1, \Phi^2)) = (\mathrm{R}(\Phi^1), \mathrm{R}(\Phi^2))$.*

**Remark 2.0.7.** *It follows directly from the given definitions that*

$$L(\Phi^1 \odot \Phi^2) \leq L(\Phi^1) + L(\Phi^2),$$
$$W(\Phi^1 \odot \Phi^2) \leq W(\Phi^1) + W(\Phi^2),$$

*and that*

$$L(\mathrm{P}(\Phi^1, \Phi^2)) = L(\Phi^1) = L(\Phi^2),$$
$$W(\mathrm{P}(\Phi^1, \Phi^2)) = W(\Phi^1) + W(\Phi^2).$$

*Concatenation and parallelization can be extended to multiple neural networks, such that*

$$\Phi^1 \odot \Phi^2 \odot \Phi^3 = (\Phi^1 \odot \Phi^2) \odot \Phi^3 = \Phi^1 \odot (\Phi^2 \odot \Phi^3),$$
$$\mathrm{P}(\Phi^1, \Phi^2, \Phi^3) = \mathrm{P}(\mathrm{P}(\Phi^1, \Phi^2), \Phi^3) = \mathrm{P}(\Phi^1, \mathrm{P}(\Phi^2, \Phi^3)).$$

Now, we state that the identity map of each space $\mathbb{R}^d$ can be realized by an NN. In the following chapters, we use the result above to realize a variety of functions.

**Proposition 2.0.8.** *[118, Proposition 2.4] Let $d, L \in \mathbb{N}$. Let $I : \mathbb{R}^d \to \mathbb{R}^d$ denote the identity map $I(x) = x$ for all $x \in \mathbb{R}^d$. Then there exists an NN $\Phi_I$ with input dimension $d$ and $L$ layers, such that $\mathrm{R}(\Phi_I)(x) = I(x) = x$ for all $x \in \mathbb{R}^d$ and $W(\Phi_I) \leq 2dL$.*

The following proposition states that the linear combination of realization of NNs is also realized by an NN.

**Proposition 2.0.9.** *[129, Lemma 5] For $d, K \in \mathbb{N}$, let $\{\Phi_j\}_{j=1}^K$ be a collection of NNs where*

$$\Phi^j = ((A_1^j, b_1^j), \dots, (A_{L_j}^j, b_L^j))$$

*where* $A_i^j \in \mathbb{R}^{d_i \times d_{i+1}}$ *and assume* $d_{L_1} = d_{L_2} = \cdots = d_{L_K}$. *For a set of scalars* $\{\alpha_j\}_{j=1}^K$, *there is an NN* $\Phi_\varrho$ *such that*

$$\mathrm{R}(\Phi_\varrho) = \sum_{j=1}^{K} \alpha_j \mathrm{R}(\Phi_j),$$

*and that*

$$W(\Phi_\Sigma) \leq 2\sum_{j=1}^{K} W(\Phi_j) + 2\sum_{j=1}^{K}(L_{max} - L_j)d_{L_j}^j \quad \text{and} \quad L(\Phi_\Sigma) = \max_{1\leq j\leq K} L(\Phi_j).$$

*where* $L_{max} := \max_{1\leq j\leq K} L(\Phi_j)$.

## 2.1 Universal Approximation Theorems

We have mentioned in the introduction that every continuous function can be approximated by the realization of an NN with arbitrary precision. In this section, we present some results that support this assertion. It is worth mentioning that this is a well-known fact within the theory of NNs, and it can be generalized to a larger class of activation functions. However, to maintain consistency with our discussion, we have adapted these results to the case of ReLU activation function.

After a simple inspection, we can observe that the realization of a ReLU NN is a piecewise linear function. In real-life applications, not all causal relations between two sets are of this type. Although this allows for the modeling of many problems, it also implies that not every function can be realized by a ReLU NN. In particular, smooth simple functions as the quadratic and the cubic function are not realized by NNs. The universal approximation theorems give a theoretical background for the approximability of a continuous function with certain limitations. We start by defining in which sense we study this approximability.

**Definition 2.1.1.** *[64, Adapted from Definition 2.7] Let* $d \in \mathbb{N}$. *A sequence of functions* $f_n : \mathbb{R}^n \to \mathbb{R}$ *is said to* converges compactly *to the function* $f : \mathbb{R}^d \to \mathbb{R}$, *if for every compact* $K \subset \mathbb{R}^d$ *follows that*

$$\lim_{n\to\infty} \sup_{x\in K} |f_n(\mathbf{x}) - f(\mathbf{x})| = 0.$$

Now, we state what a *universal approximator* is.

**Definition 2.1.2.** *[64] Let* $d \in \mathbb{N}$. *A set of functions* $\mathcal{H}$ *from* $\mathbb{R}^d$ *to* $\mathbb{R}$ *is said to be a universal approximator of* $C^0(\mathbb{R}^d)$ *if for* $\epsilon > 0$, *every compact set* $K \subset \mathbb{R}^d$ *and every* $f \in C^0(\mathbb{R}^d)$, *there is a function* $g \in \mathcal{H}$ *such that*

$$\sup_{\mathbf{x}\in K} |f(\mathbf{x}) - g(\mathbf{x})| < \epsilon.$$

As a side note, we observe that if the activation function is linear or polynomial, the set of all realizations cannot be universal. However, we know that there are several activation functions whose set of realizations is actually a universal approximator of $C^0(\mathbb{R}^d)$. As mentioned above, we state this result for the case of the ReLU activation function.

**Theorem 2.1.3.** *[64, Adapted from Theorem 2.1] Let $d \in \mathbb{N}$ and $\varrho$ the ReLU activation function. Then, the set $\mathcal{N}_d^1(d, 1, L)$ is a universal approximator of $C^0(\mathbb{R})$.*

The notation $\mathcal{N}_d^1(d, 1, L)$ indicates the set of realizations of ReLU NNs with arbitrary $B$ and $W$. Although Theorem 2.1.3 is studied for approximation in the uniform norm, we can obtain similar results when $L^p$ norms are used instead. More interesting for us is the fact that the set of realizations of NNs with a greater number of layers exhibits this property as well.

**Theorem 2.1.4.** *Let $d, L \in \mathbb{N}$. Let $K \subset \mathbb{R}^d$ be a compact set and $\varrho : \mathbb{R} \to \mathbb{R}$ an activation function that is differentiable and not constant in an open interval. Then, for $\epsilon > 0$, there is an NN $\Phi^x \in \mathcal{NN}(d, 1, L)$ such that*

$$\sup_{\mathbf{x} \in K} \|\mathrm{R}(\Phi^x)(\mathbf{x}) - \mathbf{x}\| < \epsilon,$$

*where the measure $u$ is the restriction of the Lebesgue measure.*

Universal approximation theorems shed light on the success of neural-network-based methods. As many phenomena can be described by continuous functions, NNs are able to approximate the solution of a vast number of problems. However, these results do not provide a simple method to select an NN that better fits the given data. New methods have been developed to address these issues that make use of various techniques and we study some of such techniques in the following sections. For all of the problems that we study in this thesis, we provide the architecture of the NN with which we construct our algorithm that reconstructs their solution, and we present our results from the following chapters in the form of PETs.

## 2.2 Approximation of Function Classes

Now, we study the approximation capabilities of NNs for different function sets. In particular, the results of this section show which architecture to select for the set of realizations of NNs to be a suitable hypothesis set under certain conditions on the ground-truth function to be approximated. Furthermore, we state explicit rates in terms of the number of weights and the input dimension of the NN so that its realization approximates a function with a given accuracy. We constantly revisit these results in Chapters 3, 4 and 5 as the PETs presented in these chapters rely on suitable NN sets that rely on the results of this section.

### Linear Functions as Realizations of Neural Networks

One of the simplest function sets that can be approximated by the realization of NNs is that of continuous, piecewise linear [106]. The approximability of these functions by the realization of NNs leads us to many approximation results for functions with certain regularities. We now formalize the concept of continuous, piecewise linear function.

**Definition 2.2.1.** *[106, Adapted from Definition 3] Let $d \in \mathbb{N}$, $\Omega \in \mathbb{R}^d$. A function $f : \mathbb{R}^d \supseteq \Omega \to \mathbb{R}$ is said to be a continuous, piecewise linear function (in short, cpwl function) if $f \in C^0(\Omega)$, the set of continuous functions on $\Omega$, and there exist $p \in \mathbb{N}$ affine linear functions $g_k : \mathbb{R}^d \to \mathbb{R}$ such that for all $\mathbf{x} \in \Omega$ there is at least one $j \in \{1, 2, \ldots, p\}$ with $f(\mathbf{x}) = g_j(\mathbf{x})$.*

For a given vector $\mathbf{v} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$, the simple neuron

$$\mathbf{x} \mapsto \varrho(\mathbf{v}^T\mathbf{x} - b),$$

for $\mathbf{x} \in \mathbb{R}^d$, divides the Euclidean space $\mathbb{R}^d$ into two regions, where the function is linear on each one of them. Thus, a neuron is a cpwl function with $p = 2$. The sum of different neurons gives rise to a continuous function which is continuous and piecewise linear. We now state that each cpwl function is realized by an NN. This result can be formulated in a sufficiently general form, but for the purposes of this work, we state this result for a particular case. We note that Definition 2.2.1 does not impose any restriction on the domain of each $g_j$. However, we state this result under some convexity restrictions on these domains. To this end, we recall some crucial definitions from the theory of convex sets.

**Definition 2.2.2.** *[106] Let $d \in \mathbb{N}$. The* convex hull *of a set $U \subset \mathbb{R}^d$ is defined by*

$$co(U) := \{\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i | \quad n \in \mathbb{N}, \mathbf{x}_i \in U, \alpha \geq 0, \sum_{i=1}^{n} \alpha_i = 1\}.$$

In particular, we use the convex hull of a finite number of points, denoted as $co(\{x_0, \ldots, x_n\})$ and presented in the following definition.

**Definition 2.2.3.** *[107, Chapter 50] Let $n \in \mathbb{N}_0$, $d \in \mathbb{N}$. We say that the points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ are* affinely independent *if $\{\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0, \ldots, \mathbf{x}_n - \mathbf{x}_0\}$ is a linearly independent set. In this case, we say that the set $co(\mathbf{x}_0, \ldots, \mathbf{x}_n)$ is an $n$-simplex and each element $\mathbf{x}_i$ is called a* node*.*

The partitions that we consider in this thesis are only *regular triangulation*, which are defined below.

**Definition 2.2.4.** *[61, Chapter 3] [129, Definition 5.13] Let $d \in \mathbb{N}$ and $\Omega \subset \mathbb{R}^d$ be a compact set. We say that $\mathcal{T} = \{\tau\}_{\tau \in \mathcal{T}}$ where each $\tau$ is a $d-$simplex, is a* regular triangulation *of $\Omega$ if*

1. *$\bigcup_{\tau \in \mathcal{T}} \tau = \Omega$,*
2. *for all $\tau, \tau'$, we have $\tau \cap \tau' = co(N(\tau) \cap N(\tau'))$,*

*where $N(\tau)$ denotes the set of nodes of $\tau$. The maximum number of elements shared by a single node is denoted by $K_{\mathcal{T}}$.*

Next, we formulate Theorem 2.2.5 which guarantees the existence of an NN that realizes each cpwl function when $\Omega$ admits a regular triangulation. Remarkably, the control bounds on the number of weights and layers hinge on the number of elements of the partition $\mathcal{T}$ and the quantity $K_{\mathcal{T}}$.

**Theorem 2.2.5.** *[61, Theorem 3.1] [129, Theorem 5.14] Let $d \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$ be a bounded domain, and $\mathcal{T}$ be a regular triangulation of $\Omega$. Let $f : \Omega \to \mathbb{R}$ be a cpwl function with respect to $\mathcal{T}$ which is zero at the boundary of $\Omega$. Then, there is a ReLU NN $\Phi^f$ such that*

$$\mathrm{R}(\Phi^f)(\mathbf{x}) = f(\mathbf{x}),$$

*for $\mathbf{x} \in \Omega$, and there is a constant $C > 0$ depending only on $d$ and $K_T$, such that*

$$W(\Phi^f) \leq C(|T|), \quad L(\Phi^f) \leq C.$$

Building upon Theorem 2.2.5, we can construct NNs with larger numbers of layers that allow us to approximate more complex functions, as we see in the following sections. We follow this approach in Chapter 4.

### Approximation of Polynomials by NNs

The approximation of polynomials by NNs is a crucial step to the construction of NNs that approximate further functions, particularly smooth functions or with a certain regularity. The method we discuss here was originally introduced in [172]. The proofs herein are built on the cpwl function

$$f(x) = \begin{cases} 2x, & \text{if } x \in [0, \frac{1}{2}], \\ 2 - 2x, & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$

which can be realized by a $\Phi^f$ such that $\mathrm{R}(\Phi^f)(x) = \varrho(2x) - \varrho(4x - 2)$ where $\varrho$ is, as usual, the ReLU activation function. We introduce the notation

$$f_n = f_{n-1} \circ f \text{ for all } n \in \mathbb{N}_{\geq 2}. \tag{2.2.1}$$

for the subsequent compositions of $f$ by itself. Direct calculations show that the resulting function $f_n$ is a cpwl function as well. Thus, we conclude that there is an NN $\Phi^{f_n}$, the realization of which equals $f_n$ for each $n \in \mathbb{N}$. This observation is employed for the construction of an NN, the realization of which approximates the quadratic function.

**Proposition 2.2.6.** *[119, Proposition 2.5] Let $m \in \mathbb{N}$. Let $f_m$ defined as in (2.2.1) and $g : [0, 1] \to \mathbb{R}$ given by $g(x) = x^2$. Then, $f_m$ is a piecewise linear function such that $f_m(j2^m) = (j2^m)^2$ for $j = 0, \ldots, 2^n$. Moreover, $f_m$ is a NN with $m$ hidden layers, at most 5 weights per layer and*

$$\|g - f_m\| \leq 2^{-2m-1}.$$

By using the identity

$$x \cdot y = \frac{(x+y)^2 - (x-y)^2}{4},$$

we can approximate the product of two numbers $x, y$ in a compact interval $[-B, B]$, $B > 0$, by the realization of an NN depending on the given accuracy $\epsilon > 0$ as shown in the following proposition.

**Proposition 2.2.7.** *[119, Proposition 2.5] Let $\epsilon \in (0,1)$ and $B > 0$. Let $\times : \mathbb{R}^2 \to \mathbb{R}$ be defined by $\times(x,y) = xy$. Then, there exists a ReLU NN $\widetilde{\times}_{\epsilon,B}$ such that $\mathrm{R}(\widetilde{\times}_{\epsilon,B}) : [-B,B]^2 \to \mathbb{R}$ and that*

$$\|\mathrm{R}(\widetilde{\times}_{\epsilon,B}) - \times\|_{L^\infty([-B,B]^2)} \leq \epsilon.$$

*Moreover, there is a universal constant $C > 0$ such that*

- $L(\widetilde{\times}_{\epsilon,B}) \leq C(1 + \log_2(B/\epsilon))$,
- $W(\widetilde{\times}_{\epsilon,B}) \leq C(1 + \log_2(B/\epsilon))$.

To approximate polynomials, we require not only an approximation result for the quadratic function, but also an approximation result for the product of an arbitrary number of factors. This result is established in the following proposition.

**Proposition 2.2.8.** *[119, Proposition 2.6] Let $\epsilon \in (0,1)$ $d \in \mathbb{N}$ and $B > 0$. Let $\times : \mathbb{R}^d \to \mathbb{R}$ be defined by $\times(x_1, \ldots, x_d) = \prod_{j=1}^d x_j$. Then, there exists an NN $\widetilde{\times}_{\epsilon,B}$ such that $\mathrm{R}(\widetilde{\times}_{\epsilon,B}) : [-B,B]^d \to \mathbb{R}$ and that*

$$\|\mathrm{R}(\widetilde{\times}_{\epsilon,B}) - \times\|_{L^\infty([-B,B]^d)} \leq \epsilon.$$

*Moreover, there is a universal constant $C > 0$ such that*

- $L(\widetilde{\times}_{\epsilon,B}) \leq C(1 + \log_2(d)\log_2(d/\epsilon))$,
- $W(\widetilde{\times}_{\epsilon,B}) \leq C(1 + d\log_2(d/\epsilon))$.

As each polynomial function is a linear combination of a finite number of monomials, we can invoke Proposition 2.0.9 and Proposition 2.2.7 to derive the following result.

**Proposition 2.2.9.** *[119, Proposition 2.9] Let $\epsilon \in (0,1)$. Let $p(x) = \sum_{j=0}^n c_j x^j$ be a polynomial of degree $n \in \mathbb{N}$, where $c_j \in \mathbb{R}$ for $j = 0, \ldots, n$. Then, there exists an NN $\Phi_{\epsilon,p}$ such that*

$$\|p - \mathrm{R}(\Phi_{\epsilon,p})\|_{L^\infty([-1,1])} \leq \epsilon.$$

*Moreover, there is a universal constant $C > 0$ such that*

- $L(\Phi_{\epsilon,p}) \leq C(1 + n\log_2(n) + n\log(C/\epsilon))$,
- $W(\Phi_{\epsilon,p}) \leq C(\log_2(n)^3 + (1 + \log_2(n))\log_2(C_0/\epsilon))$.

*where* $C_0 := \max\{\sum_{j=2}^{n} |c_j|, \delta\}$.

Notice that the number of layers of the constructed NNs in Theorem 2.2.7 depends only logarithmically on the input dimension $d$. Additionally, the number of weights of the constructed NNs depends at most linearly on the input dimension $d$. Following the discussion of the introductory chapter, it seems that we do not necessitate an excessively large (compared to the dimension) number of parameters to approximate polynomials by the realization of NNs. In the following sections, we continue to study such a dependency for different sets of functions.

### Approximation of Legendre Polynomials by NNs

One class of polynomials that plays an important role in applications is the set of Legendre polynomials. In particular, we use these polynomials in Chapter 5 for the approximation of an orthogonal basis of $L^2(\mathbb{R})$. We say that $\widetilde{P_k}$ is a *Legendre Polynomial* of degree $k \in \mathbb{N}$ if it is the polynomial solution of the ODE

$$\forall x \in [-1,1], \quad (1-x^2)\frac{d^2}{dx^2}\widetilde{P_k} - 2x\frac{d}{dx}\widetilde{P_k} + k(k+1)\widetilde{P_k} = 0.$$

These polynomials constitute an orthonormal basis for $L^2_{\mathfrak{u}}([-1,1])$. In [120, Equation 2.42] it is shown that the $L^2_{\mathfrak{u}}$-norm of the Legendre polynomial $\widetilde{P_k}$ on $[-1,1]$ is given by

$$\|\widetilde{P_k}\|_{L^2_{\mathfrak{u}}([-1,1])} = \frac{1}{2\sqrt{k+1/2}}. \tag{2.2.2}$$

To obtain a normalized Legendre polynomial of degree $k$, we define

$$P_k := 2\sqrt{k+1/2}\widetilde{P_k}.$$

Write $P_k(x) = \sum_{j=0}^{k} c_j^k x^j$. An explicit formula for the coefficients $c_k$ can be found in [15, Section 10.10, Equation (16)]). Therein, it is stated that the coefficients $c_j^k$ satisfy

$$c_j^k := \begin{cases} 0 & \text{if } k-j \in \{0,\dots,j\} \cap 2\mathbb{Z}+1, \\ (-1)^{\mathfrak{m}} 2^{-k} \binom{k}{\mathfrak{m}} \binom{k+j}{\mathfrak{m}} \sqrt{2k+1} & \text{if } k-j \in \{0,\dots,j\} \cap 2\mathbb{Z}. \end{cases}$$

where $\mathfrak{m} := (k-j)/2$. Thus, by invoking Proposition 2.2.9, we are able to derive an approximating result for the approximability of normalized Legendre polynomials by ReLU neural networks.

**Proposition 2.2.10.** *[119, Proposition 2.11] Let $\epsilon \in (0,1)$. For each normalized Legendre polynomial $P_k$ of degree $k \in \mathbb{N}$, there is an NN $\Phi_{\epsilon,k}$ with input and output dimension one such that*

$$\|P_k - \mathrm{R}(\Phi_{\epsilon,k})\|_{L^\infty_{\mathfrak{u}}([-1,1])} \leq \epsilon.$$

*Moreover, there exists a universal constant $C > 0$ such that for all $k \in \mathbb{N}$*

- $L(\Phi_{\epsilon,k}) \leq C(1+\log_2 k)(k+\log_2(1/\epsilon))$
- $W(\Phi_{\epsilon,k}) \leq Ck(k+\log_2(1/\epsilon))$.

*In the special case* $k=0$,

$$P_0 = \mathrm{R}(\Phi_{\epsilon,0})$$

*with*

- $L(\Phi_{\epsilon,0}) = 2$,
- $W(\Phi_{\epsilon,0}) = 1$.

These approximation rates will be used for the construction of a set of realizations of NNs in Chapter 5. In particular, we use Legendre polynomials to approximate the Slepian basis, which is later introduced, and is an orthonormal bases for the space of $L^2$ functions on the space $\mathbb{R}^d$.

### 2.2.1 Approximation of Lipschitz Functions

We now start to study the problem of approximating more complex functions by the realization of NNs. One standard assumption on the ground-truth function is that of Lipschitz continuity which we recall below.

**Definition 2.2.11.** *[75, Definition 2.1] Let* $d, q \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$ *and* $C > 0$. *We say that* $g : \Omega \to \mathbb{R}^q$ *is Lipschitz continuous with Lipschitz constant* $C$, *if for all* $\mathbf{x}, \mathbf{y} \in \Omega$, *it holds*

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_{\mathbb{R}^q} \leq C\|\mathbf{x}-\mathbf{y}\|_{\mathbb{R}^d}.$$

As usual, our goal is to prove that for a given $\epsilon > 0$, there is an NN $\Phi^f$ such that

$$\left\|f - \mathrm{R}\left(\Phi^f\right)\right\|_{L^\infty(\Omega;\mathbb{R}^q)} \leq \epsilon.$$

Moreover, we would like to have control over the number of weights $W(\Phi^f)$ and the number of layers $L(\Phi^f)$. The following result states that each Lipschitz continuous function $f$ can be efficiently approximated by the realization of NNs with control over the weights.

**Theorem 2.2.12.** *[61, Corollary 5.1] Let* $d, q \in \mathbb{N}$, $K > 0$ *and let* $\Omega = [-K, K]^d$. *Let* $g : \Omega \to \mathbb{R}^q$ *be Lipschitz continuous with Lipschitz constant* $C > 0$. *Then, for every* $\epsilon \in (0,1)$, *there exists an NN* $\Phi^{g,\epsilon}$ *such that*

- $\|g - \mathrm{R}\left(\Phi^{g,\epsilon}\right)\|_{L^\infty(\Omega;\mathbb{R}^q)} \leq \epsilon$,
- $W\left(\Phi^{g,\epsilon}\right) = \mathcal{O}\left(q\epsilon^{-d}\right)$ *for* $\epsilon \to 0$,
- $L\left(\Phi^{g,\epsilon}\right) = \lceil \log_2(d+1) \rceil + 1$.

*In the above estimate, the implicit constant depends on* $C$, $K$, *and* $d$.

We revisit this result in Chapter 4 for the construction on a convenient NNs defined on a low-dimensional manifold. Such an NN allows us to construct deep-network-based algorithms for the solution of inverse problems.

**Approximation of $C^{k,s}$ function**

Stronger regularity conditions of a function can be exploited to obtain different approximation rates using NNs. Before we reach these results, we introduce the set of $C^{k,s}$ functions.

**Definition 2.2.13.** *petersen2024mathematical Let $d \in \mathbb{N}$, $k \in \mathbb{N}_0$, $s \in [0,1]$ and $\Omega \subset \mathbb{R}^d$. We denote $C^{k,s}(\Omega)$ as the set of functions $f : \mathbb{R}^d \to \mathbb{R}$ such that*

$$\|f\|_{C^{k,s}(\Omega)} := \sup_{\mathbf{x}\in\Omega} \max_{\{\alpha\in\mathbb{N}_0^d | \alpha| \leq k\}} |D^\alpha f(\mathbf{x})| + \sup_{\mathbf{x}\neq\mathbf{y}} \max_{\{\alpha\in\mathbb{N}_0^d | \alpha| \leq k\}} \frac{|D^\alpha f(\mathbf{x}) - D^\alpha f(\mathbf{y})|}{\|\mathbf{x}-\mathbf{y}\|_2^2}$$

*is finite.*

We now present an approximation rate by NNs for the functional space $C^{k,s}$.

**Theorem 2.2.14.** *[129, Theorem 5.23] Let $d \in \mathbb{N}$, $k \in \mathbb{N}_0$, $s \in [0,1]$ and $\Omega = [0,1]^d$. Then, there is a universal constant $C > 0$ such that for every $f \in C^{k,s}(\Omega)$ and $N \geq 2$, there is a ReLU NN $\Phi_N^f$ such that*

$$\|f - \phi_N^f\| \leq C N^{-\frac{k+s}{d}} \|f\|_{C^{k,s}(\Omega)}.$$

*Moreover, the NN $\Phi_N^f$*

- $W(\Phi_N^f) \leq CN\log(N)$,
- $L(\Phi_N^f) \leq C\log(N)$.

From Theorem 2.2.14 we can notice that the higher regularity of the function $f$ leads to faster approximation rates. However, this result shows that for functions defined in high-dimensional Euclidean spaces, we may need NNs with a massive number of parameters and layers to achieve convenient approximation errors. This phenomenon appears frequently in the study of approximation methods and is commonly referred to as the *curse of dimensionality*. This is the topic of the next section.

## 2.3 The Curse of Dimensionality

It has been noted that in high-dimension settings, many algorithms to approximate their solutions become *exponentially* harder [5]. This phenomenon is usually referred to as the *curse of dimensionality*, a key concept encountered throughout the following chapters and initially coined by Bellman in [17]. Deep-learning-based algorithms for the reconstruction of functions with derivatives up to the order $k$ defined in the space $\mathbb{R}^d$ for $d \in \mathbb{N}$ may suffer from the curse of dimensionality, as we have already observed in Theorem 2.2.14 and Theorem 2.2.12. For the case of Lipschitz continuous functions defined on compact sets, the number of total parameters grows exponentially with respect to the dimension. This is a clear example of the effects of the curse of dimensionality on the problem of approximating a function by NNs.

In this section, we present two different approaches that have emerged to mitigate the curse of dimensionality for neural-network-based approximation methods. These strategies are leveraged in Chapters 3 and 4 for the construction of specific NN classes that are not subject to the curse of dimensionality. In particular, we introduced the *manifold assumption* and the space of *Barron functions*.

### The Manifold Assumption

For many applications, we can expect the input data to belong to a subset of a lower-dimensional subspace. This assertion is supported by empirical evidence, since it has been observed that the number of parameters needed for an NN to approximate certain regular functions is lower than the number of parameters predicted by standard results (see [14, 84]). To remedy this issue, a common approach is to assume that the definition domain of the ground-truth function is a low-dimensional manifold $\mathcal{M}$ (see [160, 140]). We refer to this approach as the *manifold assumption*, the low-dimensional manifold as the *intrinsic domain* and the Euclidean space as the *ambient domain*. With this terminology, the ground-truth can be represented as a function $f : \mathbb{R}^d \supset \mathcal{M} \to \mathbb{R}^p$.

Under the manifold assumption, the curse of dimensionality for neural-network-based approximation methods can be lessened, as we see in this section. This can be seen in Theorem 2.3.1, where the dimension of the manifold determines the approximation rate of the method. Notice that such a rate depends exponentially on the intrinsic dimension $d$ and not on the ambient dimension $D$. We assume that the function to be approximated belongs to the space of $C^{k,s}(\mathcal{M})$ functions where $\mathcal{M}$ is an $m-$dimensional smooth, compact submanifold of $\mathbb{R}^d$.

**Theorem 2.3.1.** *[129, Theorem 8.7] Let $d, k \in \mathbb{N}$, $s \in [0, 1)$. Let $\mathcal{M} \subset \mathbb{R}^d$ a smooth, compact and $m-$dimensional manifold. Then, there is a constant $C > 0$ such that for all $f \in C^{k,s}(\mathcal{M})$ and every $N \in \mathbb{N}$, there is a ReLU NN $\Phi_N^f$ such that,*

- $\|f - \mathrm{R}(\Phi_N^f)\|_{L^\infty(\mathcal{M})} \leq C N^{-(k+s)/m}$,
- $L(\Phi_N^f) \leq C \log(N)$,
- $W(\Phi_N^f) \leq CN \log(N)$.

Several articles have proved similar approximation rates for different function classes under various assumptions on the manifold $\mathcal{M}$ (see, for example, [31, 145]). In Chapter 4, we present a version for Lipschitz continuous functions when the intrinsic domain $\mathcal{M}$ is an $m-$dimensional smooth, compact submanifold of $\mathbb{R}^d$. We incorporate the manifold assumption in Chapter 4 for the design of a new scheme to approximate the solution of various inverse problems. Since the forward operator associated with an inverse problem is defined on infinite-dimensional spaces, it is not evident that neural networks can effectively approximate its solution. Moreover, the inverse operator is typically noncontinuous and a similar version of Theorem 2.3.1 cannot be applied directly. Particularly, we propose approximation methods

based on NNs for the solution of infinite-dimensional inverse problems where the quantity to be identified lies in a low-dimensional solutions. Some of the underlying ideas of the proof of Theorem 2.3.1 are revisited in Chapter 4 to construct a robust-to-noise NN, the realization of which approximates inverse operators.

### The Barron class

In [14], it was proved that there exists a class of functions for which the number of neurons in the set of realizations of a shallow neural network required to achieve an approximation accuracy of $1 > \epsilon > 0$ grows like $O(\epsilon^{-2})$. This result points to the fact that nonlinear approximation via NN can avoid the curse of dimensionality. This space is called *Barron space* and is discussed in this section. Intuitively, the Barron class corresponds to the set of integrable functions whose first Fourier moment is bounded. We formally introduce this space in Definition 2.3.2.

**Definition 2.3.2.** *[128, Definition 2.1] Let $d \in \mathbb{N}$. The Barron class $B(C) = B_d(C)$ with associated constant $C > 0$ is defined as the set of functions $f : B_1^d(0) \to [0,1]$ for which there is a measurable function $F : \mathbb{R}^d \to \mathbb{C}$ and a constant $c \in [-C, C]$ such that*

$$f(x) = c + \int_{\mathbb{R}^d} (e^{i\langle x,t\rangle} - 1)F(t)dt \text{ for all } x \in [0,1]^d \text{ and } C_f := \int_{\mathbb{R}^d} |t||F(t)|dt \leq C.$$

We emphasize that there are different definitions for the space of Barron functions, some of which are equivalent but are not included in this thesis. An extensive study of these equivalences can be found [49]. One such generalization stems from the fact that the Barron class is efficiently approximated by shallow NNs. This claim is supported by the following result.

**Theorem 2.3.3.** *[30, Proposition 2.2] There is a universal constant $\kappa > 0$ with the following property: For every bounded set $\Omega \subset \mathbb{R}^d$ with nonempty interior, for every $C > 0$, $x_0 \in X$ and $f \in B_C(\Omega)$, and for all $N \in \mathbb{N}$, there is a shallow NN $\Phi_N^f$ with at most $8N$ neurons in the hidden layer, such that*

$$\|f - \mathrm{R}(\Phi_N^f)\|_{L^\infty(\Omega)} \leq \kappa\sqrt{d}CN^{-1/2}. \tag{2.3.1}$$

*Moreover, the weights and biases of $\Phi_N^f$ can be bounded by*

$$(5 + \nu(A))\sqrt{C} \text{ where } \nu(A) := \sup_{z \in \mathbb{R} - \{0\}} \left(|z|_\infty / \|z\|_2\right).$$

From Theorem 2.3.3, we observe that the approximation rate term $N^{-1/2}$ does not depend exponentially on the input dimension $d$. In particular, the number of neurons needed to obtain an approximation accuracy $\epsilon$, although not completely independent of $d$ does not grow exponentially. This may be surprising, as it suggests that the curse of dimensionality may be mitigated for the class of Barron functions. In reality, there is a dependency on the

dimension in the approximation rate. The Barron condition is an example of nonclassical smoothness criteria [49]. Functions with more classical regularity belong to the Barron class, like for example, functions in the Sobolev space $H^{d/2+2}(\mathbb{R}^d)$.

A recently introduced functional space, called the $RBV^2$-space, exhibits a similar behavior. In Chapter 3, we study in detail how NNs can be used for the approximation of this type of functions without the curse of dimensionality. Indeed, approximation results similar to Theorem 2.3.3 for the class of $RBV^2$ functions have been presented in [123]. A key difference from such statements is that the exponent in the approximation rates does depend on the input dimension $d$. However, the dependency does not entail the curse of dimensionality, since the approximation rates for $RBV^2$ approach the approximation rates for Barron classes as the dimension grows.

## 2.4 Learning from Samples

The proofs of the results in the later sections rely on highly technical constructions that we have not included in this thesis. In practice, the selected NN for a ground-truth function is selected using an *optimization algorithm* on a limited number of data samples. To guarantee this, we usually use techniques from other fields of study, such as *statistical learning theory*. [162]. The NN selection process or *training* and we must guarantee that the selected NN performs well on unseen data. We intent to provide some insight into this topic in this section.

Let us assume that we wish to find the relation between two different variables $x$ and $y$. We further assume $x \in X$ and $y \in Y$ where $X$ is called the *feature space* and $Y$ is the *label space*. Moreover, we have a finite set of observations $(x_i, y_i)_{i=1}^m$ for $m \in \mathbb{N}$ where $(x_i, y_i)$ is drawn according to a probability distribution $\mathcal{D}$. We call this set the *training set*. The relation between the two components of each data point is given by a ground-truth function $f$, such that $f(x_i) = y_i$. Our approach is to search for an NN $\Phi$, the realization of which fits the observations as best as possible, i.e., reduces the approximation error on the training set and can predict accurately the output of a new data set, called the *test set*.

To measure the disparity between the ground-truth $f$ and the realization of the NN $\Phi$ we make use of a so-called *loss function*, which is a function $L : Y \times Y \to \mathbb{R}$ chosen for the problem. This loss function induces the concept of risk, which we present in the following.

**Definition 2.4.1.** *[104, Definition 2.1] Let $L : Y \times Y \to \mathbb{R}_+$ be a loss function and $\mathcal{D}$ a distribution over $X \times Y$. The risk of a function $f : X \to Y$ is defined by*

$$\mathcal{R}(h) := \mathbb{E}_{\mathcal{D}}[L(f(x), y)].$$

*Moreover, we define the Bayes risk as*

$$R^* = \inf_{f:X\to Y} \mathcal{R}(f).$$

In terms of this newly introduced concept, our objective is now to find an NN the realization of which has a risk close enough to the Bayes risk. To this end, we consider a subset

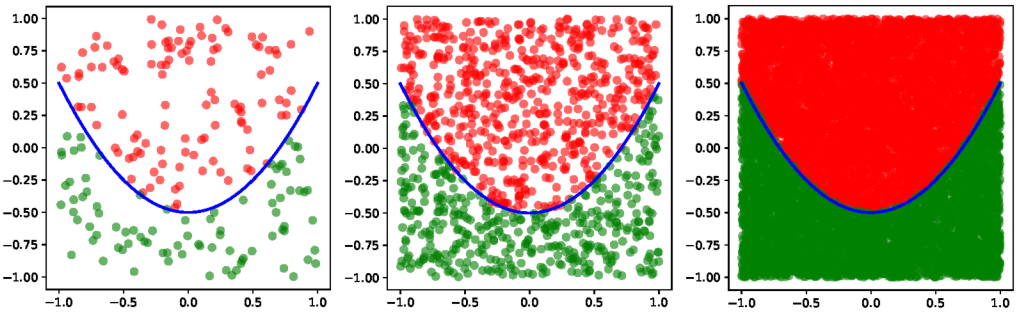

Figure 2.3: The problem of learning depends on the amount and quality of data available.

of realizations of NNs $\mathcal{H} \subset \{f : X \to Y\}$ called the *hypothesis set*. Since the distribution $\mathcal{D}$ is unknown, it may be impossible to compute the risk $\mathcal{R}(f)$ explicitly. Thus, we use the sample set to compute the *empirical risk* of the function as an approximation of the risk of $f$.

**Definition 2.4.2.** *[104, Definition 2.2] Let $m \in \mathbb{N}$. Let $\mathcal{H}$ a hypothesis set and $L : Y \times Y \to \mathbb{R}$ be a loss function. Let $S = (x_i, y_i)_{i=1}^m \in (X \times Y)^m$ be a sample. We define the* empirical risk *of a function $h \in \mathcal{H}$ as*

$$\hat{R}_S(h) := \frac{1}{m} \sum_{i=1}^{m} L(h(x_i), y_i).$$

*If there is a function $h_S$ such that*

$$\hat{R}_S(h_S) = \inf_{h \in \mathcal{H}} \hat{R}_S(h),$$

*we call this function the* empirical risk minimizer.

Neural-network-based methods use the *empirical risk minimizer* to estimate the function to be learned [143]. Notice that

$$\begin{aligned} R(h_s) - R^* &\leq 2 \sup_{h \in \mathcal{H}} |R(h) - \hat{R}(h)| + \inf_{h \in \mathcal{H}} R(h) - R^* \\ &=: 2\epsilon_{gen} + \epsilon_{app}. \end{aligned}$$

In the previous sections, the approximation rates for some classes have already been established. Now, we study methods to derive upper bounds on the generalization error. Before we do this, we briefly discuss how to find in practice the empirical risk minimizer for a given task.

**The method of gradient descent.** To find the empirical risk minimizer, we typically perform an iterative algorithm called *the Gradient Descent Method (GD)* [114, Section 1.2.3]. This method finds a minimizer on the set of NN parameters, which we consider in this section to be $\mathbb{R}^N$, where $N$ denotes the total number of parameters. Let us consider the problem

$$\min_{\mathbf{w}\in\mathbb{R}^N} f(\mathbf{w}),$$

where $N \in \mathbb{N}$ and $f$ is the empirical risk considering a sample $S$. Starting with an initial step $\mathbf{w}_0 \in \mathbb{R}^d$, we define a sequence of weights given by

$$\mathbf{w}_{k+1} := \mathbf{w}_k - h_k \nabla f(\mathbf{w}_k), \tag{2.4.1}$$

where $h_k > 0$ is called the $k-$size step and can be different for each $k \in \mathbb{N}$. To study this method, one may impose convexity assumptions on the objective function and the definition domain as the GD method does not converge for any choice of function. Since we do not analyze the convergence of the method in the following chapters, we provide only relevant literature on this matter (see [53, 86]).

If we use a large data set to define our objective function, we may have a complicated functional to optimize [69]. Thus, to find the empirical risk minimizer, we commonly take a different approach. We replace the gradient of the objective function $\nabla f(\mathbf{w}_k)$, which is computed on the whole training set in 2.4.1 by a random variable $\mathbf{G}_k$ that is an unbiased estimator of this gradient, that is,

$$\mathbb{E}(\mathbf{G}_k|\mathbf{w}_k) = \nabla f(\mathbf{w}_k),$$

which leads us to the following update rule,

$$\mathbf{w}_k := \mathbf{w}_k - h_k \mathbf{G}_k.$$

This variation of the GD method is called *stochastic gradient descent (SGD)* ([48]).

**Generalization bounds** We now formalize the notion of generalization bounds, which provide insight into how closed the risk and the empirical risk of a function are in terms of the number of samples [166, Section 5.5].

**Definition 2.4.3.** *[129, Definition 14.6] Let $\mathcal{H}$ a hypothesis set of functions mapping $X$ into $Y$. Let $L : Y \times Y \to \mathbb{R}$ be a loss function. Let $\kappa : (0,1) \times \mathbb{N} \to \mathbb{R}_+$ be a function such that for every $\delta \in (0,1)$, it holds that $\kappa(\delta, m) \to 0$ as $m \to \infty$. We say that $\kappa$ is a* generalization bound *for $\mathcal{H}$ if for every distribution $\mathcal{D}$ on $X \times Y$, and every $m \in \mathbb{N}$ and $\delta \in (0,1)$, it holds with probability at least $1 - \delta$*

$$\sup_{h\in\mathcal{H}} |\mathcal{R}(h) - \hat{\mathcal{R}}_S(h)| \leq \kappa(\delta, m).$$

*when $S \sim \mathcal{D}$.*

With the concept of *covering number*, which we present below, we can obtain generalization bounds for multiple hypothesis classes.

**Definition 2.4.4.** *[128, Definition 3.9] Let $K, X \neq \varnothing$ be sets with $K \subset X$ and let $d : X \times X \to \mathbb{R}_0^+$ a be function such that $d(x, y) \geq 0$ for distinct $x, y \in X$ and $d(x, y) = 0$ if $x = y \in X$. We say that a subset $G_\epsilon \subset X$ is an $\epsilon-$net for $K$ if for every $x \in X$, there is $y \in G_\epsilon$ such that $d(x, y) \leq \epsilon$. Furthermore, we define $V_{K,\epsilon}$ the $\epsilon-$covering entropy of $K$ as*

$$V_{K,\epsilon}(\epsilon) := \ln(\mathcal{G}_{K,d})$$

*where*

$$\mathcal{G}_{K,d} := \max\{|G| : G \text{ is an } \epsilon\text{-net for } K\}$$

*and $|G|$ denotes the cardinality of the set $G$.*

For the set of NNs $\mathcal{F} := \mathcal{NN}(d, 1, N, W, B)$, we can compute the covering entropy, which we denoted as $V_{\mathcal{F},\|\cdot\|_\infty}$. Here, $\|\cdot\|_\infty$ must be understood as the distance induced by the supremum norm. In [128, Remark 5.2], this quantity was calculated as

$$V_{\mathcal{F},\|\cdot\|_\infty}(\delta) \leq W \cdot (10 + \ln(1/\delta) + 5\ln(\lceil B \rceil) + 5\ln(\max\{d, W\}))\,,$$

for $\delta \in (0, 1]$, $d, N, W \in \mathbb{N}$ and $B > 0$. The same bound holds for $\mathcal{NN}_*(d, 1, 2, N, W, B)$ since this set is the intersection of $\mathcal{NN}(d, N, W, B)$ with a convex set. This result will be recalled in Chapter 3. The covering entropy numbers of a class can be used to derive generalization bounds. To illustrate this idea, we present the following theorem.

**Theorem 2.4.5.** *[128, Theorem 3.9] Let $d, m \in \mathbb{N}$, $C_Y$, $C_L > 0$ and $\alpha > 0$. Let $Y \subset [-C_Y, C_Y], X \subset \mathbb{R}^d$. Let $L : X \times Y \to \mathbb{R}$ be a Lipschitz continuous loss function with Lipschitz constant $C_L$. Let $\mathcal{H}$ be a subset of functions mapping $X$ into $Y$. Then, for every distribution $\mathcal{D}$ on $X \times Y$, and each independent sample $S \sim \mathcal{D}$ of $m$ elements, it holds*

$$|R(h) - \widetilde{R}(h)| \leq 4C_Y C_L \sqrt{\frac{V_{\mathcal{H},\|\cdot\|_\infty}(m^{-\alpha}) + \log(2/\delta)}{m}} + \frac{2C_L}{m^\alpha},$$

*for all $h \in \mathcal{H}$.*

The results in this section are relevant to us because PETs demonstrate the existence of a class of NNs, along with the training procedure and convenient sample complexities to determine an adequate NN for a ground-truth function. (see [3]). We use standard results from statistical learning theory as those presented above in Chapters 3 and 4. For the PET in Chapter 5, we use alternative results from sparse approximation theory [2].

# Chapter 3

# A Practical Existence Theorem for Classification Problems

In this chapter, we discuss a PET for a certain class of discontinuous functions which are typically encountered in tasks of *classification*. The ground-truth functions associated with this type of problems are commonly assumed to be defined on a compact subset $\Omega \subset \mathbb{R}^d$ which can be partitioned into a finite number of subsets $\Omega_k$.

Moreover, these functions take a unique value on each partition set and are usually referred to as the *classifier* $f$. Formally speaking, a classifier $f$ takes the form $f(x) = \sum_{k=1}^{K} c_k \mathbb{1}_{\Omega_k}(x)$ where $\Omega_k \subset \mathbb{R}^{d+1}$, $d \in \mathbb{N}$ are disjoint sets and $c_k$ is called the label of $\Omega_k$ for $k = 1, \dots, K$ (see [128, 132]). In particular, we assume all the labels to be natural numbers $1, 2, \dots, K$, but the set of labels can be a subset of a more general set. For simplicity, all the results of this chapter are proved only the case of binary classifiers, i.e. $K = 2$, but they can be directly extended for an arbitrary number of labels. In Figure 3.1, we illustrate the partition of a set $\Omega$ into two different sets $\Omega_1$ and $\Omega_2$. Neural networks have been applied to approximate and estimate classifiers from a fixed number of samples. Classifying pictures from the MNIST dataset is a famous classification problem where deep-learning-based methods have been successfully implemented [90]. Each element of this dataset is a picture of $28 \times 28$ pixels with a handwritten digit from 0 to 9. Some of the images encountered in this dataset are illustrated in Figure 3.2. Data-driven algorithms aim at assigning a discrete label from 0 to 9 to each one of the picture of the MNITS dataset. Several studies have shown how classification algorithms based on NNs perform this task, and convolutional neural networks (CNN) are usually applied to the design of algorithms to solve this problem [71, 146, 171].

For our theory to be applied, we assume that all classifiers have a particular property: the decision boundary, which corresponds to the boundary of each partition set of the classifier, must be parametrized by a $RBV^2$ function. In particular, we study how the decision boundary impacts the reconstruction properties of the algorithms. This approach has been implemented for the approximation and estimation of classifiers when the decision boundary is locally a smooth function [132] or is locally parametrized by a Barron function [128].

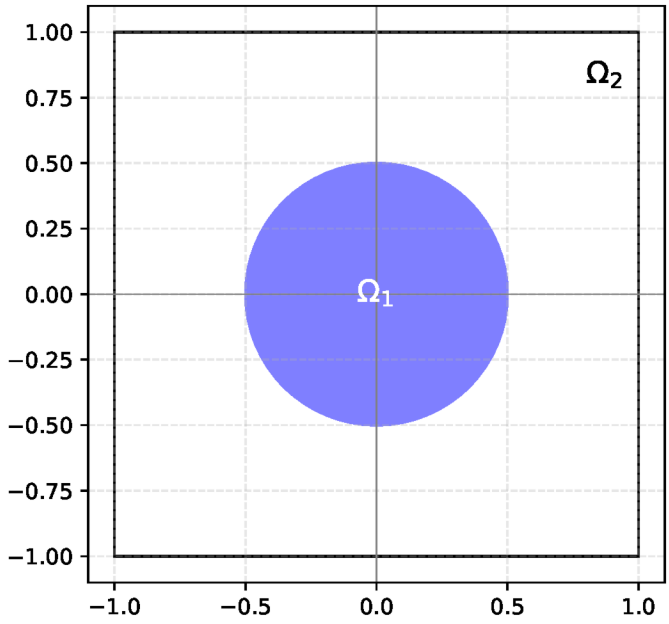


Figure 3.1: A classifier defined on $\Omega$ assigns the same level to each element of $\Omega_1$ which differs to the label assigned to each element in the set $\Omega_2$. .

## 3.1 Horizon Functions as Classifiers

Our analysis assumes that the classifiers belong to the class of horizon functions, as presented in Definition 3.1.1. However, we emphasize that our analysis can be extended to classifiers with an arbitrary number of finite labels. For simplicity, we denote the ball with radius one and center at the origin in $\mathbb{R}^d$ as $B_1^d$.

**Definition 3.1.1.** *[130] Let $d \in \mathbb{N}$, $d \geq 2$, $\Omega \subset \mathbb{R}^d$ and $\mathcal{B} \subset C(B_1^{d-1}, \mathbb{R})$. A horizon function $h_f$ associated to a function $f \in \mathcal{B}$ is a function $h_f : \Omega \to [0,1]$ of the form*

$$\mathbf{x} = (x_1, \ldots, x_d) \mapsto h_f := \mathbb{1}_{x_i \leq f(\mathbf{x}^{[i]})}$$

*where $\mathbf{x}^{[i]} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$ for $i \in [d]$. We also say that $h_f$ is* a horizon function with decision boundary described by $f \in \mathcal{B}$. *The* set of horizon functions associated to $\mathcal{B}$ *is defined by*

$$H_{\mathcal{B}} := \{h_f = \mathbb{1}_{x_i \leq h(\mathbf{x}^{[i]})} : h \in \mathcal{B}, i \in [d]\}.$$

Horizon functions are indicator functions whose *boundary* can be parametrized by a function from a given class which depends on all but one of the variables $x_i$, $i \in [d]$. Figure 3.3 illustrates the decision boundary of a horizon function on the square $[-1,1] \times [-1,1]$.

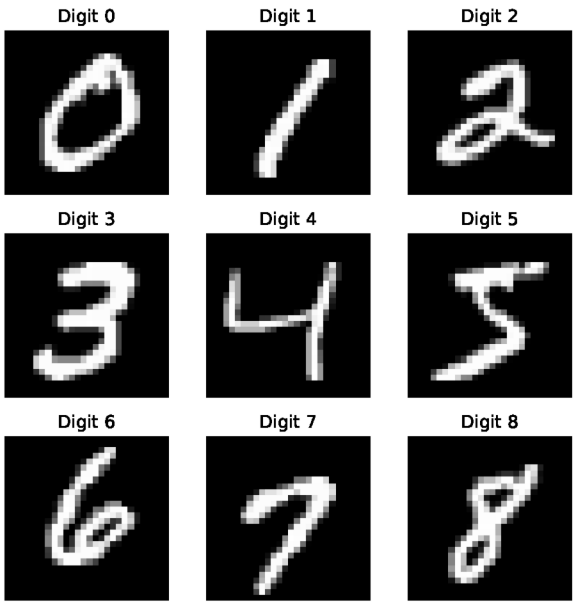


Figure 3.2: Some examples of digits from the MNIST dataset. These images are used for classification tasks involving digit recognition.

The decision boundary is the blue curve partitioning the unit square. Elements in different partition classes are assigned the labels 0 or 1.

Different approaches have been studied where a more general type of classifier is considered. In [128], the authors assume that all classifiers have decision boundaries locally parametrized by Horizon functions instead, i.e, there is a partition in rectangles of the domain $\Omega$ such that the classifier can be described by a Horizon function on each partition element. This approach could be an interesting line of research for future work.

The decision boundary of the classifier determines the approximation rates when neural-network-based methods are applied. For example, in [132], classifiers with boundary locally parametrized by $C^k$ functions are studied, and in [128], similar studies on locally parametrized by Barron functions are presented. Our approach builds upon some of the underlying ideas of these articles, but we consider the decision boundary to be parametrized by a function in the set of $RBV^2$ functions. This new assumption leads to approximation and estimation rates which do not suffer from the curse of dimensionality.

## 3.2 The Class of $RBV^2$ Functions

In this section, we introduce the functional space from which our horizon functions are parametrized: the set of $RBV^2$ functions. The setup of this section follows mainly the discussion of [122]. In that publication, the space of $RBV^2$ functions was associated with the study of the set of realizations of shallow NNs. The $R$ in $RBV^2$ represents the Radon transform and $BV$ represents the bounded variation, both concepts are involved in the

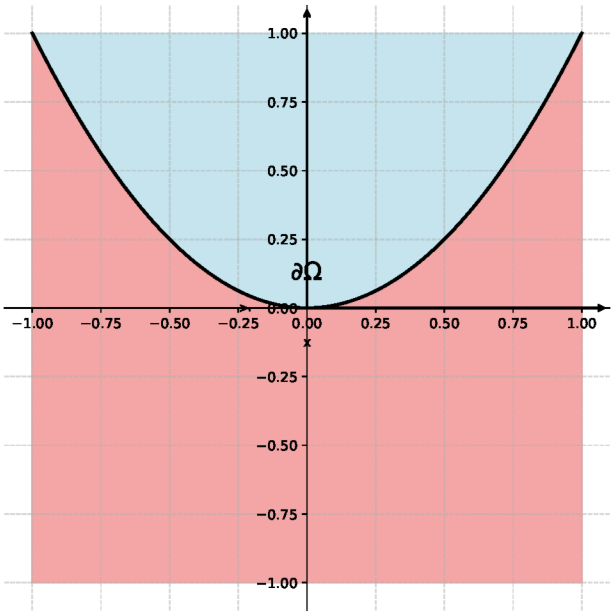


Figure 3.3: Partition of the unit square induced by a classifier with only labels. The decision boundary is a parabola.

definition of this functional space. Although this set has not received as much attention as the space of Barron functions, this functional space contains functions with high regularity such as Sobolev spaces and other functions with very low regularity as we will discuss later.

The $RBV^2$ space emerges as the suitable space for the solution of the following problem. Given a sample set $S = ((\mathbf{x}_i, y_i))_{i=1}^m$, we investigate if there exist a Banach space $\mathcal{X}$ and a seminorm $S_X : \mathcal{X} \to \mathbb{R}$ such that the functional

$$J(x) := \sum_{i=1}^{m} |f(\mathbf{x}_i) - y_i|^2 + S_X(f) \tag{3.2.1}$$

has a minimizer of the form

$$\Phi(x) := \sum_{k=1}^{N} v_k \varrho(\mathbf{w}_k^\top \mathbf{x} - b_k) + \mathbf{c}^\top x + c_0, \tag{3.2.2}$$

where $\mathbf{w}_k \in \mathbb{R}^d$ and $b_k \in \mathbb{R}$ for all $k = 1, \ldots, N$ and $\varrho$ denotes the ReLU activation function. The formulation in Equation 3.2.2 belongs to a class of special results usually referred to as *representer theorems* [45]. In other words, we seek a parametric formulation of the solutions of the variational optimization problem (3.2.1). In particular, we can observe that the solution of (3.2.1) is a linear combination of the elements of a *dictionary*. One can prove that there is a Banach space $\mathcal{X}$ where the minimizer of (3.2.1) exists and is the space of $RBV^2$ functions. We recall the concept of *Radon transform* to later define our functional space.

**Definition 3.2.1.** *[122] Let $d \in \mathbb{N}$. The* Radon transform *of the function $f : \mathbb{R}^d \to \mathbb{R}$, denoted as the function $\mathcal{R}f : \mathbb{S}^{d-1} \times \mathbb{R} \to \mathbb{R}$, is defined by*

$$\mathcal{R}f(\mathbf{w}, t) \coloneqq \int_{\{\mathbf{x}:\mathbf{w}^\top \mathbf{x}=t\}} f(\mathbf{x}) dS(\mathbf{x}),$$

*when this integral exists. Here, $(\mathbf{w}, t) \in \mathbb{S}^{d-1} \times \mathbb{R}$ and $dS$ indicates the surface integral over the domain $\{\mathbf{x} : \mathbf{w}^\top \mathbf{x} = t\}$.*

Now, we revisit the concept of *ramp filter*, a functional that can be found in the Radon inversion formula.

**Definition 3.2.2.** *Let $d \in \mathbb{N}$. The* ramp filter $\Lambda^{d-1}$ *is the functional defined as*

$$\Lambda^{d-1} \coloneqq (-\partial_t^2)^{(d-1)/2},$$

*where $\partial_t^2$ denotes the partial derivative with respect to $t$.*

The Radon transform and the ramp filter are used to induce the $RTV^2$-seminorm of a given function.

**Definition 3.2.3.** *[122] Let $d \in \mathbb{N}$. The* second-order total variation *or $RTV^2$-*seminorm *of a function $f : \mathbb{R}^d \to \mathbb{R}$ is defined by*

$$RTV^2_{\mathbb{R}^d}(f) \coloneqq \frac{1}{2(2\pi)^{d-1}} \|\partial_t^2 \Lambda^{d-1} \mathcal{R}(f)\|_{\mathcal{M}(\mathbb{S}^{d-1}\times\mathbb{R})},$$

*where $\|\cdot\|_{\mathcal{M}}$ denotes the total variation norm.*

We can see from Definition 3.2.3, that the $RTV^2$-seminorm promotes sparsity of the second derivatives in the Radon domain. Several functions fulfill this requirement without being highly-regular. Particularly, each translation of the ReLU activation function. We now formalize the concept of $RBV^2$ function.

**Definition 3.2.4.** *[123, Section 3] Let $d \in \mathbb{N}$. A function is called of $RBV^2-$regularity if its $RBV^2$-seminorm is*

$$RTV^2_{\mathbb{R}^d}(f) < \infty.$$

*We define the space of $RBV^2$ functions as*

$$RBV^2(\mathbb{R}^d) \coloneqq \{f \in L^{\infty,1} \colon RTV^2_{\mathbb{R}^d}(f) < \infty\},$$

*where $L^{\infty,1}$ denotes the set of all integrable functions $f$ such that $\sup_{x\in\mathbb{R}^d} |f(x)|(1+\|x\|_2)^{-1} < \infty$.*

The set of $RBV^2$ functions on $\mathbb{R}$ corresponds to the set of functions with second-derivative bounded in the distributional sense. That is, for $d = 1$,

$$RBV^2(\mathbb{R}) = \{f : \mathbb{R} \to \mathbb{R} : \|D^2 f\|_{\mathcal{M}(\mathbb{R})} < \infty\}.$$

We can prove that several functions of interest belong to $RBV^2$ for $d \geq 2$. For example, each function $f \in H^{d-1}(\mathbb{R}^d)$ is $RBV^2-$regular. Thus, functions with high regularity are also $RBV^2$. More interesting is the fact that functions with low regularity are also in the set of $RBV^2$ functions. Notice that for $\varrho$, the ReLU activation function, and $\mathbf{w} \in \mathbb{R}^d$, the function $\mathbf{x} \mapsto \varrho(\mathbf{w}^T\mathbf{x})$ is also in $RBV^2(\mathbb{R}^d)$. That means that the space of $RBV^2$ functions includes functions with high regularity and some with low regularity.

Now, we present a result that states the existence of a norm which makes the space of $RBV^2$ functions into a Banach space. The point evaluations in Definition 3.2.5 are well defined since $f$ is guaranteed to be Lipschitz continuous by [124, Lemma 2.11.].

**Proposition 3.2.5.** *[123, Section 3] Let $d \in \mathbb{N}$. The $RTV^2$-seminorm of a function $f \in RBV^2(\mathbb{R}^d)$ can be made into a norm as*

$$\|f\|_{RBV^2(\mathbb{R}^d)} \coloneqq RTV^2_{\mathbb{R}^d}(f) + |f(\mathbf{0})| + \sum_{k=1}^{d} |f(\mathbf{e}_k) - f(\mathbf{0})|,$$

*where $\{\mathbf{e}_k\}_{k=1}^d$ denotes the canonical basis of $\mathbb{R}^d$. Moreover, the space of $RBV^2-$ functions with this norm is a Banach space.*

The next natural step is to define the concept of an $RBV^2$-function results in arbitrary sets $\Omega \subset \mathbb{R}^d$. In fact, all the results of this chapter assume $f$ to be defined on the open unit ball of $\mathbb{R}^d$.

**Definition 3.2.6.** *[123, Section 4] Let $d \in \mathbb{N}$ and let $\Omega \subset \mathbb{R}^d$. We define the $RBV^2(\Omega)$ space as*

$$RBV^2(\Omega) \coloneqq \{f \in \mathcal{D}'(\Omega) \colon \exists g \in RBV^2(\mathbb{R}^d) \ s.t \ g|_\Omega = f\},$$

*where $\mathcal{D}'(\Omega)$ denotes the space of distributions on $\Omega$. Moreover, we define*

$$RTV^2_\Omega(f) \coloneqq \inf_{g \in RBV^2(\mathbb{R}^d) \colon f = g|_\Omega} RTV^2_{\mathbb{R}^d}(g)$$

*and*

$$\|f\|_{RBV^2(\Omega)} \coloneqq \inf_{g \in RBV^2(\mathbb{R}^d) \colon f = g|_\Omega} \|g\|_{RBV^2(\Omega)}.$$

The following result is essential to prove the approximability of $RBV^2$ functions on a finite domain by NNs. Basically, it states that there is an extension of each $RBV^2$-function on a finite domain to the whole Euclidean space.

**Lemma 3.2.7** ([123, Lemma 2]). *Let $d \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$ be a bounded set, and let $\varrho$ be the ReLU activation function. For every $f \in RBV^2(\Omega)$, there exists a function $F \in RBV^2(\mathbb{R}^d)$ such that*

$$F(\mathbf{x}) = \int_{\mathbb{S}^{d-1}\times\mathbb{R}} \varrho(\mathbf{w}^\top \mathbf{x} - b) d\mu(w, b) + \mathbf{c}^\top \mathbf{x} + c_0,$$

*for all $\mathbf{x} \in \mathbb{R}^d$, where $\mu \in \mathcal{M}(\mathbb{S}^{d-1} \times \mathbb{R})$ and the support of $\mu$ is contained in the set*

$$Z_\Omega := \overline{\{(\mathbf{w}, b) \in \mathbb{S}^{d-1} \times \mathbb{R} : \{\mathbf{x} : \mathbf{w}^\top \mathbf{x} = b\} \cap \Omega \neq \varnothing\}}.$$

*In addition, $F|_\Omega = f$ and $RTV^2_\Omega(f) = RTV^2_{\mathbb{R}^d}(F) = \|\mu\|_{\mathbb{S}^{d-1}\times\mathbb{R}}$.*

**Remark 3.2.8.** *As we restrict ourselves to the case $\Omega = B_1^d$, the unitary ball with center at the origin, we proceed to introduce a formula for $Z_\Omega$ (see [123, Lemma 2 and Remark 4])*

$$Z_\Omega = \mathbb{S}^{d-1} \times [-1, 1].$$

*Therefore, every function $f \in RBV^2(B_1^d)$ admits a representation of the form*

$$F(\mathbf{x}) = \int_{\mathbb{S}^{d-1}\times[-1,1]} \varrho(\mathbf{w}^\top \mathbf{x} - b) d\mu(\mathbf{w}, b) + \mathbf{c}^\top \mathbf{x} + c_0, \ \mathbf{x} \in B_1^d. \tag{3.2.3}$$

*Notice that, since $\mathbf{0}, \mathbf{e}_1, \ldots, \mathbf{e}_d \in B_1^d$ for every extension $F$ of $f$, it holds that $\|F\|_{RBV^2(\mathbb{R}^d)} = \|f\|_{RBV^2(B_1^d)}$. We now derive upper bounds for $\|\mathbf{c}\|_\infty$ and $|c_0|$ in terms of $\|f\|_{RBV^2}$. Assume that $\|f\|_{RBV^2} = C$ for some constant $C > 0$ and that $f(0) = 0$. Then we have the following equation,*

$$\int_{\mathbb{S}^{d-1}\times[-1,1]} \varrho(-b) d\mu(\mathbf{w}, b) + c_0 = 0.$$

*Since $0 \leq \varrho(-b) \leq 1$, and therefore bounded, it follows*

$$-C \leq \int_{\mathbb{S}^{d-1}\times[-1,1]} \varrho(-b) d\mu(\mathbf{w}, b) \leq C.$$

*Thus, it follows that $|c_0| \leq C$. Next, for the canonical basis $\{e_k\}_{k=1}^d \subset B_1^d$. By using the representation (3.2.3), we can compute $f(\mathbf{e_k})$ for all $k = 1, 2, \ldots, d$. Since $|f(\mathbf{e_k})| \leq \|f\|_{RBV^2} \leq C$, we obtain*

$$-C \leq \int_{\mathbb{S}^{d-1}\times[-1,1]} \varrho(w_k - b) d\mu(\mathbf{w}, b) + c_k + c_0 \leq C$$

*where $\mathbf{w}^T \mathbf{e_k} - b = w_k - b$ with $w_k$ is the $k$-th coordinate of the vector $\mathbf{w}$. Due to the fact $-2 \leq w_k - b \leq 2$ and by Lemma ´3.2.7, we know that the total variation of the measure $\mu$ is $\|\mu\|_{\mathbb{S}^{d-1}\times\mathbb{R}} = RTV^2(f) \leq \|f\|_{RBV^2}$, we conclude*

$$-2C \leq \int_{\mathbb{S}^{d-1}\times[-1,1]} \varrho(w_k - b) d\mu(\mathbf{w}, b) \leq 2C.$$

*From this, we obtain the following bound for*

$$-3C \leq \int_{\mathbb{S}^{d-1}\times[-1,1]} \varrho(w_k - b)d\mu(\mathbf{w}, b) + c_0 \leq 3C$$

*which leads to* $-4C \leq c_k \leq 4C$. *Finally, we can conclude the following inequalities for the coefficients*

$$\|c\|_\infty \leq 4\|f\|_{RBV^2} \text{ and } |c_0| \leq \|f\|_{RBV^2}.$$

### 3.2.1 A Practical Existence Theorem for the $RBV^2$ Space

Now, we proceed to present a practical existence theorem for the class of $RBV^2$ functions. This result is based on [123, Theorem 8] and [123, Theorem 13].

**Theorem 3.2.9.** *Let* $d \in \mathbb{N}$ *and let* $f \in RBV^2(B_1^d)$ *with* $\|f\|_{RVB^2} \leq M$. *Consider* $\{(\mathbf{x}_i, f(x_i) + \epsilon_i)\}_{i=1}^m$ *a data set with* $m \geq d+1$ *where* $\mathbf{x}_i \in [-1,1]^d$ *is drawn uniformly and independently at random, and* $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. *Then, there exists a class of shallow NNs, denoted* $\mathcal{N}$, *such that*

$$\tilde{\Psi} \in \operatorname*{arg\,min}_{\Psi\in\mathcal{N}} \frac{1}{m}\sum_{j=1}^{m} |f(\mathbf{x}_j) + \epsilon_i - \mathrm{R}(\Psi)(\mathbf{y}_j) - \eta_j|^2. \tag{3.2.4}$$

*Moreover, there exist constants* $C, C' > 0$ *depending on* $M$ *and* $d$ *such that*

$$\mathbb{E}(\|f - \mathcal{R}(\tilde{\Psi})\|^2) \leq C\left(\left(\frac{N}{\sigma^2}\right)^{-\frac{d+3}{2d+3}} + \left(\frac{N}{C'}\right)^{-\frac{d+3}{2d+3}}\right).$$

As we have already stated, Theorem 3.2.9 is a rearrangement of results in [123]. We have combined [123, Theorem 8] which guarantees the existence of an approximating NN for each $RBV^2$-function and [123, Theorem 11] which provides an estimate for the NN obtained by minimizing a functional when a sample set is available. However, we notice that we have found a few inconsistencies and some imprecision in the proof of the approximation result [123, Theorem 8]. Thus, we modify these statements and present a variation of its proof in [93]. Here, we rely on these results for the following theorem.

**Proposition 3.2.10.** *Let* $d \in \mathbb{N}$ *and* $f \in RBV^2(B_1^d)$. *Then, for every* $N \in \mathbb{N}$, *there is an NN* $\phi_f^N$ *with weights and biases* $\mathbf{w}_k \in \mathbb{R}^d$, $b_k, v_k \in \mathbb{R}$ $\mathbf{c} \in \mathbb{R}^d$ *and* $c_0 \in \mathbb{R}$, *such that,*

$$|v_k|, |b_k|, |c_0|, \|\mathbf{c}\|_2, \|\mathbf{w}_k\|_2 \leq 5\|f\|_{RBV^2},$$

*for* $k = 1, 2, \ldots, N$. *Then, the realization of* $\phi_f^N$

$$\mathrm{R}(\phi_f^N)(\mathbf{x}) \coloneqq \sum_{k=1}^{N} v_k \varrho(\mathbf{w}_k^\top \mathbf{x} - b_k) + \mathbf{c}^\top \mathbf{x} + c_0, \text{ for } \mathbf{x} \in B_1^d,$$

*satisfies the approximation bound*

$$\|f - \mathrm{R}(\phi_f^N)\|_{L^\infty(B_1^d)} \lesssim_d \|f\|_{RBV^2} N^{-\frac{d+3}{2d}}.$$

*Moreover* $\Phi_f^N \in \mathcal{NN}(d, 1, N+5, (2+d)(N+5)+1, 5\|f\|_{RBV^2})$.

Our proof of Proposition 3.2.10 begins with the integral representation of the function $f$ as given in Equation 3.2.3. Thanks to the properties of the absolute value function, we can decompose this integral into a sum of simpler integrals, which will become clear during the proof of Theorem 3.2.16. At this stage, we focus specifically on the integral

$$\int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| d\mu(\mathbf{w}, b), \text{ for } \mathbf{z} \in B_1^d. \tag{3.2.5}$$

To analyze this, we first apply a change of variables to the integral in (3.2.5), which helps establish the existence of a neural network. The domain of integration $\mathbb{S}^d \times [-1,1]$ is a cylinder with axis parallel to the axis $z_{d+1}$ of $\mathbb{R}^{d+1}$, radius $r = 1$ and lower and upper boundaries at $z_{d+1} = -1$ and $z_{d+1} = 1$, respectively. For simplicity, we first consider the case where $\mathbf{z} \in \mathbb{S}^{d-1}$. Notice that the integral term

$$\int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| d\mu(\mathbf{w}, b), \text{ for } \mathbf{z} \in \mathbb{S}^d, \tag{3.2.6}$$

can be rewritten by multiplying and dividing the integrand by $\sqrt{\|\mathbf{w}\|^2 + b^2} = \sqrt{1+b^2}$ as follows

$$\int_{\mathbb{S}^{d-1}\times[-1,1]} \sqrt{1+b^2} \left( \frac{|\mathbf{w}^\top \mathbf{z} - b|}{\sqrt{1+b^2}} \right) d\mu(\mathbf{w}, b). \tag{3.2.7}$$

The transformation of this integral into another integral over a domain contained within the sphere $\mathbb{S}^d \subset \mathbb{R}^{d+1}$ is presented in Lemma 3.2.11.

**Lemma 3.2.11.** *Let $d \in \mathbb{R}$, $z \in \mathbb{S}^d$, $\mathbf{w} = (w_1, w_2, \ldots, w_d)^\top \in \mathbb{S}^{d-1}$, and $b \in [-1,1]$. Further, let $\mathbf{v} = (v_1, v_2, \ldots, v_{d+1})^\top$ be defined as follows:*

$$v_i \coloneqq \frac{w_i}{\sqrt{1+b^2}},$$

*for $i = 1, 2, \ldots, d$ and*

$$v_{d+1} \coloneqq \frac{b}{\sqrt{1+b^2}}.$$

*Moreover, let $\phi : \mathbb{S}^{d-1} \times [-1,1] \to \mathbb{S}^d$ be the function defined as $\phi(\mathbf{w}, b) = (v_1, v_2, \ldots, v_{d+1})^\top$. Then, for all $\mathbf{z} \in \mathbb{S}^d$ we have*

$$\int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| d\mu(\mathbf{w}, b) = \int_{\mathbb{S}^d} \frac{1}{\sqrt{1 - v_{d+1}^2}} \left| \mathbf{v}^\top \widetilde{\mathbf{z}} \right| \chi_B(\mathbf{v}) d\phi^* \mu(\mathbf{v}),$$

*where $\widetilde{\mathbf{z}} = (\mathbf{z}, -1)^\top \in \mathbb{R}^{d+1}$, $B = \phi(\mathbb{S}^{d-1} \times [-1,1]) \subset \mathbb{S}^d \subset \mathbb{R}^{d+1}$, $\chi_B$ is the characteristic function of $B$, and $\phi^*\mu$ denotes the push-forward of the measure $\mu$.*

*Proof.* Starting from the integral in Equation (3.2.6), we rewrite it as:

$$\int_{\mathbb{S}^{d-1}\times[-1,1]} \sqrt{1+b^2}\left(\frac{|\mathbf{w}^\top \mathbf{z}-b|}{\sqrt{1+b^2}}\right) d\mu(\mathbf{w},b).$$

Using Equation (3.2.7), we express the integral as

$$\int_{\mathbb{S}^{d-1}\times[-1,1]} \sqrt{1+b^2}\left(\frac{|\mathbf{w}^\top \mathbf{z}-b|}{\sqrt{1+b^2}}\right) d\mu(\mathbf{w},b),$$

which can be rewritten as

$$\int_{\mathbb{S}^{d-1}\times[-1,1]} \sqrt{1+b^2}\left(\frac{|\langle(\mathbf{w},b),(z,-1)\rangle|}{\sqrt{1+b^2}}\right) d\mu(\mathbf{w},b).$$

From the definition of $v_{d+1}$, it follows

$$b^2 = \frac{v_{d+1}^2}{1-v_{d+1}^2}, \text{ and } 1+b^2 = 1+\frac{v_{d+1}^2}{1-v_{d+1}^2} = \frac{1}{1-v_{d+1}^2}.$$

Applying a *change of variables* (see for example [16, Section 19, Corollary 1]), the integral becomes

$$\int_{\mathbb{S}^{d-1}\times[-1,1]} \sqrt{1+b^2}\left(\frac{|\mathbf{w}^\top \mathbf{z}-b|}{\sqrt{1+b^2}}\right) d\mu(\mathbf{w},b) = \int_{\mathbb{S}^d} \frac{1}{\sqrt{1-v_{d+1}^2}} \left|\mathbf{v}^\top \widetilde{\mathbf{z}}\right| \chi_B(\mathbf{v}) d\phi^*\mu(\mathbf{v}),$$

where $\widetilde{\mathbf{z}}$ and $v_i$ are defined as in the lemma's statement. □

**Remark 3.2.12.** *Under the assumptions of Lemma 3.2.11, we make the following observations.*

1. *Since $b \in [-1,1]$, it follows that $2b^2 \leq b^2+1$ which implies $b^2/(b^2+1) \leq 1/2$. Taking the square root of both sides gives*

$$|v_{d+1}| \leq \frac{1}{\sqrt{2}}.$$

*Therefore, the set $B$ is a subset of the sphere for which the last coordinate lies between $-1/\sqrt{2}$ and $1/\sqrt{2}$.*

2. *We know that the function $h : B \subset \mathbb{S}^d \to \mathbb{R}$ defined by*

$$h(\mathbf{v}) = (1-v_{d+1}^2)^{-1/2}$$

*is continuous and thus integrable. This follows because the $\sigma$-algebra on $\mathbb{S}^d$ is the Borel $\sigma$-algebra, which is the restriction of the Borel $\sigma$-algebra of $\mathbb{R}^{d+1}$. In addition, we can bound $h(\mathbf{v})$. Indeed, from the previous step, we have*

$$v_{d+1}^2 \leq \frac{1}{2},$$

*which implies*

$$1 - v_{d+1}^2 \geq 1/2$$

*and*

$$\frac{1}{\sqrt{1 - v_{d+1}^2}} \leq \sqrt{2}.$$

3. *Let $C > 0$ the quantity*

$$C \coloneqq \int_{\mathbb{S}^d} \frac{1}{\sqrt{1 - v_{d+1}^2}} \chi_B(\mathbf{v}) d\phi^* \mu(\mathbf{v}),$$

*and let $\nu$ be the measure*

$$d\nu = \frac{1}{C\sqrt{1 - v_{d+1}^2}} \chi_B(\mathbf{v}) d\phi^* \mu.$$

*We notice that $\nu$ is a probability measure on the Borelian set of $\mathbb{S}^d$. Then, we can apply [149, Theorem 1] for the integral*

$$\int_{\mathbb{S}^d} \left| \mathbf{v}^\top \widetilde{\mathbf{z}} \right| d\nu(v),$$

*if $\widetilde{\mathbf{z}} \in \mathbb{S}^d$. That is, for all $r \in \mathbb{N}$, there is a set $Q = \{v^{(1)}, v^{(2)}, \ldots, v^{(r)}\} \subset \mathbb{S}^d$, real numbers $c_1, \ldots, c_r \in [0, 1]$ with $r \in \mathbb{N}$ and $c = c(d) > 0$ a constant, such that*

$$\left| \int_{\mathbb{S}^d} |\mathbf{v}^\top \mathbf{z}| d\nu(\mathbf{v}) - \sum_{i=1}^{r} c_i |((v^{(i)})^\top \mathbf{z})| \right| \leq c(d) r^{-(d+3)/(2d)}.$$

*Choosing*

$$r := \lceil (c(d)/\epsilon)^{2-6/(d+3)} \rceil \geq (c(d)/\epsilon)^{2-6/(d+3)},$$

*we have*

$$\left| \int_{\mathbb{S}^d} |\mathbf{v}^\top \mathbf{z}| d\nu(\mathbf{v}) - \sum_{i=1}^{r} c_i |((v^{(i)})^\top \mathbf{z})| \right| \leq c(d) r^{-(d+3)/(2d)} \leq c(d) \left[ (c(d)/\epsilon)^{-2+6/(d+3)} \right]^{(d+3)/2d} < \epsilon.$$

Remark 3.2.12 cannot be applied directly, because, in general $\widetilde{\mathbf{z}} \notin \mathbb{S}^d$. Notice $\mathbf{v} = (v_1, \ldots, v_{d+1})^\top \in \mathbb{S}^d \subset \mathbb{R}^{d+1}$. Indeed, for each $z \in \mathbb{S}^d$, the vector $\tilde{\mathbf{z}} = (\mathbf{z}, -1)$ belongs to $\sqrt{2}\mathbb{S}^d$. However, the result presented below shows that the conclusions drawn in Remark 3.2.12 can be generalized to spheres of any radius, allowing us to extend the argument for $\mathbf{z}$ in such spheres.

**Proposition 3.2.13.** *Let $t > 0$, $d \in \mathbb{N}$. Then, for $\epsilon > 0$, there is $\{v^{(1)}, v^{(2)}, \dots, v^{(r)}\} \subset \mathbb{S}^d$ and real numbers $c_1, \dots, c_r \in [0,1]$ where $r \in \mathbb{N}$, and $r \leq \lceil C(t,d)\epsilon^{-2+6/(d+3)} \rceil$ for $C = C(t,d) > 0$ a constant depending on the dimension, such that for all $\boldsymbol{z} \in t\mathbb{S}^d \subset \mathbb{R}^{d+1}$*

$$\left| \int_{\mathbb{S}^d} |\mathbf{v}^\top \mathbf{z}| d\nu(\boldsymbol{v}) - \sum_{i=1}^{r} c_i |((v^{(i)})^\top \boldsymbol{z})| \right| < \epsilon.$$

*Proof.* As $\mathbf{z} \in t\mathbb{S}^d$, there is a $\widetilde{z} \in \mathbb{S}^d$ such that $\mathbf{z} = t\widetilde{z}$. We now consider two cases. First, let us assume that $t > 1$. By Remark 3.2.12, for $\epsilon > 0$ there is a set $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(r)}\} \subset \mathbb{S}^d$ and real numbers $c_1, \dots, c_r \in [0,1]$ with $r \in \mathbb{N}$, $r \leq \lceil c(d)\epsilon^{-2+6/(d+3)} t^{2-6/(d+3)} \rceil$ such that

$$\left| \int_{\mathbb{S}^d} |\mathbf{v}^\top \widetilde{\mathbf{z}}| d\nu(\mathbf{v}) - \sum_{i=1}^{r} c_i |(\mathbf{v}^{(i)})^\top \widetilde{\mathbf{z}}| \right| < \epsilon/t.$$

Thus,

$$\left| \int_{\mathbb{S}^d} |\mathbf{v}^\top t\widetilde{\mathbf{z}}| d\mu(\mathbf{v}) - \sum_{i=1}^{r} c_i |(\mathbf{v}^{(i)})^\top t\widetilde{\mathbf{z}}| \right| < \epsilon.$$

Now, if $t < 1$

$$\left| \int_{\mathbb{S}^d} |\mathbf{v}^\top t\widetilde{\mathbf{z}}| d\mu(v) - \sum_{i=1}^{r} c_i |(\mathbf{v}^{(i)})^\top t\widetilde{\mathbf{z}}| \right| = t \left| \int_{\mathbb{S}^d} |\mathbf{v}^\top \mathbf{z}| d\mu(\mathbf{v}) - \sum_{i=1}^{r} c_i |(\mathbf{v}^{(i)})^\top z| \right| < t\epsilon < \epsilon.$$

Therefore, in any case $r \leq \lceil C(d)\epsilon^{-2+6/(d+3)} \max\{1,t\}^{2-6/(d+3)} \rceil = \lceil C(d,t)\epsilon^{-2+6/(d+3)} \rceil$. □

**Remark 3.2.14.** *We would like to highlight that the previous result holds in particular for $t = \sqrt{2}$. As $(\boldsymbol{z}, -1) \in \sqrt{2}\mathbb{S}^d$ when $\boldsymbol{z} \in \mathbb{S}^d$, this case is especially relevant. That being said, we proceed to approximate the term* (3.2.5)*. For a given element $v^{(i)} \in B \subset \mathbb{S}^d$ as in Proposition 3.2.13, we define $\boldsymbol{w}^{(i)} = (v_1^{(i)}, v_2^{(i)}, \dots, v_d^{(i)})$ and $b_i = v_{d+1}^{(i)}$ where $v_k^{(i)}$ denotes the $k-th$ component of $v^{(i)}$ for all $i = 1, 2, \dots r$. It is clear that $\sup_{k=1,\dots,d+1} |w_k^{(i)}| \leq 1$ and $|b_i| \leq 1$. Therefore, for $z \in \mathbb{S}^d$*

$$\left| \int_{\mathbb{S}^d} |\boldsymbol{w}^\top \boldsymbol{z} - b| d\mu(\boldsymbol{v}) - \sum_{i=1}^{r} c_i |(\boldsymbol{w}^{(i)})^\top \boldsymbol{z} - b_i| \right| < \epsilon.$$

Now, we are set to prove that there is an NN that approximates the integral term of Equation 3.2.3. We first assume that $\mu$ of (3.2.3) is a probability measure.

**Lemma 3.2.15.** *Let $d \in \mathbb{N}$, and let $\mu$ be a probability measure over $\mathbb{S}^{d-1} \times [-1,1]$. Then, let $f : B_d^1 \to \mathbb{R}$ be given by*

$$f(\boldsymbol{z}) = \int_{\mathbb{S}^{d-1} \times [-1,1]} \varrho(\boldsymbol{w}^\top \boldsymbol{z} - b) d\mu(\boldsymbol{w}, b),$$

*where* $\boldsymbol{z} \in B_d^1$. *Then, there exist vectors* $\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \dots \boldsymbol{w}^{(r)} \in \mathbb{R}^d$, *real numbers* $b_1, b_2, \dots b_r$ *and* $c_1, \dots, c_r$ *with* $r \in \mathbb{N}$, *such that for all* $\boldsymbol{z} \in \mathbb{S}^{d-1}$

$$\left| \int_{\mathbb{S}^{d-1}\times[-1,1]} |\boldsymbol{w}^\top \boldsymbol{z} - b| d\mu(\boldsymbol{w}, b) - \sum_{i=1}^{r} c_i \varrho((\boldsymbol{w}^{(i)})^\top \boldsymbol{z} - b_i) - c_i \varrho(-(\boldsymbol{w}^{(i)})^\top \boldsymbol{z} + b_i) \right| < \epsilon,$$

*where* $\sup_{k=1,\dots,d+1} |\boldsymbol{w}_k^{(i)}| \leq 1$, $|b_i| \leq 1$ *and* $c_i \in [0,1]$, $i = 1, 2, \dots r$, *where* $r \leq \lceil C(d)\epsilon^{-2+6/(d+3)} \rceil$.

*Proof.* By Remark 3.2.14 there are $\mathbf{w}^{(i)} \in [-1,1]^d$ and $b_i, c_i \in [-1,1]$ such that

$$\left| \int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| d\mu(\mathbf{v}) - \sum_{i=1}^{r} c_i |(\mathbf{w}^{(i)})^\top \mathbf{z} - b_i| \right| < \epsilon,$$

where $r \leq \lceil C(d)\epsilon^{-2+6/(d+3)} \rceil$. As $|x| = \varrho(x) - \varrho(-x)$, we have

$$\sum_{i=1}^{r} c_i |(\mathbf{w}^{(i)})^\top \ \mathbf{z} - b_i| = \sum_{i=1}^{r} c_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{z} - b_i) - c_i \varrho(-(\mathbf{w}^{(i)})^\top \mathbf{z} + b_i), \tag{3.2.8}$$

and hence

$$\left| \int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| d\mu(w, b) - \sum_{i=1}^{r} c_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{z} - b_i) - c_i \varrho(-(\mathbf{w}^{(i)})^\top \mathbf{z} + b_i) \right| < \epsilon.$$

□

We now have all the ingredients to prove Theorem 3.2.16. In the proof of the result, we use Lemma 3.2.7 twice when (3.2.3) is expressed as the addition of three terms. We consider the case when $\mu$ is a probability measure. The case of general measures will be shown in a corollary thereafter.

**Theorem 3.2.16.** *Let* $d \in \mathbb{N}$ *and* $\mu \in \mathcal{M}(\mathbb{S}^{d-1} \times [-1,1])$ *be a probability measure. Let* $g : B_1^d \to \mathbb{R}$ *be the function defined as*

$$g(\mathbf{z}) = \int_{\mathbb{S}^{d-1}\times[-1,1]} \varrho(\mathbf{w}^\top \mathbf{z} - b) d\mu(\mathbf{w}, b), \ \textit{for } \mathbf{z} \in B_1^d.$$

*Then, for all* $\epsilon > 0$, $g$ *can be uniformly approximated with approximation error* $\epsilon$ *by a shallow NN*

$$\widetilde{f}(\mathbf{z}) = \sum_{i=1}^{K} v_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{z} - b_i) + (\mathbf{w}^{(0)})^\top \mathbf{z} + b_0,$$

*where* $w^{(0)}, \mathbf{w}^{(i)} \in \mathbb{R}^d$, $b_0, b_i \in \mathbb{R}$, $v_i \in \mathbb{R}$ *and*

$$\|\mathbf{w}^{(0)}\|_\infty, \|\mathbf{w}^{(i)}\|_\infty, |v_i|, |b_i|, |b_0| \leq 1,$$

*for all* $i = 1, 2, \dots, K$ *where* $K \leq 4\lceil C(d)\epsilon^{-2+6/(d+3)} \rceil$.

*Proof.* Because the ReLU activation function can be expressed as $\varrho(x) = (x + |x|)/2$ and due to the following argument in [12, Proposition 1], we have that

$$\begin{aligned} g(\mathbf{z}) &= \frac{1}{2}\int_{\mathbb{S}^{d-1}\times[-1,1]} (\mathbf{w}^\top \mathbf{z} - b) d\mu(\mathbf{w}, b) + \frac{1}{2}\int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| d\mu(\mathbf{w}, b) \\ &= \frac{1}{2}\int_{\mathbb{S}^{d-1}\times[-1,1]} (\mathbf{w}^\top \mathbf{z} - b) d\mu(\mathbf{w}, b) \\ &\quad + \frac{\mu_+(\mathbb{S}^{d-1}\times[-1,1])}{2}\int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| \frac{d\mu_+(\mathbf{w}, b)}{\mu_+(\mathbb{S}^{d-1}\times[-1,1])} \\ &\quad - \frac{\mu_-(\mathbb{S}^{d-1}\times[-1,1])}{2}\int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| \frac{d\mu_-(\mathbf{w}, b)}{\mu_-(\mathbb{S}^{d-1}\times[-1,1])}, \end{aligned} \tag{3.2.9}$$

for $z \in \mathbb{S}^d$ and $\mu = \mu_+ + \mu_-$ is a Jordan decomposition. Furthermore, it was proved in Lemma 3.2.15 that for $\epsilon > 0$ and probability measure $\mu \in \mathcal{M}(\mathbb{S}^{d-1}\times[-1,1])$, there is an $r \in \mathbb{N}$ and $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(r)} \in \mathbb{R}^d$, $b_1, b_2, \ldots, b_r \in [-1,1]$, $c_1, \ldots, c_r \in [0,1]$ such that

$$\left| \int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| d\mu - \sum_{i=1}^{r} c_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{z} - b_i) - c_i \varrho(-(\mathbf{w}^{(i)})^\top \mathbf{z} + c_i) \right| < \epsilon,$$

for all $\mathbf{z} \in B_1^d$ and $r \leq \lceil C(d)\epsilon^{-2+6/(d+3)} \rceil$ where $C = C(d) > 0$ is a constant depending on the dimension. Due to the fact that

$$\frac{d\mu_+}{\mu_+(\mathbb{S}^{d-1}\times[-1,1])} \text{ and } \frac{d\mu_-}{\mu_-(\mathbb{S}^{d-1}\times[-1,1])}$$

are probability measures, we can approximate each of the last two terms of Equation 3.2.9 according to Lemma 3.2.15 as

$$\left| \int_{\mathbb{S}^{d-1}\times[-1,1]} |\mathbf{w}^\top \mathbf{z} - b| d\mu_\pm - \mu_{s,\pm} \sum_{i=1}^{r} c_{\pm,i} \varrho((\mathbf{w}_\pm^{(i)})^\top \mathbf{z} - b_{\pm,i}) - c_{\pm,i} \varrho(-(\mathbf{w}_\pm^{(i)})^\top \mathbf{z} + b_{\pm,i}) \right| < \epsilon \mu_{s,\pm} \leq \epsilon,$$

respectively, where $\mu_{s,\pm} = \mu_\pm(\mathbb{S}^{d-1}\times[-1,1])) \leq 1$ and $\mathbf{w}_+^{(i)}, c_{+,i}$ and $b_{+,i}$ denotes the weights and biases when the measure involved is $\mu_+$ and $\mathbf{w}_-^{(i)}, c_{-,i}$ and $b_{-,i}$ are defined likewise for $\mu_-$. Moreover, the first term of (3.2.9) can be expressed as

$$\frac{1}{2}\int_{\mathbb{S}^{d-1}\times[-1,1]} (\mathbf{w}^\top \mathbf{z} - b) d\mu(w, b) = (\mathbf{w}^{(0)})^\top \mathbf{z} - b_0,$$

where $\|\mathbf{w}^{(0)}\|_\infty \leq 1$ and $|b_0| \leq 1$. Setting

$$\begin{aligned} \widetilde{f}(\mathbf{z}) &= (\mathbf{w}^{(0)})^\top \mathbf{z} - b_0 + \frac{\mu_{s,+}}{2} \sum_{i=1}^{r} c_{+,i} \varrho((\mathbf{w}_+^{(i)})^\top \mathbf{z} - b_{+,i}) - c_{+,i} \varrho(-(\mathbf{w}_+^{(i)})^\top \mathbf{z} + b_{+,i}) \\ &\quad + \frac{\mu_{s,-}}{2} \sum_{i=1}^{r} c_{-,i} \varrho((\mathbf{w}_-^{(i)})^\top \mathbf{z} - b_{-,i}) - c_{-,i} \varrho(-(\mathbf{w}_-^{(i)})^\top \mathbf{z} + b_{-,i}), \end{aligned}$$

and $K = 4r$, we can rearrange $\widetilde{f}$ in such a way that

$$\widetilde{f}(\mathbf{z}) = \sum_{i=1}^{K} v_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{z} - b_i) + (\mathbf{w}^{(0)})^\top \mathbf{z} + b_0,$$

where $v_i$ is either $\mu_{s,+}/(2r)$ or $\mu_{s,-}/(2r)$ depending on whether $w^{(i)}$ was determined by $\mu_{s,+}$ or $\mu_{s,-}$. Notice that, by construction $K \leq 4\lceil C(d)\epsilon^{-2+6/(d+3)} \rceil$ for $C = C(d) > 0$ a constant. Additionally, all weights and biases are bounded in norm by 1,

$$|v_i|, |b_i|, |b_0|, \|\mathbf{w}^{(i)}\|_\infty, \|\mathbf{w}^{(0)}\|_\infty \leq 1$$

for $i = 1, 2, \ldots, K$. To conclude, notice that by construction $\|g - \widetilde{f}\|_\infty < \epsilon$. □

We now proceed by generalizing Theorem 3.2.16 to arbitrary measures.

**Corollary 3.2.17.** *Let $d \in \mathbb{N}$, and let $f \in RBV^2(B_1^d)$ with $f(0) = 0$. Then, for all $\epsilon > 0$, there exist $\mathbf{w}^{(i)} \in \mathbb{R}^d$, $b_i, v_i \in \mathbb{R}$, $c \in \mathbb{R}^d$ and $c_0 \in \mathbb{R}$ with*

$$|v_i|, |b_i|, |c_0|, \|c\|_\infty, \|\mathbf{w}^{(i)}\|_\infty \leq 5\|f\|_{RBV^2},$$

*where $i = 1, 2, \ldots, K$ and $K \leq 4\lceil C(d)\|f\|_{RBV^2}^{2-6/(d+3)} \epsilon^{-2+6/(d+3)} \rceil$, such that $f$ can be uniformly approximated with approximation error $\epsilon$ by a shallow NN*

$$f_K(\mathbf{x}) = \sum_{i=1}^{K} v_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{x} - b_i) + \mathbf{c}^\top \mathbf{x} + c_0, \quad \text{for } \mathbf{x} \in \mathbb{R}^d.$$

*Proof.* We assume first that $\|f\|_{RBV^2} = 1$. By Remark 3.2.8, we know that $f$ has an integral representation of the form

$$f(\mathbf{x}) = \int_{\mathbb{S}^{d-1} \times [-1,1]} \varrho(\mathbf{w}^\top \mathbf{x} - b) d\mu(\mathbf{w}, b) + \widetilde{\mathbf{c}}^\top \mathbf{x} + \widetilde{c}_0, \quad \text{for all } \mathbf{x} \in B_d^1,$$

where $\|\widetilde{c}\|_\infty \leq 4$ and $|\widetilde{c}_0| \leq 1$ and $\mu \in \mathcal{M}(\mathbb{S}^{d-1} \times [-1,1])$ such that $\|\mu\|_{\mathbb{S}^{d-1} \times [-1,1]} = RTV^2(f)$.

If $RTV^2(f) = 0$ the result holds, since $f(\mathbf{x}) = \widetilde{\mathbf{c}}^T \mathbf{x}$ in this case. If $RTV^2(f) \neq 0$, we define $\hat{f} \coloneqq f/RTV^2(f)$ and choose $\epsilon > 0$.

According to Theorem 3.2.16, the function

$$x \mapsto \frac{1}{RTV^2(f)} \int_{\mathbb{S}^{d-1} \times [-1,1]} \varrho(\mathbf{w}^\top \mathbf{x} - b) d\mu(\mathbf{w}, b)$$

can be approximated by the realization of an NN

$$\widetilde{f}_K(\mathbf{x}) = \sum_{i=1}^{K} \widetilde{v}_i \varrho((\widetilde{\mathbf{w}}^{(i)})^\top \mathbf{x} - \widetilde{b}_i) + (\widetilde{\mathbf{w}}^{(0)})^\top \mathbf{x} + \widetilde{b}_0, \text{ for } \mathbf{x} \in \mathbb{R}^d$$

with approximation error $\epsilon/RTV^2(f)$ and where $K \leq \lceil C(d)RTV^2(f)^{2-6/(d+3)}\epsilon^{-2+6/(d+3)}\rceil$ for a constant $C = C(d) > 0$ depending on $d$. This holds because the measure of the integrand $\nu = \mu/RTV^2(f)$ fulfills that $\|\nu\|_{\mathbb{S}^{d-1}\times[-1,1]} = 1$. Moreover, for all $i = 1, 2, \ldots, K$,

$$|\widetilde{v}_i|, |\widetilde{b}_i|, |\widetilde{b}_0|, \|\widetilde{\mathbf{w}}^{(i)}\|_\infty, \|\widetilde{\mathbf{w}}^{(0)}\|_\infty \leq 1.$$

Let us define

$$\widehat{f}_K \coloneqq \widetilde{f}_K + \mathbf{a}^\top \mathbf{x} + a_0.$$

where $a = \widetilde{c}/RTV^2(f)$, $a_0 = \widetilde{c}_0/RTV^2(f)$. This leads us to the fact that

$$|\widehat{f}(\mathbf{x}) - \widehat{f}_K(\mathbf{x})| = \left| \int_{\mathbb{S}^{d-1}\times[-1,1]} \varrho(\mathbf{w}^\top \mathbf{x} - b) d\nu(\mathbf{w}, b) - \widetilde{f}_K(\mathbf{x}) \right| < \epsilon/RTV^2(f)$$

for all $\mathbf{x} \in B_1^d$. Finally, if we set

$$f_K(\mathbf{x}) = RTV^2(f)\widehat{f}(\mathbf{x}) = \sum_{i=1}^{K} v_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{x} - b_i) + (\mathbf{w}^{(0)} + \widetilde{\mathbf{c}})^\top \mathbf{x} + (b_0 + \widetilde{c}_0),$$

where $\mathbf{w}^{(0)} = RTV^2(f)\widetilde{\mathbf{w}}^{(0)}$, $b_0 = RTV^2(f)\widetilde{b}_0$ and for all $i = 1, 2, \ldots, K$ we choose $\mathbf{w}^{(i)} = RTV^2(f)\widetilde{\mathbf{w}}^{(i)}$, $v_i = RTV^2(f)\widetilde{v}_i$ and $b_i = RTV^2(f)\widetilde{b}_i$, it holds that

$$|f(\mathbf{x}) - f_K(\mathbf{x})| = |RTV^2(f)\widehat{f}(\mathbf{x}) - RTV^2(f)\widehat{f}_K(\mathbf{x})| = RTV^2(f)|\widehat{f}(\mathbf{x}) - \widehat{f}_K(\mathbf{x})| < \epsilon.$$

We denote $c = w^{(0)} + \widetilde{c}$ and $c_0 = b_0 + \widetilde{c}_0$ and observe that $|c_0| \leq |b_0| + |\widetilde{c}_0| \leq 1 + \|f\|_{RBV^2} \leq 2$. As $\|\widetilde{c}\|_\infty \leq 4$, we conclude that $\|c\|_\infty \leq 5$. In combination with the bounds for the weights in Theorem 3.2.16, we obtain

$$|v_i|, |b_i|, |\widetilde{c_0}|, \|\widetilde{\mathbf{c}}\|_2, \|\mathbf{w}^{(i)}\| \leq 5.$$

For the general case where $\|f\|_{RBV^2} \neq 1$, we define

$$\hat{f} \coloneqq \frac{f}{\|f\|_{RBV^2}},$$

if $\|f\|_{RBV^2} \neq 0$ and apply the previous argument to conclude that for every $\epsilon > 0$ there is a $NN$ such that $\|\widehat{f} - \widehat{f}_K\| < \epsilon/\|f\|_{RBV^2}$ where $K \leq 4\lceil C(d)\|f\|_{RBV^2}^{2-6/(d+3)}\epsilon^{-2+6/(d+3)}\rceil$ with all weights bounded by 5. If we denote $f_K = \|f\|_{RBV^2}\widehat{f}_K$, we observe that $\|f - f_K\|_\infty < \epsilon$. The boundedness of the weights follows immediately. If $\|f\|_{RBV^2} = 0$, $f = 0$ and the result holds trivially.

□

We now have all the results to prove Proposition 3.2.10. Essentially, we only need to express the approximation accuracy $\epsilon$ in terms of the number of neurons of a neural network.

*Proof of Proposition 3.2.10.* For $N \in \mathbb{N}$, we set

$$\epsilon = \left(4C(d)\|f\|_{RBV^2}^{2-6/(d+3)}/N\right)^{(d+3)/(2d)}.$$

Then, according to Corollary 3.2.17, there is $\{\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(r)}\} \subset [-1,1]^d$, $\{v_1, v_2, \dots, v_r\} \subset [-1,1]$ and $\{b_1, b_2, \dots, b_r\} \subset [-1,1]$ such that if we set

$$f_r(x) = \sum_{i=1}^{r} v_i \varrho((\mathbf{w}^{(i)})^\top z - b_i) + \mathbf{c}^T\mathbf{x} + c_0,$$

it holds that

$$|f(\mathbf{x}) - f_r(\mathbf{x})| = \left| \int_{\mathbb{S}^d} \varrho(\mathbf{w}^\top \mathbf{x} - b) d\mu(\mathbf{w}, b) - \sum_{i=1}^{r} v_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{x} - b_i) \right| < \epsilon,$$

where $r \leq 4\lceil C(d)\|f\|_{RBV^2}^{2-6/(d+3)} \epsilon^{-2+6/(d+3)} \rceil < N + 4$. for all $\mathbf{x} \in B_1^d$. We use the fact $\lceil \mathbf{x} \rceil < x + 1$. We set $\mathbf{w}^{(r+1)} = \cdots = \mathbf{w}^{(N+3)} = 0 \in \mathbb{R}^d$, $v_{r+1} = \cdots = v_{N+3} = 0 \in \mathbb{R}$ and $b_{r+1} = \cdots = b_{N+3} = 0 \in \mathbb{R}$ and obtain

$$\left| \int_{\mathbb{S}^d} \varrho(\mathbf{w}^\top \mathbf{x} - b) d\mu(w, b) - \sum_{i=1}^{N+3} v_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{x} - b_i) \right| < \epsilon$$

for all $\mathbf{x} \in B_1^d$. Then, if

$$\mathrm{R}(\Phi_N^f)(\mathbf{x}) = \sum_{i=1}^{N+3} v_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{x} - b_i) + \mathbf{c}^T\mathbf{x} + c_0,$$

the result follows. Finally, notice that $\mathbf{c}^\top \mathbf{x} = \varrho(\mathbf{c}^T\mathbf{x}) - \varrho(-\mathbf{c}^T\mathbf{x})$. Hence, $\mathrm{R}(\Phi_N^f)$ can be expressed as

$$\mathrm{R}(\Phi_N^f)(\mathbf{x}) = \sum_{i=1}^{N+3} v_i \varrho((\mathbf{w}^{(i)})^\top \mathbf{x} - b_i) + c_0,$$

and after counting the number of weights we conclude that

$$\Phi_N^f \in \mathcal{NN}(d, 2, N+5, (2+d)(N+5) + 1, 5\|f\|_{RBV^2}).$$

□

## 3.3 Practical Existence Theorem for Horizon Functions

Here, we present a practical existence theorem for the estimation of horizon functions with $RBV^2$ boundary by NNs. To this end, we demonstrate first the existence of an approximating

NN for each horizon function with $RBV^2$ boundary by NNs. Then, we proceed to study the problem of estimating this kind of functions from a fixed data set.

Although the 0-1 loss is traditionally used in classification settings, this is not a useful error measure in practice, due to its lack of continuity. Instead, we formulate our learning bounds with respect to the Hinge loss, which is significantly more convenient in applications. As we consider horizon functions $h : [0,1]^d \to \{0,1\}$ as our target classifiers whose range is not precisely the set $\{-1,1\}$ as would be typical for the Hinge loss, we present a slightly different definition of Hinge function.

**Definition 3.3.1.** *[128, Definition 5.5] Define $\phi\colon \mathbb{R} \to \mathbb{R}$, $\phi(x) \coloneqq \max\{0, 1-x\}$. Let $d \in \mathbb{N}$ and let $\mu$ be a Borel measure on $B_1^d$. For a (measurable) target concept $h^*\colon B_1^d \to \{0,1\}$, we define the* Hinge risk *of a (measurable) function $h\colon B_1^d \to [0,1]$ as*

$$\mathcal{E}_{\phi,\mu,h^*}(h) \coloneqq \mathbb{E}_{X\sim\mu}\Big[\phi\Big(\big(2h^*(X)-1\big)\cdot\big(2h(X)-1\big)\Big)\Big].$$

*Let $\Lambda = B_1^d \times [0,1]$. For a sample $S = (\mathbf{x}_i, y_i)_{i=1}^m \in \Lambda^m$, $m \in \mathbb{N}$, we define the* empirical $\phi$-risk *of $h\colon B_1^d \to [0,1]$ as*

$$\widehat{\mathcal{E}}_{\phi,S}(h) \coloneqq \frac{1}{m}\sum_{i=1}^m \phi\big(\big(2y_i-1\big)\cdot\big(2h(\mathbf{x}_i)-1\big)\big).$$

*Finally, for a sample $S = (\mathbf{x}_i, y_i)_{i=1}^m \in \Lambda^m$ and a set $\mathcal{H} \subset \{h\colon B_1^d \to [0,1]\}$, we call $h_S \in \mathcal{H}$ an empirical $\phi$-risk minimizer, if*

$$\widehat{\mathcal{E}}_{\phi,S}(h_S) = \min_{h\in\mathcal{H}} \widehat{\mathcal{E}}_{\phi,S}(h). \tag{3.3.1}$$

In the following PET, the NN which estimates the horizon function is obtain by minimizing the empirical $\phi$-risk.

**Theorem 3.3.2.** *Let $d \in \mathbb{N}$ and let $\boldsymbol{\lambda}$ be a probability measure on $B_1^{d+1}$. Then, there is a class of NNs denoted by $\mathcal{F}$ such that for all $h \in H_{\mathcal{B}}$, where $\mathcal{B} \coloneqq \{f \in RBV^2(B_1^d)\colon f(0) = 0 \quad \|f\|_{RBV^2} \le q\}$ and for each $m \in \mathbb{N}$, let $S$ be a training sample of sizem; that is, $S = \big(X_i, h(X_i)\big)_{i=1}^m$ with $X_i \overset{iid}{\sim} \boldsymbol{\lambda}$ for $i \in \{1,\dots,m\}$ the problem*

$$\widehat{\mathcal{E}}_{\phi,S}\big(\Phi_{m,S}\big) = \min_{f\in\mathcal{F}} \widehat{\mathcal{E}}_{\phi,S}(f)$$

*has a solution. Moreover, if $H \coloneqq \mathbb{1}_{[1/2,\infty)}$, every minimizer satisfies*

$$\mathbb{E}_S\Big[\mathbb{E}_{X\sim\boldsymbol{\lambda}}\big(\big|H(\Phi_{m,S}(X)) - h(X)\big|^2\big)\Big] = \mathbb{E}_S\big[\mathbb{P}_{X\sim\boldsymbol{\lambda}}\big(H(\Phi_{m,S}(X)) \neq h(X)\big)\big] \lesssim m^{-\frac{d+3}{3d+3}+\kappa}, \tag{3.3.2}$$

*where the implied constant only depends on $d, \kappa, \tau$.*

Notice that Theorem 3.3.2 only guarantees the existence of a class of NNs from which we can estimate each horizon function with boundary parametrized by a $RBV^2$ function. We obtain upper bounds for the number of layers, neurons and weights of this class of NNs in the following sections. This result follows from the approximation result Theorem 3.3.4 and estimation bound 3.3.6. The class of NNs $\mathcal{F}$ is explicitly stated in Theorem 3.3.6.

### 3.3.1 Approximation of Horizon Functions by Neural Networks

When faced with the task of reconstructing an approximation function, if we use continuous functions, we can not measure the approximation error with the $L^\infty$ norm. As the kind of NNs we use in this dissertation are continuous, we need to introduce a different way to do it. The notion of *tube compatible measure* helps us in this task.

**Definition 3.3.3.** *[130, Definition 3.5] Let $d \in \mathbb{N}$. Let $\mu$ be a finite Borel measure on $\Omega \subset \mathbb{R}^d$. We say that $\mu$ is a* tube compatible measure with parameters $\alpha \in (0, 1]$ and $C > 0$ *if for each measurable function $f\colon \Omega \to \mathbb{R}$, each $i \in [d]$ and each $\epsilon \in (0, 1]$,*

$$\mu(T^i_{f,\epsilon}) \leq C\epsilon^\alpha \text{ where } T^i_{f,\epsilon} \coloneqq \{\mathbf{x} \in \mathbb{R}^d \colon |x_i - f(\mathbf{x}^{[i]})| \leq \epsilon\}, \tag{3.3.3}$$

*where each set $T^i_{f,\epsilon}$ is called a* tube of width $\epsilon$ (associated to $f$).

In Figure 3.4, we illustrate the sets $T_{f,\epsilon}$ in the case $d = 2$. Notice that the volume of a tube around the graphic of $f$ decays exponentially with parameter $\alpha > 0$ in terms of $\epsilon$. A crucial instance of tube compatible measure is the Lebesgue measure, with parameter $\alpha = 1$ and $C = 1$. We notice that the product of tube-compatible measures is also a tube compatible measure. Moreover, each continuous measure with respect to a tube compatible measure belongs to the set of tube compatible measures as well.

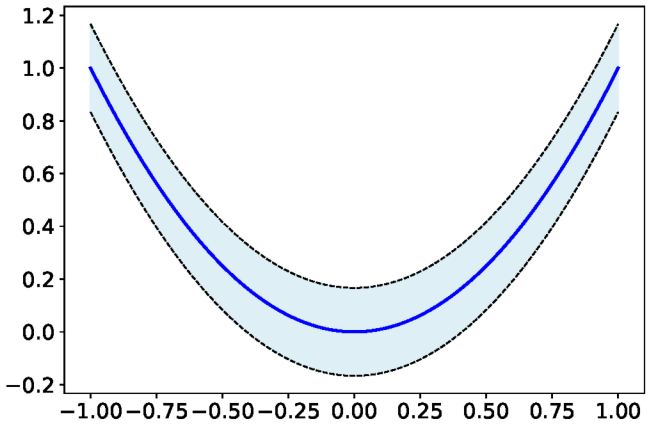


Figure 3.4: The graphic of a real-valued function $f$ and a visualization of a tube $T^1_{f,0.2}$.

The next results shows the existence of an approximating NN for each horizon function with a boundary in the space of $RBV^2$ functions. The proof follows the scheme of [128], but uses $RBV^2$ functions instead of Barron. Under this assumption, we achieve a slightly faster approximation rate.

**Theorem 3.3.4.** *Let $d \in \mathbb{N}_{\geq 2}$, $N \in \mathbb{N}$, $q, C > 0$, and $\alpha \in (0,1]$. Further let, $h \in H_{\mathcal{B}}$, where $\mathcal{B} \coloneqq \{f \in RBV^2(B_1^d) \colon f(0) = 0, \quad \|f\|_{RBV^2(B_1^d)} \leq q\}$.*
*There exists an NN $I_N$ with two hidden layers such that for each tube compatible measure $\mu$ with parameters $\alpha, C$, we have*

$$\mu(\{\mathbf{x} \in B_1^{d+1} : h(\mathbf{x}) \neq \mathrm{R}(I_N)(\mathbf{x})\}) \lesssim_d C q^{\alpha} N^{-\alpha \frac{d+3}{2d}}. \tag{3.3.4}$$

*Moreover, $0 \leq \mathrm{R}(I_N(\mathbf{x})) \leq 1$ for all $\mathbf{x} \in \mathbb{R}^d$ and the architecture of $I_N$ is given by*

$$\mathcal{A} = \big(d+1,\ N+5,\ 2, 1\big).$$

*Thus, $I_N$ has at most $d + N + 9$ neurons and at most $(d+4)N + 5d + 25$ non-zero weights. The weights (and biases) of $I_N$ are bounded in magnitude by $\max\{1, \lceil \sqrt{2} q N^{\frac{d+3}{2d}} \rceil\}$.*

*Proof.* Since $h \in H_{\mathcal{B}}$, there exists $f \in \mathcal{B}$ such that $h(\mathbf{x}) = \mathbb{1}_{x_i \leq f(\mathbf{x}^{[i]})}$. We have by Proposition 3.2.10, that for all $N \in \mathbb{N}$, there exist an NN $\Phi_N^f$ having weights and biases $\mathbf{w}^{(k)} \in \mathbb{R}^d$, $b_k, v_k \in \mathbb{R}$ for $k = 1, 2, \ldots, N$, $\mathbf{c} \in \mathbb{R}^d$ and $c_0 \in \mathbb{R}$ with

$$|v_k|, |b_k|, |c_0|, \|\mathbf{c}\|, \|\mathbf{w}^{(k)}\| \leq 5 \|f\|_{RBV^2},$$

such that for

$$\mathrm{R}(\Phi_N^f)(\mathbf{x}) = \sum_{k=1}^{N} v_k \varrho((\mathbf{w}^{(k)})^{\top} \mathbf{x} - b_k) + \mathbf{c}^{\top} \mathbf{x} + c_0,$$

it holds that

$$\|f - \mathrm{R}(\Phi_N^f)\|_{L^{\infty}(B_1^{d-1})} \lesssim_d \|f\|_{RBV^2} N^{-\frac{d+3}{2d}}. \tag{3.3.5}$$

We define for $1 \geq \delta > 0$ the one layer NN $\Phi^{\delta}$ with

$$\mathrm{R}(\Phi^{\delta})(z) \coloneqq \frac{1}{\delta} \cdot (\varrho(z) - \varrho(z - \delta)) \text{ for } z \in \mathbb{R}$$

and for $\mathbf{x} \in \mathbb{R}^{d+1}$, we define

$$\begin{aligned}
&\mathrm{R}(\Phi_N^{\delta})(\mathbf{x}) \coloneqq H_{\delta}(\mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]}) - x_i) = H_{\delta}\left(\mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]}) - \varrho(x_i) + \varrho(-x_i)\right), \\
&\mathrm{R}(\Phi_N)(\mathbf{x}) \coloneqq \mathbb{1}_{\mathbb{R}^+}(\mathrm{R}(\Phi_N^f)(x^{[i]}) - x_i),
\end{aligned}$$

where $x_i$ is the $i$-th coordinate of $\mathbf{x}$. Note that $\Phi_N^{\delta}$ is an NN with two hidden layers, architecture

$$\mathcal{A} = \big(d+1,\ N+5,\ 2, 1\big),$$

and all weights bounded by $\max\{1/\delta, 5\|f\|_{RBV^2}\}$. We choose $\delta \coloneqq N^{-\frac{d+3}{2d}}$ and proceed to prove (3.3.4) for $I_N = \mathrm{R}(\Phi_N^f)^{\delta}$. Towards this estimate, we define

$$S \coloneqq \{\mathbf{x} \in B_1^{d+1} \colon h(\mathbf{x}) = 1\} \text{ and } T \coloneqq \{\mathbf{x} \in B_1^{d+1} \colon \mathrm{R}(\Phi_N^{\delta}) = 1\}.$$

We have that

$$\begin{aligned}\left\{\mathbf{x} \in B_1^{d+1} : h(\mathbf{x}) \neq \mathrm{R}(\Phi_N^\delta)(\mathbf{x})\right\} &\subset \left\{\mathbf{x} \in B_1^{d+1} : 0 < \mathrm{R}(\Phi_N^\delta)(\mathbf{x}) < 1\right\} \cup \left\{\mathbf{x} \in B_1^{d+1} : \mathrm{R}(\Phi_N^\delta)(\mathbf{x}) = 1, h(\mathbf{x}) = 0\right\} \\ &\quad \cup \left\{\mathbf{x} \in B_1^{d+1} : \mathrm{R}(\Phi_N^\delta)(\mathbf{x}) = 0, h(\mathbf{x}) = 1\right\} \\ &= \left\{\mathbf{x} \in B_1^{d+1} : 0 < \mathrm{R}(\Phi_N^\delta)(\mathbf{x}) < 1\right\} \cup \left\{\mathbf{x} \in B_1^{d+1} : \mathrm{R}(\Phi_N)(\mathbf{x}) = 1, h(\mathbf{x}) = 0\right\} \\ &\quad \cup \left\{\mathbf{x} \in B_1^{d+1} : \mathrm{R}(\Phi_N)(\mathbf{x}) = 0, h(\mathbf{x}) = 1\right\} \\ &=: \left\{\mathbf{x} \in B_1^{d+1} : 0 < \mathrm{R}(\Phi_N^\delta)(\mathbf{x}) < 1\right\} \cup S\Delta T.\end{aligned}$$

We observe that by (3.3.5), for $\gamma \sim_d RTV^2(f)$

$$\begin{aligned}&S\Delta T \\ &= \left\{\mathbf{x} \in B_1^{d+1} \colon f(\mathbf{x}^{[i]}) < x_i \leq \mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]})\right\} \cup \left\{\mathbf{x} \in B_1^d \colon \mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]}) < x_i \leq f(\mathbf{x}^{[i]})\right\} \\ &\subset \left\{\mathbf{x} \in B_1^{d+1} \colon 0 \leq \mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]}) - x_i < \gamma N^{-\frac{d+3}{2d}}\right\} \cup \left\{\mathbf{x} \in B_1^{d+1} \colon -\gamma N^{-\frac{d+3}{2d}} \leq \mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]}) - x_i < 0\right\} \\ &\subset \left\{\mathbf{x} \in B_1^{d+1} \colon |\mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]}) - x_i| \leq \gamma N^{-\frac{d+3}{2d}}\right\}.\end{aligned}$$

In addition,

$$\left\{\mathbf{x} \in B_1^{d+1} : 0 < \mathrm{R}(\Phi_N^\delta)(\mathbf{x}) < 1\right\} \subset \{\mathbf{x} \in B_1^{d+1} : |\mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]}) - x_i| \leq \delta\}.$$

We conclude by the $\alpha, C$ tube-compatibility of $\mu$ that

$$\begin{aligned}&\mu\left(\left\{\mathbf{x} \in B_1^{d+1} : h(\mathbf{x}) \neq \mathrm{R}(\Phi_N^\delta)(\mathbf{x})\right\}\right) \\ &\quad \leq \mu\left(\left\{\mathbf{x} \in B_1^{d+1} : 0 < \mathrm{R}(\Phi_N^\delta)(\mathbf{x}) < 1\right\}\right) + \mu(S\Delta T) \\ &\quad \leq \mu\left(\{\mathbf{x} \in B_1^{d+1} : |\mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]}) - x_i| \leq \delta\}\right) + \mu\left(\{\mathbf{x} \in B_1^{d+1} \colon |\mathrm{R}(\Phi_N^f)(\mathbf{x}^{[i]}) - x_i| \leq \gamma N^{-\frac{d+3}{2d}}\}\right) \\ &\quad \leq C \cdot \left(\delta^\alpha + \gamma^\alpha N^{-\alpha\frac{d+3}{2d}}\right).\end{aligned}$$

Due to the choice of $\delta = N^{-\frac{d+3}{2d}}$, this completes the proof. □

### 3.3.2 Estimation Bounds

Now, we focus our attention on computing upper bounds for the estimation problem. To this end, we need to introduce the following result.

**Theorem 3.3.5.** *[128, Theorem 5.6] Let $\Omega \subset [0,1]^d$ be a measurable set and $h = \mathbb{1}_\Omega$. Assume that:*

1. *There exists a positive sequence $(a_m)_{m\in\mathbb{N}}$ with $a_m = O(m^{-a_0})$ as $m \to \infty$ for some $a_0 > 0$ and a sequence of function classes $(\mathcal{F}_m^*)_{m\in\mathbb{N}} \subset L^\infty([0,1]^d)$ such that*

$$\forall m \in \mathbb{N} \quad \exists f_m^* \in \mathcal{F}_m^* : \quad \mathcal{E}_{\phi,\boldsymbol{\lambda},h}(f_m^*) \leq a_m.$$

*In addition, $0 \leq f_m(\mathbf{x}) \leq 1$ for all $\mathbf{x} \in [0,1]^d$, $f_m \in \mathcal{F}_m^*$ and $m \in \mathbb{N}$.*

2. *There exists* $C > 0$ *and* $(\delta_m^*)_{m\in\mathbb{N}} \subset (0,\infty)$ *such that the* $\delta-$*covering entropy* $V_{\mathcal{F}_m^*,\|\cdot\|_{L^\infty}}(\delta)$ *of the class* $\mathcal{F}_m^*$ *with respect to the norm* $\|\cdot\|_{L^\infty} = \|\cdot\|_{L^\infty([-1,1]^d)}$ *satisfies*

$$V_{\mathcal{F}_m^*,\|\cdot\|_{L^\infty}}(\delta_m^*) \leq C \cdot m \cdot \delta_m^* \quad \forall m \in \mathbb{N}.$$

3. *There exists* $\iota > 0$ *such that the sequence* $(\epsilon_m^*)_{m\in\mathbb{N}} \subset \mathbb{R}$ *with* $\epsilon_m^* := \max\{a_m, \delta_m^*\}$, $m \in \mathbb{N}$ *satisfies*

$$m^{1-\iota} \cdot \epsilon_m^* \gtrsim 1 \quad \forall m \in \mathbb{N}.$$

*Under these assumptions, for each training sample* $S = (\mathbf{x}_i, h(\mathbf{x}_i))_{i=1}^m$ *with* $\mathbf{x}_i$ *uniformly distributed for each* $i \in \{1,\dots,m\}$ *and* $h_S$ *satisfying* (3.3.1) *with* $\mathcal{H} = \mathcal{F}_m^*$, *it holds that*

$$\mathbb{E}_S[\boldsymbol{\lambda}(\{\mathbf{x} \in [0,1]^d : (2h(\mathbf{x}) - 1) \neq sign(2f_S(\mathbf{x}) - 1)\})] \lesssim \epsilon_m^*,$$

*where the implied constant is independent of* $h$ *and only depends on the implied constants from conditions 1-3. Here we use* $sign(0) = 1$.

We now present the main result of this subsection.

**Theorem 3.3.6.** *Let* $\kappa > 0$ *and* $q > 0$. *There is a constant* $\tau \geq 1$ *depending on* $d \in \mathbb{N}$ *only such that the following holds: Let* $h \in H_{\mathcal{B}}$, *where* $\mathcal{B} := \{f \in RBV^2(B_1^d) \colon f(0) = 0 \quad \|f\|_{RBV^2} \leq q\}$ *and let* $\boldsymbol{\lambda}$ *be a probability measure on* $B_1^{d+1}$. *For each* $m \in \mathbb{N}$, *let*

$$\begin{aligned}
\widetilde{N}(m) &:= \lceil \tau m^{2d/(3d+3)} \rceil, \\
N(m) &:= \lceil \widetilde{N}(m) + d + 9 \rceil, \\
W(m) &:= \lceil (d+4)\widetilde{N}(m) + 5d + 25 \rceil, \\
B(m) &:= \left\lceil \max\{1, \sqrt{2}\tau \widetilde{N}(m)^{\frac{d+3}{2d}}\} \right\rceil.
\end{aligned}$$

*For each* $m \in \mathbb{N}$, *let* $S$ *be a training sample of size* $m$; *that is,* $S = \big(X_i, h(X_i)\big)_{i=1}^m$ *with* $X_i \overset{iid}{\sim} \boldsymbol{\lambda}$ *for* $i \in \{1,\dots,m\}$. *Furthermore, let* $\Phi_{m,S} \in \mathcal{NN}_*\big(d+1, 2, N(m), W(m), B(m)\big)$ *be an empirical Hinge-loss minimizer; that is,*

$$\widehat{\mathcal{E}}_{\phi,S}\big(\Phi_{m,S}\big) = \min_{f \in \mathcal{NN}_*(d,N(m),W(m),B(m))} \widehat{\mathcal{E}}_{\phi,S}(f).$$

*Then, with* $H := \mathbb{1}_{[1/2,\infty)}$, *we have*

$$\mathbb{E}_S\Big[\mathbb{E}_{X\sim\boldsymbol{\lambda}}\big(\big|H(\mathrm{R}(\Phi_{m,S})(X)) - h(X)\big|^2\big)\Big] = \mathbb{E}_S\big[\mathbb{P}_{X\sim\boldsymbol{\lambda}}\big(H(\mathrm{R}(\Phi_{m,S})) \neq h(X)\big)\big] \lesssim m^{-\frac{d+3}{3d+3}+\kappa}, \tag{3.3.6}$$

*where the implied constant only depends on* $d, \kappa, \tau$.

*Proof.* The proof is analogous to [128, Theorem 5.7] with two main differences. Note that, [128, Theorem 5.7] is stated for functions defined on $[0,1]^{d+1}$. This is only due to the fact that it is based on [76, Theorem 5] which, while stated for functions on $[0,1]^{d+1}$ holds for

every compact subset of $\mathbb{R}^{d+1}$ as noted at the beginning of Section 2 of [76]. Therefore, [128, Theorem 5.7] also holds on $B_1^{d+1}$ equipped with a probability measure.

Notice that for all $\mathbf{x} \in \mathbb{R}^{d+1}$, $|H(\mathrm{R}(\Phi_{m,S})(\mathbf{x})) - h(\mathbf{x})| \in \{0,1\}$. Thus,

$$\mathbb{E}_{X\sim\boldsymbol{\lambda}}\left(|H(\mathrm{R}(\Phi_{m,S})(X)) - h(X)|\right) = \mathbb{P}_{X\sim\boldsymbol{\lambda}}\left(H(\mathrm{R}(\Phi_{m,S})(X)) \neq h(X)\right)$$

Since $sign(x) = 2\mathbb{1}_{[0,\infty)}(x) - 1$, it holds

$$sign(2x-1) \neq 2y-1 \iff \mathbb{1}_{[0,\infty)}(2x-1) \neq y \iff \mathbb{1}_{[1/2,\infty)}(x) \neq y \iff H(x) \neq y.$$

Substituting $x = \mathrm{R}(\Phi_{m,S})(X)$ and $y = h(x)$, we derive

$$\mathbb{P}_{X\sim\boldsymbol{\lambda}}\left(H(\mathrm{R}(\Phi_{m,S})(X)) \neq h(X)\right) = \mathbb{P}_{X\sim\boldsymbol{\lambda}}\left(sign(2\mathrm{R}(\Phi_{m,S})(X)) \neq 2h(X)-1\right)$$

which gives the first part of (3.3.2).

To prove the second part of (3.3.2), we need to verify the conditions of Theorem 3.3.5, following the approach in [128, Theorem 5.7]. In contrast to the proof of [128, Theorem 5.7], we choose the values

$$a_m = m^{-\frac{d+3}{3d+3}}, \quad \delta_m^* = m^{-\frac{d+3}{3d+3}+\kappa}, \text{ and } \iota = \frac{2d}{3d+3} + \kappa.$$

Additionally, to construct appropriate NN approximations, we use Theorem 3.3.4 instead of [128, Theorem 5.3] which yields the class

$$\mathcal{F}_m^* = \mathcal{NN}_*(d+1, N(m), W(m), B(m)). \tag{3.3.7}$$

Notice $0 \leq f_m(\mathbf{x}) \leq 1$ for all $f_m \in \mathcal{F}_m^*$. As there is a $f_m \in \mathcal{F}_m^*$ such that

$$\boldsymbol{\lambda}(\{\mathbf{x} \in [0,1]^{d+1} : h(\mathbf{x}) \neq f_m(\mathbf{x})\}) \leq Cq^\alpha N^{-\alpha\frac{d+3}{2d}} \lesssim a_m/2$$

and noting that $(2h(x)-1)(2f_m(\mathbf{x})-1) \in [-1,1]$ because $0 \leq f_m(\mathbf{x}) \leq 1$ and $h(\mathbf{x}) \in \{0,1\}$, it follows

$$\mathcal{E}_{\phi,\boldsymbol{\lambda},h}\big(f_m\big) \leq 2\boldsymbol{\lambda}\left(\{\mathbf{x} \in [0,1]^{d+1} : h(\mathbf{x}) \neq f_m(\mathbf{x})\}\right) \leq a_m.$$

This proves the first condition. Now, we compute

$$\begin{aligned} V_{F_m,\|\cdot\|_{L^\infty}}(\delta_m^*) &\leq W(m)\cdot(10 + \ln(1/\delta_m^*) + 5\ln(B(m)) + 5\ln(W(m)) \\ &\lesssim m^{2-d/(3d+3)}(10 + \ln(m) + \ln(m) + \ln(m)) \\ &\lesssim m^{2d/(3d+3)}(1+\ln(m)) \leq m\cdot m^{-2d/(3d+3)+\kappa} = m\cdot\delta_m^* \end{aligned}$$

which proves the second condition of Theorem 3.3.5. Finally, by the choice of $\iota$,

$$\epsilon_m^* = \delta_m^* = m^{-2d/(3d+3)+\kappa} = m^{\iota-1}$$

or equivalently

$$\epsilon_m^* m^{\iota-1} \gtrsim 1 \text{ for all } m \in \mathbb{N}.$$

Thus, by Theorem 3.3.4

$$\mathbb{P}_{X\sim\boldsymbol{\lambda}}(sign(2\mathrm{R}(\Phi_{m,S}(X))) - 1) \neq 2h(X)-1)\epsilon_m^* = \delta_m^* = m^{-\frac{d+3}{3d+3}+\kappa},$$

establishing the second part of (3.3.2).

□

## 3.4 Numerical Reconstruction of Horizon Functions with Different Boundaries

We first assume that our target classifier outputs a binary label for each point $x \in \mathbb{R}^{d+1}$ with $d = 1, 2$ and 3. Afterwards, we assume that our classifier is defined on higher-dimensional spaces. Here, the decision boundary is a sufficiently smooth function $f : B_1^d(0) \to \mathbb{R}$.

### 3.4.1 Setup for Low Dimensions

Our objective is to compare the performance of empirical risk minimization over NNs for classification problems with decision boundaries normalized with respect to various norms. To this end, we invoke the following setup:

- *Base function class:* We first describe a base set of functions that will later be normalized to form the boundary functions to be learned. For the case $d = 1$, we consider the set of functions

$$B \coloneqq \{B_1^1(0) \ni x \mapsto \sin(2\pi k x) : k = 1 + s/24,\ s = 0, 1, \ldots, 24,\ x \in [-1, 1]\},$$

  where for each $f \in B$ we define $|f| \coloneqq |k|$. Now, for $d = 2$, the functions to be learned belong to the set

$$B \coloneqq \{B_1^2(0) \ni (x, y) \mapsto \sin(2\pi k x)\sin(2\pi l y) : k, l = 1 + s/5, s = 0, 1, \ldots, 5\ x \in [-1, 1]\},$$

  and for $f \in B$ we denote $|f| = |k| + |l|$. Finally, for $d = 3$, we learn the functions

$$B \coloneqq \{B_1^3(0) \ni (x, y, z) \mapsto \sin(2\pi k x)\sin(2\pi l y)\sin(2\pi j z) : k, l, j = 1, 4/3, 7/3, 2\ x \in [-1, 1]\}.$$

  and again, for $f \in B$ we employ the notation $|f| = |k| + |l| + |j|$.

- *Normalization:* We consider the following norms: first, the uniform norm

$$\|f\|_\infty = \sup_{x \in B_1^{d-1}} |f(x)|,$$

  second, the $L^1$-norm

$$\|f\|_{L^1} = \int_{B_1^{d-1}} |f(x)| dx,$$

  third, the $C^1$-norm

$$\|f\|_{C^1} = \sup_{|k| \leq 1} \|f^k\|_\infty,$$

  where $k \in \mathbb{N}_0^d$, fourth, the Barron norm,

$$\|f\|_{\text{Barron}} = \int_{-\infty}^{\infty} |\widehat{f}(w)||w| dw$$

as defined in [14, Equation 3], where $\widehat{f}$ is the Fourier transform of the function $f$, and the Barron norm. As the function $f$ is defined on a bounded domain, its Fourier transform is not well defined. To solve this issue and leverage the properties of the functions in the sets $B$ to be learned, we replace the Fourier transform with the appropriate term

$$\|f\|_{\text{Barron}} \approx 2\pi|f|.$$

Finally, we use the $RBV^2$ norm. For the computation of the $RBV^2$ norm, we use different approaches. For the case $d = 1$ we use [123, Remark 4]. It is stated there that the $RBV^2([-1, 1])$ space is equivalent to the set of functions of second-order bounded variation

$$BV^2([-1, 1]) = \{f : [-1, 1] \to \mathbb{R} : TV^2(f) < \infty\},$$

where

$$TV^2(f) = \int_{-1}^{1} |D^2 f(x)| dx$$

is the second-order total variation of $f : [-1, 1] \to \mathbb{R}$.

In [123], it is proved that $RTV^2(f) = TV^2(f)$ and the second total-variation seminorm of $f$ is well-defined for all smooth functions $f$. Hence, for each $f$ with $f(0) = 0$, we can compute its $RBV^2$ norm by

$$\|f\|_{RBV^2} = TV^2(f) + |f(1)|.$$

For the case $d = 2$ and $d = 3$, we use a different approach to compute the integral term in Definition 3.2.5. To this end, we generate a set of 20 vectors $w_i \in S^{d-1}$ and 20 equidistant parameters $t_i \in [-1, 1]$. Next, we compute the Radon transform at these points by using randomly generated points drawn according to the uniform distribution in the hyperplane $\{z : w^T z = t\}$. Then, by using a method of finite differences, we compute the corresponding derivatives, and we continue to compute the total variation of the function. To compute the norm, we have to evaluate the function $f$ at the origin and at the elements of the canonical basis as in Definition 3.2.5.

- *Normalized function classes:* To produce the functions to be learned, we normalize the elements of the corresponding base set $B$ with respect to the uniform, $L^1$, $C^1$, Barron, and $RBV^2$ norms, which yields five sets of functions for each dimension $d$. Next, we compare the perform of empirical risk minimization over appropriate sets of NNs to learn the classifiers associated to these normalized function classes.

- *Data generation:* The sample set consists of $m$ points $(x_i, y_i)_{i=1}^m$. The points $x_i$ are randomly generated points drawn with respect to uniform distribution on the set $B_{d+1}(0)$. To determine the value of each $y_i$, we evaluate the function $y = \mathbf{1}_{x_{d+1} \leq f(x^{[d+1]})}$ at each $x_i$. We split this sample set into two subsets: the training and the test sets. The first one is randomly chosen with 80% of all samples and is used to find an NN that minimizes the empirical error.

- *Neural network set-up:* We use a three-layer NN with ReLU activation function, and the number of neurons is determined according to Theorem 3.3.6 with $\tau = 1$. We use the Hinge function as our loss function and the Adam optimizer.

- *Performance metric:* We compare the performance of the selected NN on the test sets. This error is measured in the mean squared sense.

We record the generalization error for each function $f \in B$ and number of samples $m$ and present the results in Figure 3.4.1. In the $x$-axis, we register the number of samples, and in the $y$-axis, the average test error. In each figure, we plot the error for the five different norms introduced above.

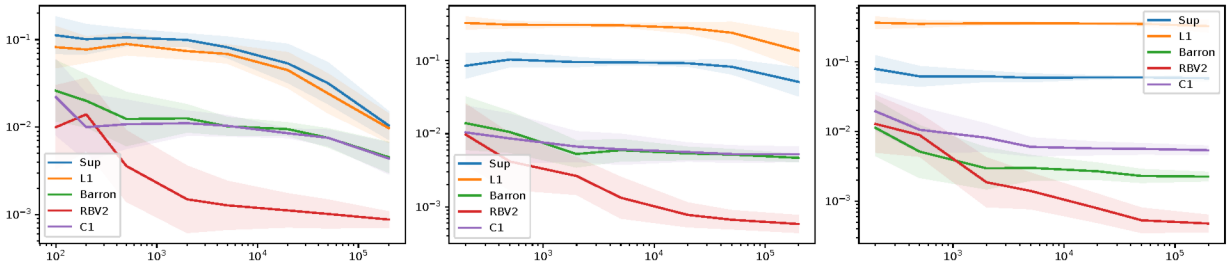


Figure 3.5: Plot of the mean relative test error for the sets of case $d+1 = 2$ (left), $d+1 = 3$ (middle) and $d+1 = 4$ (right). The number of samples varies along the horizontal axis and the mean relative test error is shown on the vertical axis.

**Evaluation and interpretations of the results** There are a few points we would like to highlight that can be observed from the figures.

1. In Figure 3.4.1, we can observe that when the classification boundary function is normalized with respect to $L^1$ or $L^\infty$ normalized, then the test error is usually the highest. Interestingly, in the case of $d+1 = 2$, the test error behaves more favorably compared to higher dimensions. For dimensions larger than two, increasing the number of samples results in almost no improvement in classification performance. This finding is consistent with the lower bounds on learnability discussed by [128], which show that classification functions with boundaries in normed spaces characterized by large packing numbers are difficult to learn effectively.

2. For the case $d + 1 = 3$ and $d + 1 = 4$ in Figure 3.4.1, we observe that learning classification functions normalized using either the $C^1$ norm or the Barron norm exhibit similar performance. However, when the dimension increases to $d + 1 = 4$, the Barron norm yields better test error results, which suggests that its advantages become more evident in higher dimensions.

3. In all cases, the classification functions with $RBV^2$ normalized boundary shows the best performance among all the tested norms. Moreover, the test errors show essentially the

same decay for all dimensions tested. These examples support the idea that functions with $RBV^2$ boundary are easier to learn than functions with lower regularity and are in line with the theoretical results of Theorem 3.3.6.

4. We can observe that the performance of $RBV^2$ boundaries surpasses that of Barron spaces. This is consistent with our theoretical results from Theorem 3.3.6 and aligns with the findings in [128], further supporting the fact that higher regularity in function boundaries leads to a better learning outcome.

### 3.4.2 Setup for High Dimensions

We now present the setup for our numerical experiments, similar to the one in Section 3.4.1, but adapted for high-dimensional settings. However, since it is computationally infeasible to compute the Radon and Fourier transforms of a high-dimensional function numerically, we present the results only for sets of functions where we have a closed form for the Radon and Barron norms. Because of this, we study two sets of function classes, one where the Barron norm is immediately computable and one where this is the case for the $RBV^2$-norm.

- *Base function class:* We describe the boundary functions used for the classifiers that we aim to learn. We begin by generating a sequence of weights $w_k \in \mathbb{R}^d$, $v_k \in \mathbb{R}$ and biases $b_k \in \mathbb{R}$ where each component is drawn from a normal distribution with zero mean and unit variance. This sequence is then used to assemble the function

$$f_k(x) = \sum_{j=1}^{K} v_k \sigma(w_k^T x - b_k) \tag{3.4.1}$$

  where $\sigma(x)$ is an activation function. As mentioned before, we conduct two sets of experiments, we use two different activation functions. In the experiments comparing the effect of the $RBV^2-$norm, we choose $\sigma(x) = \varrho(x)$, the ReLU, because the $RBV^2$-norm of (3.4.1) can then be immediately computed using [125, Equation (32)]. Indeed, the seminorm can be written as:

$$RBV^2(f_k) = \sum_{i=1}^{k} |v_i| \|w_i\|_2$$

  where $|v_k|$ denotes the absolute value of the $v_i$ and $\|w\|_2$ is the Euclidean norm of the inner weights for the $k$-th neuron. In the case of comparing the performance of the Barron norm, we choose the function $\varrho(x) = \sin(x)$, since this allows an analytic computation of the Fourier transform and the Barron norm.

- *Normalization:* In the numerical experiments presented below, we perform a normalization of the parametrized function with respect to the uniform, $L^1$, $C^1$, Barron and

$RBV^2$ norms and the $TV$ seminorm. For the total variation norm of the function $f : B_1^d \to \mathbb{R}$, we use the expression

$$\|f\|_{TV} := \int_{B_1^d} |\nabla f(x)| dx. \tag{3.4.2}$$

In order to compute the integrals for each norm, we use the trapezoidal rule with 2000 randomly generated samples. We estimate the $L^\infty, L^1$, and $C^1$ using similar Monte Carlo estimates.

- *Data generation:* As in the low-dimensional setting, we use a sample set consisting of $m$ points $(x_i, y_i)_{i=1}^m$. The points $x_i$ are randomly generated points drawn with respect to uniform distribution on the set $B_1^{d+1}(0)$. The maximum number of training samples is 100000, and the maximum number of test samples is 10000.

  We compare the results in Figure 3.4.2 for dimension $d + 1 = 20$ and in Figure 3.4.2 for $d + 1 = 40$.

- *Neural network setup:* We use a three-layer NN with a ReLU activation function, and the number of neurons is determined according to Theorem 3.3.6 with $\tau = 1$. The Hinge loss function is used, and the Adam optimizer is applied.

- *Performance metric:* Each experiment is run 25 times, and we compute the average Mean Squared Error (MSE) across all runs.

In each run, for dimensions $d + 1 = 20$ and $d + 1 = 40$, we generate a function $f_k$. Then, we record the average MSE across all runs to illustrate the generalization error as a function of the number of samples $m$ in Figures 3.4.2 and 3.4.2, respectively. The $x$-axis represents the number of samples and the $y$-axis shows the average test error. In each figure, we plot the error for the five different norms introduced above. In addition, in Figure 3.4.2, we record the generalization error as a function of the dimension. The $x$-axis represents the dimension, and the $y$-axis shows the average test error.

**Evaluation and interpretations of the results** There are a couple of observations we would like to emphasize from the figures

1. As we predicted by our theory and in accordance with previous works, for a fixed dimension $d$, when the classifier to be learned has a boundary normalized by the $RBV^2$ or the Barron norm, we observe a better test error compared to other norms. This effect becomes more pronounced as the number of samples increases.

2. In all cases, the reconstruction under the $L^1$ norm produces the worst test errors. Once again, this facts aligns with the lower bounds on learnability discussed by [128].

3. In Figure 3.4.2, we observe the influence of the dimension in the value of the test error. For a fixed number of samples, when the dimension increases, the test error increases. This is in line with previous results on classifiers with boundaries parametrized with different norms. Note that, even for the Barron and $RBV^2$ norms, the error for a fixed number of samples does increase with the dimension. This is not in violation of our results, because they describe the asymptotic rates. There are constants involved that do depend on the dimension, and their effect is visible here.

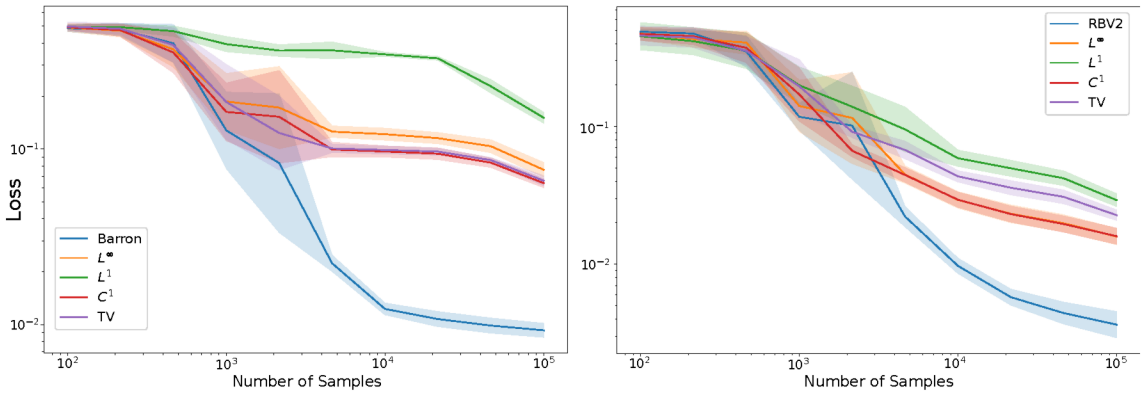


Figure 3.6: Plot of the mean relative test error for the sets of case $d + 1 = 20$, normalizing the boundary with respect to the Barron norm (left) and $RBV^2$ norm (right). The number of samples varies along the horizontal axis and the mean relative test error is shown on the vertical axis.

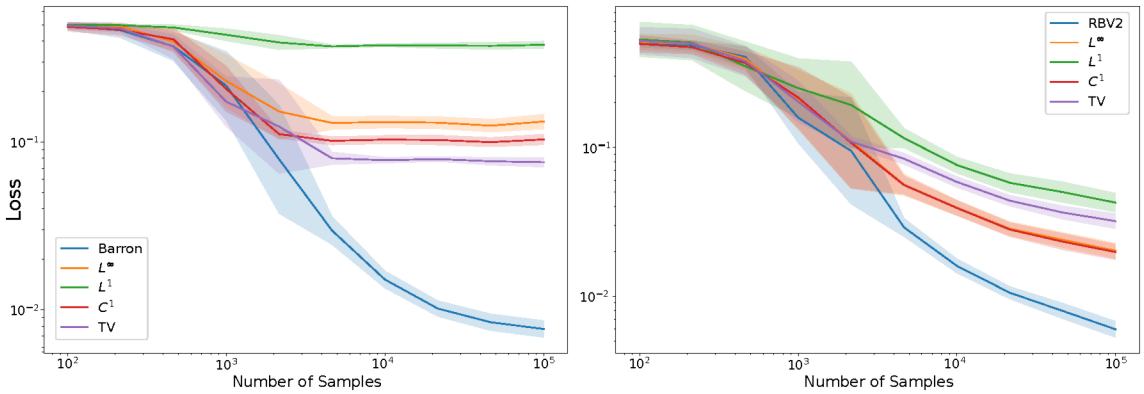


Figure 3.7: Plot of the mean relative test error for the sets of case $d + 1 = 40$, normalizing the boundary with respect to the Barron norm (left) and $RBV^2$ norm (right). The number of samples varies along the horizontal axis and the mean relative test error is shown on the vertical axis.

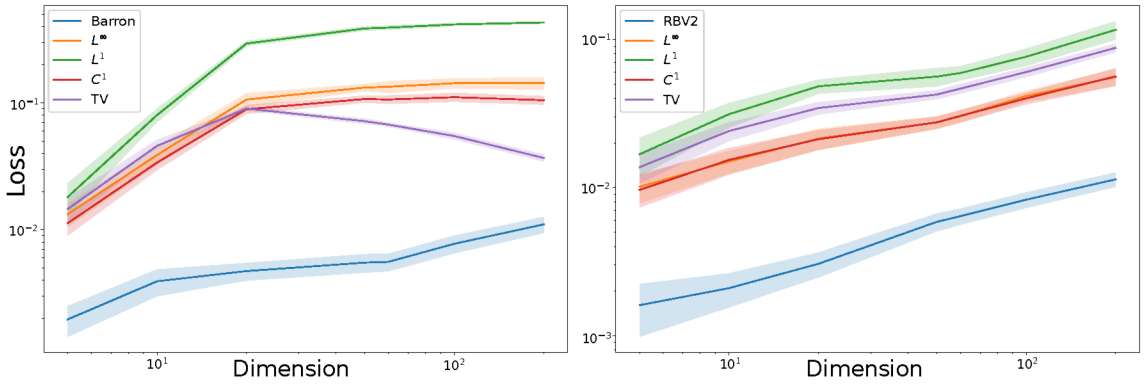


Figure 3.8: Plot of the mean relative test error for the sets of case $m = 50000$, normalizing the boundary with respect to the Barron norm (left) and $RBV^2$ norm (right). The dimension $d$ varies along the horizontal axis and the mean relative test error is shown on the vertical axis.

# Chapter 4

# A Practical Existence Theorem for the Solution of Inverse Problems

We dedicate this chapter to the solution of ill-posed infinite-dimensional inverse problems from samples using neural-network-based methods. Following the setting in Chapter 3, our main result corresponds to a PET which guarantees the existence of an NN, the realization of which approximates the solution of an inverse problem that satisfies certain conditions. Although we have already discussed what an inverse problem is, we present here a more rigorous concept. All of our discussed inverse problems are assumed to be modeled by an operator $F$ between two Banach spaces $\mathcal{X}$ and $\mathcal{Y}$ as discussed in Definition 4.0.1.

**Definition 4.0.1.** *[79] Let $\mathcal{X}, \mathcal{Y}$ be normed spaces and $F : \mathcal{X} \to \mathcal{Y}$ an operator. Let us consider the equation*

$$F(x) = y \tag{4.0.1}$$

*for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The problem of reconstruction $x \in \mathcal{X}$ given $y \in \mathcal{Y}$ and $F$ is called the inverse problem associated to the equation 4.0.1. The problem of prediction $y \in \mathcal{Y}$ given $x \in \mathcal{X}$ and $F$ is called the direct problem associated with the equation 4.0.1.*

Inverse problems exhibit undesirable properties that make the determination and analysis of their solution difficult. In particular, there may not be a solution, or there may be more than one solution or such a solution may be unstable. These properties are formalized below using the well-known notion of *ill-posed* inverse problem as presented in Definition 4.0.2.

**Definition 4.0.2.** *[79] Let $\mathcal{X}, \mathcal{Y}$ be two normed spaces and $F : \mathcal{X} \to \mathcal{Y}$ an operator. We say that an inverse problem associated to equation* 4.0.1 *is well-posed if the following conditions are fulfilled*

1. ***Existence:*** *There is at least one solution $x \in \mathcal{X}$ of the equation $F(x) = y$ for $y \in \mathcal{Y}$.*
2. ***Uniqueness:*** *There is at most one solution of the equation $F(x) = y$ for $y \in \mathcal{Y}$.*
3. ***Stability:*** *If $y_n = F(x_n)$ with $y_n = F(x_n)$ and $y_n \to y = F(x)$, then $x_n \to x$ in $\mathcal{X}$.*

*Otherwise, we say that the problem is ill-posed.*

A well-known inverse problem is that of the *differentiation* of a function between normed spaces. For simplicity, we consider the space of continuous functions on the real line. This problem is ill-posed since not every continuous function is differentiable. But even if we restrict ourselves to the set of differentiable functions, the stability property cannot be ensured. We illustrate this issue in more detail for the following.

**Example 4.0.3.** *Consider the operator $F : C([0,1]) \to C([0,1])$ defined as*

$$g(t) = F(f)(t) := \int_0^t f(s)ds, \tag{4.0.2}$$

*for all $f \in C([0,1])$ and $g \in C([0,1])$. We can guarantee the existence and uniqueness of the solution for the associated inverse problem if $g \in C^1([0,1])$. However, even if we consider only the space $C^1([0,1])$, the associated inverse problem is ill-posed. By the fundamental theorem of calculus, we know that the inverse problem associated with Equation 4.0.2 corresponds to the problem of computing the derivative of the function $g = g(t)$. Now we analyze the stability of the inverse problem. Take, for example, $g(t) = e^t$. Clearly, $f(t) = g'(t) = e^t$. But if instead, we consider $g_n(t) = e^t + \frac{1}{n}\sin n^2 t$,*

$$\|g_n - g\|_{C([0,1])} \to 0,$$

*and*

$$\|(g_n)' - f\|_{C([0,1])} \to \infty.$$

*In Figure 4.1, we represent the functions $g$ and $g_n$ for $n = 20$ and in Figure 4.2 we illustrate their derivatives. The functions $g$ and $g_{100}$ are close, but their derivatives differ. In particular, the derivative of $g_n$ oscillates several times in the interval $[0,1]$. This example shows the instability of the derivative.*

One can conveniently restrict the domain and the codomain of the operator $F$ to ensure the existence and uniqueness of solutions for inverse problems. However, the question of stability involves the topologies of the spaces $X$ and $Y$ and changing the underlying topologies may pose a big challenge [79]. To remedy the ill-posedness of an inverse problems, one typically applies the so-called *regularization techniques*.

**Definition 4.0.4.** *[79] Let $\alpha, \delta > 0$ and let $F : \mathcal{X} \to \mathcal{Y}$ be an injective operator between normed spaces. We say that $R_\alpha : Y \to X$ for $\alpha > 0$ is a regularization operator if*

$$\lim_{\alpha \to 0^+} R_\alpha F x = x.$$

*for $x \in \mathcal{X}$. Moreover, we say that the regularization operator defines a regularization method for the inverse problem associated to (4.0.1), if there is a choice $\alpha = \alpha(\delta)$ such that*

$$\lim_{\alpha \to 0^+} R_\alpha F x = x$$

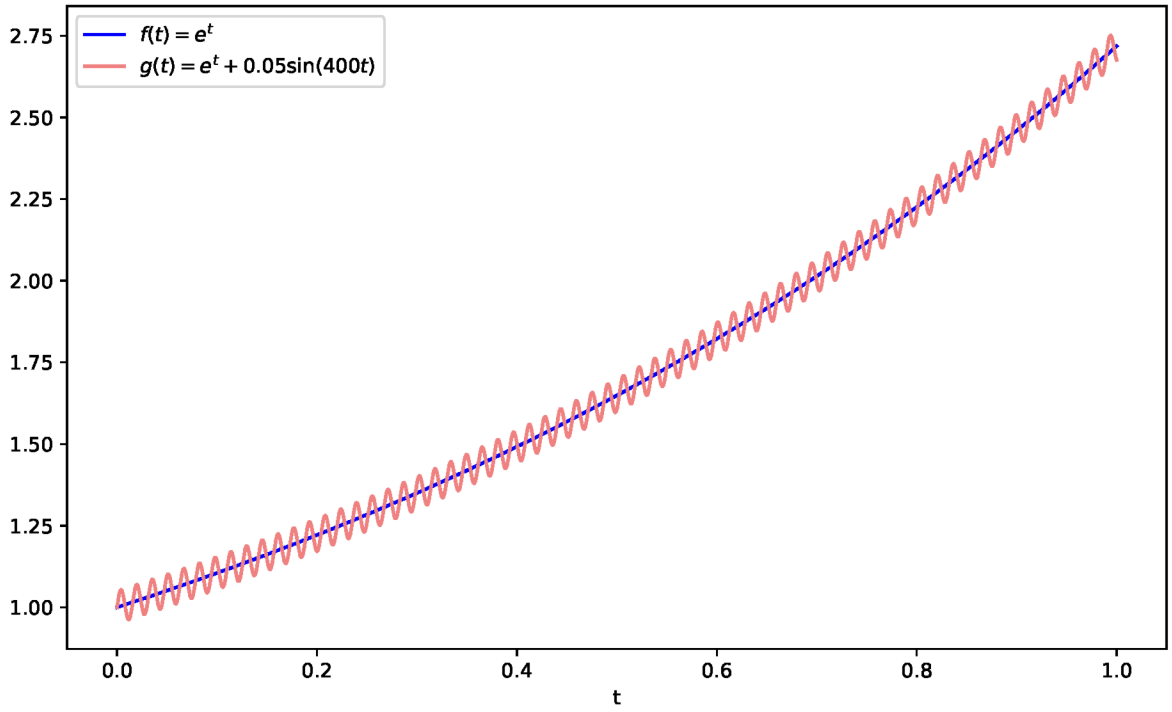


Figure 4.1: Plot of the functions $g_{100}$ and $g$. The difference between the two functions is small.

*and*

$$\lim_{\delta \to 0^+} \|R_\alpha y^\delta - x\|_{\mathcal{X}} = 0.$$

There are at least four main categories in which regularization techniques can be classified. These categories were discussed in [10] and will be briefly reviewed below. The simplest of these approaches are the *analytic inversion methods* which consist of determining a closed-form expression for $F^{-1}$ and tryingting to stabilize it (see [112, 113]). Those methods provide well-posed solutions to linear inverse problems where the forward operator is compact and there is a spectral decomposition that allows a closed form of $F^{-1}$. The second category is the family of *iterative methods* based on gradient-descent algorithms to minimize a functional $x \mapsto \|F(x) - y\|$ in a stable way (see [50, 79]). The third category is that of the *discretization methods* such as the *Galerkin methods* where the solution is discretized to a finite-dimensional space. It is based on the idea that appropriate discretization stabilizes the inversion, e.g., [111, 134]. Finally, the last category is composed of *variational methods* that reconstruct the solution by minimizing a functional of the form

$$J_\alpha(x) = \|F(x) - y\| + S_\alpha(x),$$

where $S_\alpha : \mathcal{X} \to \mathbb{R}$ is chosen to enforce a priori known properties of the solution and $x \in \mathcal{X}$ (see [73]).

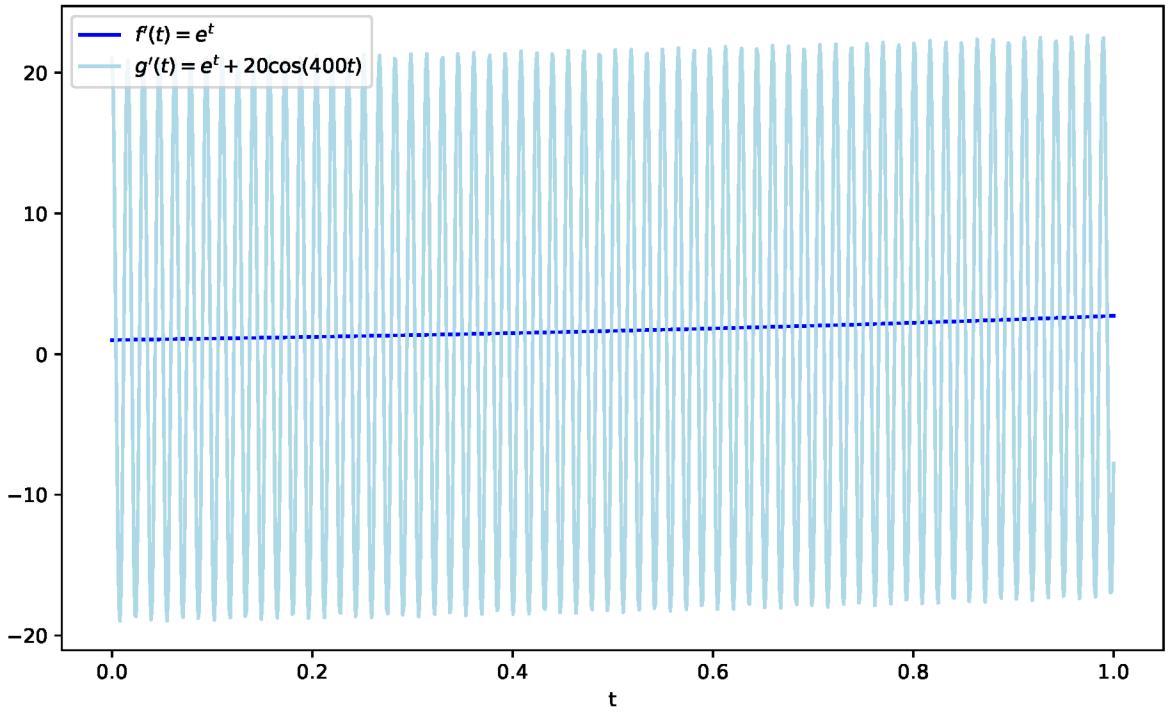


Figure 4.2: Plot of the derivative functions $g'_{100}$ and $g'$. The difference between the two functions at most points is is very large. Indeed, the function $g'_{100}$ oscillates several times in the interval $[0, 1]$ with large amplitude.

## 4.1 Transformation of Infinite-dimensional Inverse Problems into Finite-dimensional Problems

In this section, we present a strategy to transform infinite-dimensional discontinuous inverse problems into Lipschitz regular finite-dimensional problems. This approach forms the basis of our results and will be briefly discussed below. The ideas for this methods where presented in [4]. Assuming again an inverse problem in the form of $F : \mathcal{X} \to \mathcal{Y}$, we choose a finite-dimensional space $W_0 \subset \mathcal{X}$ and restrict the forward operator to this space. However, restricting the domain only, does not guarantee that the image of the operator lies in a finite-dimensional linear space. To overcome this problem, a family of finite-rank bounded operators $Q_N : \mathcal{Y} \to \mathcal{Y}$, $N \in \mathbb{N}$, approximating the identity $I_{\mathcal{Y}}$ is introduced in [4] so that the function between finite-dimensional spaces $Q_N \circ F|_{W_0} : W_0 \to \mathrm{Im}(Q_N)$ can be interpreted as a continuous approximation of $F$. This approach fits into what can be expected in real-life problems. In practice, $Q_N$ corresponds to a sampling procedure. The following theorem, proved in [4], allows the Lipschitz continuous discretization of infinite-dimensional unstable inverse problems.

**Theorem 4.1.1** ([4, Corollary 1]). *Let $\mathcal{X}$, $\mathcal{Y}$ be Banach spaces, $A \subset \mathcal{X}$ be an open subset, $W_0 \subset \mathcal{X}$ be a finite-dimensional subspace and $K \subset W_0 \cap A$ be a compact and convex subset.*

*If $F \in C^1(A, \mathcal{Y})$ is such that $F|_{W_0 \cap A}$ is injective and $F'(x)|_{W_0}$ is injective for all $x \in W_0 \cap A$, then there is a constant $C > 0$ and $D \in \mathbb{N}$ such that*

$$\|x_1 - x_2\|_{\mathcal{X}} \leq C \|Q_D(F(x_1)) - Q_D(F(x_2))\|_{\mathcal{Y}}, \tag{4.1.1}$$

*for all $x_1, x_2 \in K$.*

Theorem 4.1.1 states that the inverse of the function $Q_D \circ F|_K : K \to \mathrm{Im}(Q_D)$ is Lipschitz continuous. This observation will be significant for the solution of inverse problems by neural-network-based methods that will be proposed in this work. The constant $C > 0$ can be estimated in terms of a lower bound of the Fréchet derivative and the moduli of continuity of $(F|_K)^{-1}$ and $F'$. In [11] and [137] some ad-hoc techniques have been tailored to estimate such a constant $C$ for specific problems.

All these inverse problems are modeled by an operator $F$ between two Banach spaces $\mathcal{X}$ and $\mathcal{Y}$. We solve the inverse problem if the solution belongs to a manifold $\Gamma$ which is a subset of a compact and convex subset $K$ of a finite-dimensional subspace $W_0 \subset \mathcal{X}$. The first step is to choose a sequence of finite-rank operators $\widehat{Q}_N : \mathcal{Y} \to Y_N$ where $Y_N \subset \mathcal{Y}$ is an $N$-dimensional subspace for all $N \in \mathbb{N}$ such that $(\widehat{Q}_N)_{N \in \mathbb{N}}$ is a sequence of operators converging strongly to the identity operator on $\mathcal{Y}$, i.e., for all $x \in \mathcal{X}$ it holds that $\|\widehat{Q}_N x - x\|_{\mathcal{Y}} \to 0$ as $N \to \infty$. We identify the finite-dimensional spaces $W_0$ and $Y_N$ with $\mathbb{R}^d$ and $\mathbb{R}^N$ via affine isomorphisms $P : \mathbb{R}^d \to W_0$ and $T_N \colon Y_N \to \mathbb{R}^N$. Then, the inverse problem of finding $x \in \Gamma$ given $y \in \mathrm{ran}\, \widehat{Q}_N \circ F$ modeled by the equation

$$\widehat{Q}_N \circ F(x) = y$$

is equivalent to the inverse problem of finding $x' \in P^{-1}(\Gamma)$ given $y' \in \mathbb{R}^N$ such that

$$T_N \circ \widehat{Q}_N \circ F \circ P(x') = y'$$

where $x' = P^{-1}(x)$ and $y' = T_N(y)$.

For each inverse problem that will be introduced below, we show that there is a $D \in \mathbb{N}$ such that the following discretized version of $F$,

$$\begin{aligned} \widetilde{F} &\colon P^{-1}(\Gamma) \subset \mathbb{R}^d \to \mathbb{R}^D, \\ \widetilde{F} &\coloneqq T_D \circ \widehat{Q}_D \circ F|_K \circ P, \end{aligned}$$

where $F|_K$ is the restriction of $F$ to $K$, satisfies

(i) $\mathrm{ran}\, \widetilde{F}$ is an $(M, \delta)$-covered submanifold $\mathcal{M}$. The notion of $(M, \delta)-$covered submanifold will be introduced in Definition 4.2.1,

(ii) $\widetilde{F}$ is invertible on $\mathcal{M}$,

(iii) $\widetilde{F}^{-1} \colon \mathcal{M} \to P^{-1}(\Gamma)$ is Lipschitz continuous.

In the sequel, for the sake of clarity, we will always use $\Gamma$ to denote a manifold contained in the domain of a forward operator and $\mathcal{M}$ to denote a submanifold of the image. To show the conditions (i), (ii), and (iii) for the inverse problems introduced below, we derive closed expressions for the forward operators on certain domains $H \subset \mathcal{X}$. Then, we demonstrate that the respective forward operator $F$ is injective on $H$ and Fréchet differentiable at each $a \in H$. Next, we use Theorem 4.1.1 to show that for every compact and convex set $K \subset W_0 \cap H$ where $W_0$ is a $d$-dimensional subspace of $\mathcal{X}$, there is a $D \in \mathbb{N}$ such that $(\widehat{Q}_D \circ F|_K)^{-1}$ is Lipschitz continuous.

Thereafter, we identify the finite-dimensional spaces $W_0$ and $Y_D$ with the finite-dimensional spaces $\mathbb{R}^d$ and $\mathbb{R}^D$ through isomorphisms $P : \mathbb{R}^d \to W_0$ and $T_D : Y_D \to \mathbb{R}^D$. For the inverse problems presented in this paper, $P$ is usually an affine isomorphism and $T_D \circ \widehat{Q}_D$ is a point sampling operator. Hence, due to the composition of invertible functions we see that $\widetilde{F}$ defined as above is invertible and bi-Lipschitz, which implies that conditions (ii) and (iii) hold. We will see in all cases below, that the discretized operator $\widetilde{F}$ is $C^\infty$ and bi-Lipschitz. Hence it follows that $\widetilde{F}$ is a diffeomorphism which implies that $\mathcal{M}$ is an $(M, \delta)$-covered submanifold.

## 4.2 Practical Existence Theorem for Inverse Problems

Before we study solutions for inverse problem by neural networks, we fix some notation: for a given $\mathbf{y} \in \mathcal{M}$ and $\delta' > 0$, we denote by $B_{\delta'}(\mathbf{y})$ the ball with radius $\delta'$ and center at $\mathbf{y}$, and by $\mathrm{P}_{\mathbf{y},\mathcal{M}}$ the orthogonal projection onto the tangent space $T_y\mathcal{M}$ of $\mathcal{M}$ at $\mathbf{y}$ defined by $\mathrm{P}_{\mathbf{y},\mathcal{M}} : B_{\delta'}(\mathbf{y}) \cap \mathcal{M} \to V_y \subset T_y\mathcal{M}$. Before we prove the main result of this paper, we introduce the property of $(M, \delta)$-coveredness below.

**Definition 4.2.1.** *Let $n \in \mathbb{N}$ and $\mathcal{M} \subset \mathbb{R}^n$ be a relatively compact manifold. We say that $\mathcal{M}$ is* $(M, \delta)$-covered, *where $M \in \mathbb{N}$ and $\delta > 0$ if there exist a set of points $(\mathbf{y}_i)_{i=1}^M \subset \mathcal{M}$ such that*

$$\bigcup_{i=1}^{M} B_{\frac{\delta}{4}}(\mathbf{y}_i) \supset \mathcal{M},$$

*where the inverse of $\mathrm{P}_{\mathbf{y}_i,\mathcal{M}}$ is Lipschitz continuous with Lipschitz constant bounded by 2 for all $i \in \{1, \dots, M\}$.*

Definition 4.2.1 is crucial for the proof of Theorem 4.2.2. In fact, to reconstruct a stable approximation for the inverse operator of every inverse problem presented in Section 4.3, we need to prove that all the corresponding manifolds $\mathcal{M}$ are $(M, \delta)$-covered for some values of $M$ and $\delta$. To this end, notice that if $F$ is a bi-Lipschitz diffeomorphism, then for $(a, b)^d \subset \mathbb{R}^d$ with $a, b \in \mathbb{R}$, it follows that

$$\widetilde{F}((a, b)^d) \subset \overline{\widetilde{F}((a, b)^d)} \subset \widetilde{F}((a - \varepsilon, b + \varepsilon)^d),$$

for $\varepsilon > 0$. Hence, $\mathcal{M} = \widetilde{F}((a, b)^d)$ is a relatively compact subset of a smooth submanifold. The fact that $\mathcal{M}$ and $\widetilde{F}((a - \varepsilon, b + \varepsilon)^d)$ are submanifolds follows from [91, Proposition 5.2]. The $(M, \delta)$-coveredness then follows from a standard compactness argument.

**Theorem 4.2.2.** *Let $q, m, d, D \in \mathbb{N}$, $\mathcal{D}$ be a distribution on $K \subset [0,1]^q$. Let $F\colon K \to \mathcal{M} \subset [0,1]^D$, be such that $\mathcal{M}$ is a $(M,\delta)$-manifold and $F$ satisfies the assumptions of Theorem 4.1.1. Let $\eta \in \mathbb{R}^D$ be a Gaussian random variable where all components are independent, zero-mean with variance $\sigma^2$. Let $S = (\mathbf{s}_i^{(1)}, \mathbf{s}_i^{(2)})_{i=1}^m = (F(\mathbf{x}_i) + \eta_i, \mathbf{x}_i)_{i=1}^m$ be a sample where $\mathbf{x}_i \sim \mathcal{D}$ and $\eta_i \sim \eta$ are independent random variables for $i = 1, \dots, m$. Then, there is an $r_0$ such that $\mathcal{E}(|\eta_i|^2) \leq r_0^2$ and a class of NNs, denoted $\mathcal{NN}_\epsilon$ such that with probability $1-2p$ over the choice of $S$ such that the solution of*

$$\Phi_S \in \operatorname*{arg\,min}_{\Phi \in \mathcal{NN}_*(D,q,L,W,B)} \left\{ \frac{1}{m} \sum_{i=1}^m \left| \mathrm{R}(\Phi)\left(\mathbf{s}_i^{(1)}\right) - \mathbf{s}_i^{(2)} \right|^2 \right\}. \tag{4.2.1}$$

*satisfies*

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\eta} \left| \mathrm{R}\left(\Phi_S\right)\left(F(\mathbf{x}) + \eta\right) - \mathbf{x} \right|^2 \leq \overline{C}\kappa + c' \sqrt{\frac{q\kappa^{-d/2}\ln(1/\kappa)}{m}} + 4De^{-\frac{r_0^2}{2\sigma^2}} + \sqrt{\frac{16\ln(2/p)}{m}}, \tag{4.2.2}$$

*where $c' > 0$ depends on $\mathcal{M}$ and $\overline{C} > 0$ depends only on $C$ (it is in particular independent from $D$, $d$, $q$, and $M$). Moreover, the NN is robust-to-noise, i.e,*

$$\mathbb{E}\left( \sup_{\mathbf{z}\in\mathcal{M}} \left| \mathrm{R}\left(\Phi_S\right)\left(\mathbf{x}+\eta\right) - \mathrm{R}\left(\Phi_S\right)\left(\mathbf{x}\right) \right|^2 \right) \leq \frac{\hat{C}}{D} \mathbb{E}(|\eta|^2) \cdot \left( \sqrt{2d\log(M)} + 2\log(M) + d \right) + 8\epsilon^2 + 2De^{-\frac{r_0^2}{2\sigma^2}}, \tag{4.2.3}$$

for $\mathbf{x} \in \mathcal{M}$.

**Remark 4.2.3.** *The realization of an approximating NN acting on $\mathbb{R}^D$ is robust to small perturbations. Concretely, in (4.2.4), the variance of the noise $\eta$ is multiplied with a dampening factor which is inversely propositional to the dimension $D$. More concretely, the noise in the data is multiplied with a dampening factor that decreases when the ratio between the ambient dimension and the manifold dimension increases.*

To demonstrate Theorem 4.2.2, we use different auxiliary results. First, we prove in Theorem 4.2.5 that every Lipschitz continuous function can be robust-approximated by ReLU NNs. In the proof of such a theorem, we explicitly construct an approximating NN and provide upper bounds for its number of weights and layers. By robustness, we mean that the realizations are Lipschitz regular functions that dampen the noise. We leverage this construction to define the class of NNs $\mathcal{NN}_*$ from Theorem 4.2.2. The learning estimates are studied in Theorem 4.2.10.

### 4.2.1 Robust Approximation of Lipschitz Functions on Manifolds by ReLU Networks

Using Theorem 2.2.12, we will demonstrate in Theorem 4.2.5 below, that realizations of relatively small NNs can well approximate every Lipschitz continuous function $f \in L^\infty(\mathcal{M}; \mathbb{R}^q)$.

We first recall a result, which is a modification of Proposition 2.2.7 as we introduce the parameter $\theta$.

**Lemma 4.2.4** ([131, Lemma A.3]). *Let $\theta > 0$ be arbitrary. Then, for every $L \in \mathbb{N}$ with $L > \theta^{-1}$ and each $K \geq 1$, there are constants $c = c(L, K, \theta) \in \mathbb{N}$, $s = s(K) \in \mathbb{N}$, and an absolute constant $c' \in \mathbb{N}$ with the following property: For each $\epsilon \in (0, 1/2)$, there is an NN $\widetilde{\times}$ with $L(\widetilde{\times}) \leq c'L$, $W(\widetilde{\times}) \leq c\epsilon^{-\theta}$, and such that $\widetilde{\times}$ satisfies, for all $x, y \in [-K, K]$,*

- $\left|xy - \mathrm{R}\left(\widetilde{\times}\right)(x, y)\right| \leq \epsilon$,
- $\mathrm{R}_\varrho\left(\widetilde{\times}\right)(x, y) = 0$ *if* $xy = 0$.

In addition, we describe how robust the realizations of the approximating NNs are to noise in the ambient space. We will see that, in this regard, a large ambient dimension $D$ compared to the intrinsic dimension $d$ of the manifold will make the realization of the NN *more* robust to ambient noise. We state the result below. In this theorem, the distance on the manifold $\mathcal{M}$ is the restriction of the Euclidean distance from the ambient space.

**Theorem 4.2.5.** *Let $M$, $D$, $d$, $q \in \mathbb{N}$, $\delta > 0$ and let $\mathcal{M} \subset [0, 1]^D$ be an $(M, \delta)$-covered $d$-dimensional submanifold.*

*Then, for every Lipschitz continuous function $f : \mathcal{M} \to [0, 1]^q$ with Lipschitz constant $C > 0$ and every $\epsilon \in (0, 1)$ there is an NN $\Phi_{\mathcal{M}}^{f,\epsilon}$ with $D-$dimensional input such that the following holds*

- $\left\| f - \mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)\right\|_{L^\infty(\mathcal{M};\mathbb{R}^q)} \leq \epsilon$,
- $W\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right) = \mathcal{O}\left(qM^{d+1}\epsilon^{-d}\right)$ *for* $\epsilon \to 0$,
- $L\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right) = \lceil \log_2(d) \rceil + \lceil \log_2(D) \rceil + c$,

*where in the implicit constant in the estimates of the weights above there is an* additive *constant that depends on $\mathcal{M}$ (hence on $D$) and $C$, and $c > 0$ is a universal constant. Moreover, $\mathrm{R}(\Phi_{\mathcal{M}}^{f,\epsilon})([0, 1]^D) \subset [0, 1]^q$.*

*Additionally, there exists $r_0 > 0$ such that for Gaussian random variables $\eta = (\eta_i)_{i=1}^D$ where all components are independent, zero-mean, and satisfy $\mathbb{E}(|\eta_i|^2) = \sigma^2 \leq r_0^2$, it holds that*

$$\mathbb{E}\left(\sup_{\mathbf{x}\in\mathcal{M}}\left|\mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x}+\eta) - \mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x})\right|^2\right) \leq \frac{\hat{C}}{D}\mathbb{E}(|\eta|^2) \cdot \left(\sqrt{2d\log(M)} + 2\log(M) + d\right) \tag{4.2.4}$$

$$+ 8\epsilon^2 + 2De^{-\frac{r_0^2}{2\sigma^2}},$$

$$= \hat{C}\sigma^2 \cdot \left(\sqrt{2d\log(M)} + 2\log(M) + d\right) \tag{4.2.5}$$

$$+ 8\epsilon^2 + 2De^{-\frac{r_0^2}{2\sigma^2}},$$

*where the implicit constant $\hat{C} > 0$ depends on $C$ only.*

**Remark 4.2.6.** *1. Theorem 4.2.5 demonstrates that the inverse problem can be approximately solved using an NN of moderate size. Notably, the size of the NN increases asymptotically for $\epsilon \to 0$ and the size and depth depend weakly on $D$ (through additive constants) and strongly on $d$ (in the form of the approximation rate).*

*Note that, in this model, the variance of the noise is $D\sigma^2$. Hence, for $\sigma$ constant, the variance of the noise in the data scales linearly with $D$, while the variance of the noise in the output of the neural network is essentially independent of $D$ as we can see in* (4.2.5).

*2. We will demonstrate in the proof of Theorem 4.2.5 that $\hat{C} \leq 8C^2$. In practice, one can expect that increasing the number of measurements, i.e. $D$ implies that the Lipschitz constant of the inverse operator will decrease. This suggests that* the stability of the realizations of the NNs will increase with increasing dimension $D$ even beyond the dampening factor $d/D$. *We will observe this phenomenon in the numerical experiments in Section 4.4.*

**Remark 4.2.7.** *The proof of Theorem 4.2.5, which is given below, is based on a concrete construction of an NN. This construction is based on first applying a partition of unity on $[0,1]^D$ to localize the function $f$ and then locally writing $f$ as smooth functions composed with projections onto low dimensional sets. This idea has been used repeatedly, such as, for example, in [32, 142, 145, 109]. However, a stability statement, such as the one of* (4.2.4), *is not included in any of the references above.*

*Since this is our focus, our construction differs from the constructions in [32, 142, 145, 109]. In particular, we require an exact and very localized partition of unity for which we use results of [61], which has not been used in the results above. Moreover, we attempt to produce approximation results where the depth is independent of the accuracy, which is not done in [32, 142]. The approximation of the localized functions in [145], is based on wavelet-type approximation and [32, 142, 109] on Taylor polynomials. Instead, our result uses finite element-based approximation.*

*Proof.* Since $\mathcal{M}$ is $(M, \delta)$-covered, there exists a set $Y^{\mathcal{M}} := (\mathbf{y}_i)_{i=1}^M \subset \mathcal{M}$ such that

$$\bigcup_{i=1}^{M} B_{\frac{\delta}{4}}(\mathbf{y}_i) \supset \mathcal{M}.$$

Moreover, for every $\mathbf{y}_i \in Y^{\mathcal{M}}$, the orthogonal projection onto the tangent space $T_{\mathbf{y}_i}\mathcal{M}$ of $\mathcal{M}$ at $\mathbf{y}_i$ is a diffeomorphism and its inverse, denoted by $\widetilde{P}_{\mathbf{y}_i}$, is Lipschitz continuous with Lipschitz constant bounded by 2. By the triangle inequality, it follows that

$$\bigcup_{i=1}^{M} B_{\frac{\delta}{2}}(\mathbf{y}_i) \supset \mathcal{M} + B_{\frac{\delta}{4}}(0).$$

Let $G_\delta := (\delta/(8\sqrt{D}))\mathbb{Z}^D$ be a uniform grid in $\mathbb{R}^D$ with grid size $\delta/(8\sqrt{D})$. For $\mathbf{z} \in G_\delta$, we define $\phi^\delta_{\mathbf{z}}$ as the linear finite element function that equals 1 on $z$ and vanishes on all other elements of $G_\delta$. It is well-known and not hard to see that $(\phi^\delta_{\mathbf{z}})_{z\in G_\delta}$ forms a partition of unity, supp $\phi^\delta_{\mathbf{z}} \subset B_{\delta/4}(\mathbf{z})$, and by the boundedness of $\mathcal{M}$ there exists $Z^{\mathcal{M}} = (\mathbf{z}_i)_{i=1}^{M_\delta} \subset G_\delta$ such that

$$\sum_{i=1}^{M_\delta} \phi^\delta_{\mathbf{z}_i}(\mathbf{x}) = 1 \text{ for all } \mathbf{x} \in \mathcal{M} + B_{\delta/4}(0) \tag{4.2.6}$$

and supp $\phi^\delta_{\mathbf{z}_i} \subset \bigcup_{i=1}^{M_\delta} B_{3\delta/4}(\mathbf{y}_i)$.

By these considerations, it is clear that for each $\mathbf{z} \in Z^{\mathcal{M}}$ there exists $\mathbf{y}(\mathbf{z}) \in Y^{\mathcal{M}}$ such that supp $\phi^\delta_{\mathbf{z}} \subset B_\delta(\mathbf{y}(\mathbf{z}))$. Therefore, we can write $f$ in the following localized form: for all $\mathbf{x} \in \mathcal{M}$,

$$\begin{aligned} f(\mathbf{x}) = \sum_{i=1}^{M_\delta} \phi^\delta_{\mathbf{z}_i}(\mathbf{x}) f(\mathbf{x}) &= \sum_{i=1}^{M_\delta} \phi^\delta_{\mathbf{z}_i}(\mathbf{x}) f\left(\widetilde{P}_{y(\mathbf{z}_i)}\left(\mathrm{P}_{y(\mathbf{z}_i),\mathcal{M}}(\mathbf{x})\right)\right) \\ &= \sum_{i=1}^{M} \left( \sum_{\substack{z\in Z^{\mathcal{M}} \\ \mathbf{y}(\mathbf{z})=\mathbf{y}_i}} \phi^\delta_z(\mathbf{x}) \right) f\left(\widetilde{P}_{\mathbf{y}_i}\left(\mathrm{P}_{\mathbf{y}_i,\mathcal{M}}(\mathbf{x})\right)\right) \\ &=: \sum_{i=1}^{M} \phi_i(\mathbf{x}) f\left(\widetilde{P}_{\mathbf{y}_i}\left(\mathrm{P}_{\mathbf{y}_i,\mathcal{M}}(\mathbf{x})\right)\right). \end{aligned} \tag{4.2.7}$$

We define, for $i \in \{1,\dots,M\}$, $\widehat{f_i} := f \circ \widetilde{P}_{\mathbf{y}_i}$. It is not hard to see that $\widehat{f_i}$ is Lipschitz continuous with Lipschitz constant $2C$. Additionally, we define for $i \in \{1,\dots,M\}$

$$\begin{aligned} \widetilde{f_i} : [0,1] \times V_{\mathbf{y}_i} &\to \mathbb{R}^q, \\ (a,\mathbf{b}) &\mapsto a\widehat{f_i}(\mathbf{b}). \end{aligned}$$

We have that

$$f(\mathbf{x}) = \sum_{i=1}^{M} \widetilde{f_i}\left(\phi_i(\mathbf{x}), \mathrm{P}_{\mathbf{y}_i,\mathcal{M}}(\mathbf{x})\right), \text{ for all } x \in \mathcal{M}. \tag{4.2.8}$$

Without loss of generality, we can think of $V_{\mathbf{y}_i}$ as a compact interval $[-K,K]^D$, since every Lipschitz function can be extended without increasing its Lipschitz constant according to [100]. By Theorem 2.2.12, there exists for every $\epsilon > 0$ and every $i \in \{1,\dots,M\}$ an NN $\widehat{\Phi}^{i,\epsilon}$ with $d$-dimensional input such that

- $\left\| \mathrm{R}\left(\widehat{\Phi}^{i,\epsilon}\right) - \widehat{f_i} \right\|_{L^\infty([0,1]\times V_{\mathbf{y}_i};\mathbb{R}^q)} \leq \epsilon$,
- $W\left(\widehat{\Phi}^{i,\epsilon}\right) = \mathcal{O}\left(q\epsilon^{-d}\right)$ for $\epsilon \to 0$,

- $L\left(\widehat{\Phi}^{i,\epsilon}\right) = \lceil \log_2(d+1) \rceil + 1$.

Applying Lemma 4.2.4 with $\theta = d$ yields an NN that can be concatenated with $\widehat{\Phi}^{i,\epsilon}$ through Definition 2.0.5. This yields that, for every $\epsilon > 0$, there exists an NN $\Phi^{\widetilde{f}_i,\epsilon}$ with $(d+1)$-dimensional input such that

- $\left|\mathrm{R}\left(\Phi^{\widetilde{f}_i,\epsilon}\right)(a,\mathbf{b}) - a\mathrm{R}\left(\widehat{\Phi}^{i,\epsilon}\right)(\mathbf{b})\right| \leq \epsilon$ for all $a \in [0,1], \mathbf{b} \in V_{\mathbf{y}_i}$, (4.2.9)
- $W\left(\Phi^{\widetilde{f}_i,\epsilon}\right) = \mathcal{O}\left(q\epsilon^{-d}\right)$ for $\epsilon \to 0$,
- $L\left(\Phi^{\widetilde{f}_i,\epsilon}\right) = \lceil \log_2(d) \rceil + c$,

where $c > 0$ is a universal constant. Note that the implicit constant in the estimate of the number of weights is independent from $D$. Moreover, by [61, Theorem 3.1] we have that, for every $i \in \{1, \ldots, M_\delta\}$ there exists an NN $\Phi^\delta_{\mathbf{z}_i}$ with $D$-dimensional input such that

- $\mathrm{R}\left(\Phi^\delta_{\mathbf{z}_i}\right) = \phi^\delta_i$,
- $W\left(\Phi^\delta_{\mathbf{z}_i}\right) = \mathcal{O}(D)$,
- $L\left(\Phi^\delta_{\mathbf{z}_i}\right) = \lceil \log_2(D+1) \rceil + 1$.

Therefore, we conclude that for every $i \in \{1, \ldots, M\}$ there exists an NN $\Phi_i$ with $D$-dimensional input such that

- $\mathrm{R}\left(\Phi_i\right) = \phi_i$,
- $W\left(\Phi_i\right) = \mathcal{O}\left(M_\delta D\right)$,
- $L\left(\Phi_i\right) = \lceil \log_2(D+1) \rceil + 1$.

The estimate on the weights will later only contribute to the overall bound on the weights as an additive constant.

Finally, as $\mathrm{P}_{\mathbf{y}_i,\mathcal{M}}$ is affine linear it can be represented as a one-layer NN $\overline{\Phi}_{P,i,\mathcal{M}}$ with $D$-dimensional input and $d$-dimensional output. It follows directly from Definition 2.0.5 and Proposition 2.0.8 that $\overline{\Phi}_{P,i,\mathcal{M}}$ can be extended to have depth $L(\Phi_i)$, i.e., there exists an NN $\Phi_{P,i,\mathcal{M}}$ such that

- $\mathrm{R}\left(\Phi_{P,i,\mathcal{M}}\right) = \mathrm{P}_{\mathbf{y}_i,\mathcal{M}}$,
- $W\left(\Phi_{P,i,\mathcal{M}}\right) = \mathcal{O}\left(Dd + d\log_2(D)\right) = \mathcal{O}\left(dD\right)$,
- $L\left(\Phi_{P,i,\mathcal{M}}\right) = L\left(\Phi_i\right)$.

We define for every $i \in \{1, \dots, M\}$

$$\widetilde{\Phi}^{i,\epsilon} \coloneqq \Phi^{\widetilde{f}_i, \epsilon/M} \odot \mathrm{P}\left(\Phi_i, \Phi_{P,i,\mathcal{M}}\right),$$

and

$$\Phi_{\mathcal{M}}^{f,\epsilon} \coloneqq \Phi^1 \odot \mathrm{FP}\left(\widetilde{\Phi}^{1,\epsilon}, \widetilde{\Phi}^{2,\epsilon}, \dots, \widetilde{\Phi}^{M,\epsilon}\right),$$

where $\Phi^1 \coloneqq ((1_{\mathbb{R}^{1,M}}, 0))$ and $1_{\mathbb{R}^{1,M}}$ is a row vector of length $M$ with all entries equal to 1. It is clear from the construction of $\Phi_{\mathcal{M}}^{f,\epsilon}$ and (4.2.8) that

$$\left\| f - \mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right) \right\|_{L^\infty(\mathcal{M};\mathbb{R}^q)} = \left\| \sum_{i=1}^{M} \widetilde{f}_i \left(\phi_i(\cdot), \mathrm{P}_{\mathbf{y}_i,\mathcal{M}} \cdot\right) - \mathrm{R}\left(\widetilde{\Phi}^{i,\epsilon}\right) \right\|_{L^\infty(\mathcal{M};\mathbb{R}^q)} \tag{4.2.10}$$

$$= \left\| \sum_{i=1}^{M} \widetilde{f}_i \left(\phi_i(\cdot), \mathrm{P}_{\mathbf{y}_i,\mathcal{M}} \cdot\right) - \mathrm{R}\left(\Phi^{\widetilde{f}_i, \epsilon/M}\right) \left(\phi_i(\cdot), \mathrm{P}_{\mathbf{y}_i,\mathcal{M}} \cdot\right) \right\|_{L^\infty(\mathcal{M};\mathbb{R}^q)} \leq \epsilon, \tag{4.2.11}$$

where we applied the triangle inequality to obtain the final estimate. Moreover, by construction, $W(\Phi_{\mathcal{M}}^{f,\epsilon}) = \mathcal{O}(qM^{d+1}\epsilon^{-d}) + \mathcal{O}(M \cdot (M_\delta D + dD))$ and $L(\Phi_{\mathcal{M}}^{f,\epsilon}) = \lceil \log_2(d) \rceil + \lceil \log_2(D) \rceil + c + 3$.

We assume without loss of generality that $\Phi_{\mathcal{M}}^{f,\epsilon}$ already satisfies that $\mathrm{R}(\Phi_{\mathcal{M}}^{f,\epsilon})$ maps into $[0,1]^q$. If this is not the case, then concatenating $\Phi_{\mathcal{M}}^{f,\epsilon}$ with a simple NN the realization of which implements the function $(x_i)_{i=1}^q \mapsto (\min\{\max\{0, x_i\}, 1\})_{i=1}^q$ would enforce the desired property for the concatenated NN. This concatenation does not increase the number of weights or the layers by a significant amount.

Finally, we demonstrate statement (4.2.4). First of all, set $r_0 \coloneqq \min\{\delta/(2\sqrt{D}), 1\}$. Then, for any random vector $\eta = (\eta_j)_{j=1}^D$ where each component $\eta_j$ with $j \in [D]$ is a normally distributed random variable with mean zero and $\sigma^2 = \mathbb{E}(|\eta_j|^2) \leq r_0^2$ the following upper bound holds

$$\mathbb{P}(|\eta_1| > r_0 \vee |\eta_2| > r_0 \vee \ldots \vee |\eta_D| > r_0) \leq \sum_{i=1}^{D} \mathbb{P}(|\eta_i| > r_0) < 2De^{-r_0^2/(2\sigma^2)} =: p.$$

Therefore, with probability $1 - p$, it holds that for all $x \in \mathcal{M}$

$$\begin{aligned} \left| \mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x}) - \mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x} + \eta) \right| \leq & \left| \mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x}) - \sum_{i=1}^{M} \phi_i(\mathbf{x}) \widehat{f}_i \left(\mathrm{P}_{\mathbf{y}_i,\mathcal{M}}(\mathbf{x})\right) \right| \\ & + \left| \sum_{i=1}^{M} \phi_i(\mathbf{x}) \widehat{f}_i \left(\mathrm{P}_{\mathbf{y}_i,\mathcal{M}}(\mathbf{x})\right) - \sum_{i=1}^{M} \phi_i(\mathbf{x} + \eta) \widehat{f}_i \left(\mathrm{P}_{\mathbf{y}_i,\mathcal{M}}(\mathbf{x} + \eta)\right) \right| \\ & + \left| \sum_{i=1}^{M} \phi_i(\mathbf{x} + \eta) \widehat{f}_i \left(\mathrm{P}_{\mathbf{y}_i,\mathcal{M}}(\mathbf{x} + \eta)\right) - \mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x} + \eta) \right| \end{aligned}$$

$$
\begin{aligned}
&\leq \left| \sum_{i=1}^{M} \phi_i(\mathbf{x}) \widehat{f_i} \left( \mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(\mathbf{x}) \right) - \sum_{i=1}^{M} \phi_i(\mathbf{x}+\eta) \widehat{f_i} \left( \mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(\mathbf{x}+\eta) \right) \right| + 2\epsilon \\
&\leq \left| \sum_{i=1}^{M} \phi_i(\mathbf{x}) \widehat{f_i} \left( \mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(\mathbf{x}) \right) - \phi_i(\mathbf{x}+\eta)) \widehat{f_i} \left( \mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(\mathbf{x}) \right) \right. \\
&\qquad \left. + \sum_{i=1}^{M} \phi_i(\mathbf{x}+\eta) \left( \widehat{f_i} \left( \mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(\mathbf{x}) \right) - \widehat{f_i} \left( \mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(\mathbf{x}+\eta) \right) \right) \right| + 2\epsilon \\
&\leq \left| \sum_{i=1}^{M} (\phi_i(\mathbf{x}) - \phi_i(\mathbf{x}+\eta)) f\left(x\right) \right| \\
&\qquad + \left| \sum_{i=1}^{M} \phi_i(x+\eta) \left( \widehat{f_i} \left( \mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(\mathbf{x}) \right) - \widehat{f_i} \left( \mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(\mathbf{x}+\eta) \right) \right) \right| + 2\epsilon \\
&=: \mathrm{I} + \mathrm{II} + 2\epsilon,
\end{aligned}
$$

where we applied the triangle inequality in the first, (4.2.11) in the second, and the definition of $\widehat{f_i}$ in the third step. From (4.2.6), we conclude that

$$
\sum_{i=1}^{M} \phi_i(\mathbf{x}+\eta) = \sum_{i=1}^{M} \phi_i \mathbf{x} = 1
$$

and therefore, $\mathrm{I} = 0$. Also, due to the linearity of projections and the Lipschitz continuity of $\widehat{f_i}$ we have for $\widetilde{C} = 2C$ that

$$
\begin{aligned}
\mathrm{II} &\leq \widetilde{C} \sum_{i=1}^{M} \phi_i(\mathbf{x}+\eta) |\mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(\eta)| \\
&\leq \widetilde{C} \sum_{i=1}^{M} \phi_i(\mathbf{x}+\eta) \cdot \left( \max_{j \in [M]} |\mathrm{P}_{y_j, \mathcal{M}}(\eta)| \right) \\
&= \widetilde{C} \max_{j \in [M]} |\mathrm{P}_{\mathbf{y}_j, \mathcal{M}}(\eta)|.
\end{aligned}
$$

Thus,

$$
\sup_{\mathbf{x} \in \mathcal{M}} \left| \mathrm{R} \left( \Phi_{\mathcal{M}}^{f,\epsilon} \right) (\mathbf{x}+\eta) - \mathrm{R} \left( \Phi_{\mathcal{M}}^{f,\epsilon} \right) (\mathbf{x}) \right|^2 = (\mathrm{II} + \epsilon)^2 \leq \left( \widetilde{C} \max_{j \in [M]} |\mathrm{P}_{\mathbf{x}_j, \mathcal{M}}(\eta)| + 2\varepsilon \right)^2.
$$

After the estimate $(z + 2\epsilon)^2 \leq 2z^2 + 8\epsilon^2$ is applied, we obtain

$$
\sup_{\mathbf{x} \in \mathcal{M}} \left| \mathrm{R} \left( \Phi_{\mathcal{M}}^{f,\epsilon} \right) (\mathbf{x}+\eta) - \mathrm{R} \left( \Phi_{\mathcal{M}}^{f,\epsilon} \right) (\mathbf{x}) \right|^2 \leq 2 \left( \widetilde{C} \max_{j \in [M]} |\mathrm{P}_{\mathbf{y}_j, \mathcal{M}}(\eta)| \right)^2 + 8\varepsilon^2. \tag{4.2.12}
$$

We observe due to the boundedness of $\mathrm{R}(\Phi_{\mathcal{M}}^{f,\epsilon})$ that

$$\mathbb{E}\left(\sup_{\mathbf{x}\in\mathcal{M}}\left|\mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x}+\eta)-\mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x})\right|^2\right)\leq\mathbb{E}\left(2\left(\widetilde{C}\max_{j\in[M]}|\mathrm{P}_{\mathbf{y}_j,\mathcal{M}}(\eta)|\right)^2+8\varepsilon^2\right)+p. \tag{4.2.13}$$

Next, we will find a bound for the first term on the right-hand side of the inequality (4.2.13). As $\eta=(\eta_i)_{i=1}^D$ is normally distributed with zero mean and covariance matrix $\sigma^2 I$ , where $I$ denotes the identity matrix, it follows that $y=(\eta_i/\sigma)_{i=1}^D$ is normally distributed with zero mean and covariance matrix $I$. It is well-known that every orthogonal projection operator $P$ on a linear subspace $V$ is self-adjoint and idempotent. Let $P$ be an orthogonal projection operator mapping to a $d$-dimensional subspace, then by [55, Theorem 3.1], it holds that

$$\|P\mathbf{y}\|^2=\mathbf{y}^TP^TP\mathbf{y}=\mathbf{y}^TP^2\mathbf{y}=\mathbf{y}^TP\mathbf{y}$$

is $\chi_d^2$ distributed. Therefore, it can be concluded that $|\mathrm{P}_{\mathbf{y}_j,\mathcal{M}}(\eta)|^2/\sigma^2$ is a $\chi_d^2$ distributed random variable for $j\in[M]$. Moreover, according to the maximal inequality for $\chi_d^2$ random variables (see [21, Example 2.7]), it holds that

$$\mathbb{E}\left(\max_{j\in[M]}\frac{|\mathrm{P}_{\mathbf{y}_j,\mathcal{M}}(\eta)|^2}{\sigma^2}-d\right)\leq\sqrt{2d\log(M)}+2\log(M).$$

Hence, we observe that

$$\begin{aligned}\mathbb{E}\left(\left(\max_{j\in[M]}|\mathrm{P}_{\mathbf{y}_j,\mathcal{M}}(\eta)|\right)^2\right)&=\mathbb{E}\left(\max_{j\in[M]}|\mathrm{P}_{\mathbf{y}_j,\mathcal{M}}(\eta)|^2\right)\\&\leq\left(\sqrt{2d\log(M)}+2\log(M)+d\right)\cdot\sigma^2\\&\leq\left(\sqrt{2d\log(M)}+2\log(M)+d\right)\cdot\frac{1}{D}\mathbb{E}(|\eta|^2),\end{aligned} \tag{4.2.14}$$

and therefore

$$\begin{aligned}\mathbb{E}\left(\sup_{\mathbf{x}\in\mathcal{M}}\left|\mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x}+\eta)-\mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\epsilon}\right)(\mathbf{x})\right|^2\right)&\leq\mathbb{E}(2\mathrm{II}^2+8\epsilon^2)+p\\&\leq2\widetilde{C}^2\cdot\left(\sqrt{2d\log(M)}+2\log(M)+d\right)\cdot\frac{1}{D}\mathbb{E}(|\eta|^2)+8\epsilon^2+p\\&\leq\frac{\hat{C}}{D}\mathbb{E}(|\eta|^2)\cdot\left(\sqrt{2d\log(M)}+2\log(M)+d\right)+8\epsilon^2+p,\end{aligned}$$

where $\hat{C}=2\widetilde{C}^2=8C^2$. □

**Remark 4.2.8.** *The weights in the NNs used in the construction of Theorem 4.2.5 are all polynomially bounded in $\epsilon$. This can be seen by observing that the weights are polynomially bounded for all NNs used as building blocks. For the multiplication NN of Lemma 4.2.4 this is explicitly stated in [131, Lemma A.3]. For the NNs of Theorem 2.2.12, this bound*

*is not spelt out in the original reference, but is clear since linear FEM approximation is emulated, where the weights need to grow only like the inverse of the mesh-size. The same holds for the construction of the partition of unity in the proof. All additional operations require universally bounded weights only. In the sequel, we assume that the weights of the NNs of Theorem 4.2.5 are bounded by $\epsilon^{-r}$ for an $r \in \mathbb{N}$.*

Theorem 4.2.5 guarantees the existence of an NN the realization of which is robust with respect to noisy inputs. In practice, we also need to find this NN or one with similar properties by performing some form of empirical risk minimization over a set of NNs for an appropriately chosen training set.

If we attempt to create an NN with robustness to noise that implements the inverse to a forward operator $F\colon [0,1]^q \to \mathcal{M} \subset [0,1]^D$, for $q, D \in \mathbb{N}$, then it appears to be natural to train on noisy data, $(F(x_i)+\eta_i, x_i)_{i=1}^m$, where $m \in \mathbb{N}$ and $\eta_i$ are additive noise vectors. Note that, if $F$ is known, then such data can be readily generated.

However, the theoretically-established robust-to-noise NN of Theorem 4.2.5 is one that instead of approximating $F^{-1}$ approximates, for an $r_0 > 0$, the function

$$f(x) = \sum_{i=1}^{M} \phi_i(\mathbf{x}) \widehat{f}_i \left( \mathrm{P}_{\mathbf{y}_i, \mathcal{M}}(x) \right), \text{ for } \mathbf{x} \in \mathcal{M} + B_{r_0}(0),$$

where $\hat{f}_i$ are the Lipschitz continuous functions defined under Equation (4.2.7). Note that $f$ only necessarily agrees with $F^{-1}$ on $\mathcal{M}$. To learn $f$, it appears more appropriate to train on samples of the form $(\mathbf{y}+\eta, f(\mathbf{y}+\eta))$ for $y$ following some distribution on $\mathcal{M}$ and $\eta$ Gaussian. However, while one could in principle compute $f$ for many practical problems, it would be preferable to only access the forward operator $F$.

We will show below, that training on values $(F(\mathbf{x}_i)+\eta_i, \mathbf{x}_i)_{i=1}^m$ with $x_i$ drawn according to some distribution on $[0,1]^q$ and $\eta_i$ Gaussian yields a robust NN as well. Next, we state a bound for the generalization error of empirical risk minimization when multi-dimensional output is used. Similar results may exist in the literature, see e.g. [7, Theorem 10.1] for the univariate case. To keep the manuscript self-contained, we add a proof of the result.

**Proposition 4.2.9.** *Let $m, q, D, L, W \in \mathbb{N}$. Let $g\colon [0,1]^D \to [0,1]^q$ and let $\mathcal{D}$ be a distribution on $[0,1]^D$.*

*Then, for every $\Phi \in \mathcal{NN}_*(D, q, L, W, B)$ and every $\epsilon > 0$*

$$\begin{aligned} &\mathbb{P}_{(\mathbf{x}_i)_{i=1}^m \sim \mathcal{D}^m} \left( \mathbb{E}_{x\sim\mathcal{D}} |\mathrm{R}(\Phi)(\mathbf{x}) - g(\mathbf{x})|^2 - \frac{1}{m}\sum_{i=1}^m |\mathrm{R}(\Phi)(\mathbf{x}_i) - g(\mathbf{x}_i)|^2 > \epsilon \right) \\ &\quad \leq 2 \left( 44 \frac{2q}{\epsilon} L^4 (\lceil B \rceil \max\{q, W\})^{L+1} \right)^{Wq} e^{-m\epsilon^2/8}. \end{aligned} \tag{4.2.15}$$

*In particular, for $p \in (0,1)$ it holds with probability $1-p$ over the choice of $(\mathbf{x}_i)_{i=1}^m \sim \mathcal{D}^m$*

*that*

$$\mathbb{E}_{x\sim\mathcal{D}}|\mathrm{R}(\Phi)(\mathbf{x})-g(\mathbf{x})|^2 \tag{4.2.16}$$
$$\leq \frac{1}{m}\sum_{i=1}^{m}|\mathrm{R}(\Phi)(\mathbf{x}_i)-g(\mathbf{x}_i)|^2 + \sqrt{\frac{8+8Wq(5+\ln(q)+4\ln(L)+\ln(1/\epsilon)+(L+1)(\ln\lceil B\rceil+\ln\max\{q,W\}))}{m}+\frac{8\ln(1/p)}{m}}. \tag{4.2.17}$$

*Proof.* We first observe that $\mathcal{RNN}_*(D,q,L,W,B)\subset\bigotimes_{j=1}^{q}\mathcal{RNN}_*(D,1,L,W,B)$ and hence, by [58, Lemma 6.1], that $\mathcal{RNN}_*(D,q,L,W,B)$ can be covered by

$$\left(\frac{44}{\epsilon}L^4(\lceil B\rceil\max\{q,W\})^{L+1}\right)^{Wq}$$

balls in $L^\infty([0,1]^D,[0,1]^q)$ of radius $\epsilon>0$, where the $L^\infty$ norm of a function $f:[0,1]^D\to[0,1]^q$ is defined as

$$\|f\|_\infty \coloneqq \sup_{k=1,\dots,q}\|f_k\|_\infty.$$

For $g\colon[0,1]^D\to[0,1]^q$, it follows that the function class

$$\mathcal{H}_{q,g}\coloneqq\left\{x\mapsto\sum_{i=1}^{q}|(h(\mathbf{x}))_i-(g(\mathbf{x}))_i|^2\colon h\in\mathcal{RNN}_*(D,q,L,W,B)\right\}$$

can be covered by $((44/\epsilon)L^4(\lceil B\rceil\max\{q,W\})^{L+1})^{Wq}$ balls in $L^\infty([0,1]^D)$ of radius $2q\epsilon$ since the map

$$K_q\colon L^\infty([0,1]^D,[0,1]^q)\to L^\infty([0,1]^D)$$
$$h\mapsto K_q(h)=\sum_{i=1}^{q}|h_i-g_i|^2$$

satisfies for all $h^{(1)},h^{(2)}\in L^\infty([0,1]^D,[0,1]^q)$ that

$$\begin{aligned}\left\|K_q(h^{(1)})-K_q(h^{(2)})\right\|_\infty &= \left\|\sum_{i=1}^{q}|h_i^{(1)}-g_i|^2-\sum_{i=1}^{q}|h_i^{(2)}-g_i|^2\right\|_\infty\\ &\leq\sum_{i=1}^{q}\left\||h_i^{(1)}-g_i|^2-|h_i^{(2)}-g_i|^2\right\|_\infty\leq 2q\left\|h^{(1)}-h^{(2)}\right\|_\infty.\end{aligned}\tag{4.2.18}$$

Here (4.2.18) holds since for $x>y$ and $x,y,b\in[0,1]$

$$(x-b)^2-(y-b)^2=\int_y^x 2(z-b)dz\leq 2(x-y).$$

Applying [7, Theorem 10.1] to $\mathcal{H}_{q,g}$ yields (4.2.15). Finally, to demonstrate (4.2.17), observe that $\ln(2 \cdot 44) \leq 5$, set

$$\epsilon = \sqrt{\frac{8 + 8Wq(5 + \ln(q) + 4\ln(L) + \ln(1/\epsilon) + (L+1)(\ln\lceil B \rceil + \ln\max\{q, W\}))}{m} + \frac{8\ln(1/p)}{m}},$$

and apply (4.2.15). □

Next we show that training an NN of a specific size on data of the form $(F(\mathbf{x}_i)+\eta_i, \mathbf{x}_i)_{i=1}^m$ with $x_i$ drawn according to some distribution on $\mathcal{M}$ and $\eta_i$ Gaussian generates an NN that has robustness properties similar to the NN of Theorem 4.2.5. In particular, the robustness to noise is independent from $D$ and a small value of $d$ dampens the noise.

**Theorem 4.2.10.** *Let $r, q, m, D \in \mathbb{N}$, $\mathcal{D}$ be a distribution on $K \subset [0,1]^q$, let $F\colon K \to \mathcal{M} \subset [0,1]^D$, be such that $\mathcal{M}$, $F^{-1}$ satisfy the assumptions of Theorem 4.2.5, let $c, C, d, M$ be as in Theorem 4.2.5, and let $r > 0$ be as in Remark 4.2.8. Let $\eta \in \mathbb{R}^D$ be a Gaussian random variable where all components are independent, zero-mean with variance $\sigma^2$. Let $S = (\mathbf{s}_i^{(1)}, \mathbf{s}_i^{(2)})_{i=1}^m = (F(\mathbf{x}_i) + \eta_i, \mathbf{x}_i)_{i=1}^m$ be a sample where $x_i \sim \mathcal{D}$ and $\eta_i \sim \eta$ are independent random variables for $i = 1, \ldots, m$. For $\kappa \geq C \cdot (\sqrt{2d\log(M)} + 2\log(M) + d) \cdot \sigma^2$, we choose*

$$W \sim qM^{d+1}\kappa^{-d/2}, \quad L \sim \lceil \log_2(d) \rceil + \lceil \log_2(D) \rceil + c, \quad B \sim \kappa^{-r/2},$$

*to be the values corresponding to $\epsilon = \sqrt{\kappa}$ in Theorem 4.2.5 and we let*

$$\Phi_S \in \underset{\Phi \in \mathcal{NN}_*(D,q,L,W,B)}{\arg\min} \left\{ \frac{1}{m} \sum_{i=1}^m \left| \mathrm{R}(\Phi)\left(\mathbf{s}_i^{(1)}\right) - \mathbf{s}_i^{(2)} \right|^2 \right\}. \tag{4.2.19}$$

*Then, with probability $1 - 2p$ over the choice of $S$*

$$\mathbb{E}_{x\sim\mathcal{D},\eta} \left| \mathrm{R}(\Phi_S)(F(\mathbf{x}) + \eta) - \mathbf{x} \right|^2 \leq \overline{C}\kappa + c' \sqrt{\frac{q\kappa^{-d/2}\ln(1/\kappa)}{m}} + 4De^{-\frac{r_0^2}{2\sigma^2}} + \sqrt{\frac{16\ln(2/p)}{m}}, \tag{4.2.20}$$

*where $c' > 0$ depends on $\mathcal{M}$ and $\overline{C} > 0$ depends only on $C$ (it is in particular independent from $D$, $d$, $q$, and $M$).*

*Proof.* First, we note that for $m \in \mathbb{N}$ and $p \in (0, 1/2)$ by Proposition 4.2.9 with probability at least $1 - p$ over the draw of $S = (F(\mathbf{x}_i) + \eta_i, \mathbf{x}_i)_{i=1}^m$ it holds that

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D},\eta} \left| \mathrm{R}(\Phi_S)(F(\mathbf{x}) + \eta) - \mathbf{x} \right|^2 \\
&\leq \frac{1}{m} \sum_{i=1}^m \left| \mathrm{R}(\Phi_S)(F(\mathbf{x}_i) + \eta_i) - \mathbf{x}_i \right|^2 \\
&\quad + \sqrt{\frac{8 + 8Wq(5 + \ln(q) + 4\ln(L) + \ln(1/\epsilon) + (L+1)(\ln\lceil B \rceil + \ln\max\{q, W\}))}{m} + \frac{8\ln(1/p)}{m}}.
\end{aligned} \tag{4.2.21}$$

Next, we note that per definition of $\Phi_S$

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^{m}|\mathrm{R}(\Phi_S)(F(\mathbf{x}_i)+\eta_i)-\mathbf{x}_i|^2 &= \inf_{\Phi\in\mathcal{NN}_*(D,q,L,W,B)}\frac{1}{m}\sum_{i=1}^{m}|\mathrm{R}(\Phi)(F(\mathbf{x}_i)+\eta_i)-\mathbf{x}_i|^2\\ &\leq \inf_{\Phi\in\mathcal{NN}_*(D,q,L,W,B)}\left(\frac{2}{m}\sum_{i=1}^{m}|\mathrm{R}(\Phi)(F(\mathbf{x}_i)+\eta_i)-\mathrm{R}(\Phi)(F(\mathbf{x}_i))|^2\right.\\ &\qquad\left.+\frac{2}{m}\sum_{i=1}^{m}|\mathrm{R}(\Phi)(F(x_i))-x_i|^2\right). \end{aligned}\tag{4.2.22}$$

By Theorem 4.2.5, it holds that $\Phi_{\mathcal{M}}^{f,\sqrt{\kappa}}\in\mathcal{NN}_*(D,q,L,W,B)$ and hence

$$\frac{2}{m}\sum_{i=1}^{m}\left|\mathrm{R}\left(\Phi_{\mathcal{M}}^{f,\sqrt{\kappa}}\right)(F(\mathbf{x}_i))-\mathbf{x}_i\right|^2\leq 2\kappa.\tag{4.2.23}$$

Next, we denote for $\Phi\in\mathcal{NN}_*(D,q,L,W,B)$

$$\begin{aligned}e_S(\Phi)&\coloneqq\frac{1}{m}\sum_{i=1}^{m}|\mathrm{R}(\Phi)(F(\mathbf{x}_i)+\eta_i)-\mathrm{R}(\Phi)(F(\mathbf{x}_i))|^2,\\ e_\infty(\Phi)&\coloneqq\mathbb{E}_{x\sim\mathcal{D},\eta}|\mathrm{R}(\Phi)(F(\mathbf{x})+\eta)-\mathrm{R}(\Phi)(F(\mathbf{x}))|^2.\end{aligned}$$

By Hoeffding's inequality, we have that

$$\mathbb{P}\left(e_S(\Phi)-e_\infty(\Phi)>\sqrt{\ln(2/p)/m}\right)\leq p.\tag{4.2.24}$$

We conclude that with probability at least $1-p$ by (4.2.22)

$$\frac{1}{m}\sum_{i=1}^{m}|\mathrm{R}(\Phi_S)(F(\mathbf{x}_i)+\eta_i)-\mathbf{x}_i|^2\leq 2e_\infty\left(\Phi_{\mathcal{M}}^{f,\sqrt{\kappa}}\right)+2\kappa+2\sqrt{\frac{\ln(2/p)}{m}}.\tag{4.2.25}$$

By Theorem 4.2.5, we can estimate

$$\begin{aligned}e_\infty\left(\Phi_{\mathcal{M}}^{f,\sqrt{\kappa}}\right)&\leq\frac{\hat{C}}{D}\mathbb{E}(|\eta|^2)\cdot\left(\sqrt{2d\log(M)}+2\log(M)+d\right)+8\kappa+2De^{-\frac{r_0^2}{2\sigma^2}}\\ &\leq C'\kappa+2De^{-\frac{r_0^2}{2\sigma^2}},\end{aligned}\tag{4.2.26}$$

for a constant $C'=\hat{C}+8>0$.

Applying (4.2.26) to (4.2.25) yields that with probability at least $1-p$

$$\frac{1}{m}\sum_{i=1}^{m}|\mathrm{R}(\Phi_S)(F(\mathbf{x}_i)+\eta_i)-\mathbf{x}_i|^2\leq C''\kappa+4De^{-\frac{r_0^2}{2\sigma^2}}+2\sqrt{\frac{\ln(2/p)}{m}}.\tag{4.2.27}$$

Combining (4.2.27) with (4.2.21) and noting that since both estimates hold with probability at least $1-p$ the overall estimate holds with probability at least $1-2p$ completes the proof. □

## 4.3 Admissible Inverse Problems

Below, we will present a list of inverse problems that can be discretized in the way necessary for our theory to work. We discuss the transmissivity coefficient identification in Subsection 4.3.1, as well as the coefficient identification problem of the Euler-Bernoulli equation in Section 4.3.2. Thereafter, we discuss inverse problems associated to the Volterra-Hammerstein integral equation and the gravimetric problem in Sections 4.3.3 and 4.3.4, respectively.

### 4.3.1 Identification of the Transmissivity Coefficient

In this section, we introduce a problem arising from the study of aquifers (see [116]) in groundwater hydraulics. An aquifer is a natural underground formation capable of storing and transmitting underground water. Mathematically speaking, an aquifer is a three dimensional set $V \subset \mathbb{R}^3$ with an associated potential energy per unit weight: $u \in C^2(V)$. We can consider an aquifer as a one-dimensional object if its width and depth are negligible compared to its length or if its energy varies only with respect to one of its dimensions. In that case, the aquifer can be thought of as being the interval $I \coloneqq [0,1]$ where 0 and 1 are its endpoints and $x$ is referred to as the position or distance with respect to the first endpoint. Under certain circumstances, the energy transported by the water, $u \in C^2([0,1])$, is modeled by the equation

$$\frac{d}{dx}\left(a(x)u'(x)\right) = f(x) \text{ for all } x \in [0,1],$$

where the coefficient $a \in C^1([0,1])$ is referred to as the *transmissivity coefficient* which measures the ability of the aquifer to allow the passing of groundwater and $f \in C([0,1])$ measures the change of mass over time at a given point [173]. If $a \in C([0,1])$, then we assume the product of $a$ and $u'$ to be continuously differentiable. Solving the direct problem is to find $u$ given $f$ and $a$ under certain boundary or initial conditions. The inverse problem is then to find the coefficient $a$ when $u$ and $f$ are known. Note that if $u' > 0$ and the value $c_0 = a(0)u'(0)$ with $c_0 \in \mathbb{R}$ is also known, the coefficient $a(x)$ at the point $x \in [0,1]$ can be directly reconstructed by the formula

$$a(x) = \frac{\int_0^x f(t)dt + c_0}{u'(x)}.$$

Nevertheless, this formula may be useless if noise is in the data or only point samples of the functions $f$, $u$ are known. If in addition to the condition $c_0 = a(0)u'(0) \in \mathbb{R}$ mentioned above, a second condition $c_1 = u(0) \in \mathbb{R}$ is given, we express $u$ as a function of $a$ through the forward operator $F : H_\lambda \subset C([0,1]) \to C([0,1])$

$$u = F(a) \coloneqq \left(x \mapsto \int_0^x \frac{\int_0^z f(s)ds + c_0}{a(z)}dz + c_1\right), \tag{4.3.1}$$

where $H_\lambda \coloneqq \{a \in C([0,1]) : a(x) > \lambda > 0 \text{ for all } x \in [0,1]\}$ for fixed $\lambda > 0$. Notice that $H_\lambda$ is an open set and if $f > 0$ and $c_0 = c_1 = 0$, then $F$ is an injective operator on $H_\lambda$.

Moreover, we prove that this is a Fréchet-differentiable operator where its Fréchet derivative at every $a \in H_\lambda$ is computed below, via the Gateaux derivative at $a$ evaluated at $x \in [0,1]$ along the direction $\delta a \in C([0,1])$ denoted by $D_{\delta a}F(a)$.

**Proposition 4.3.1.** *Let $\lambda > 0, H_\lambda := \{a \in C([0,1]) : a(x) > \lambda > 0$ for all $x \in [0,1]\}$ and $F : H_\lambda \to C([0,1])$ be the operator defined by*

$$F(a) := \left( x \mapsto \int_0^x \frac{\int_0^z f(s)ds}{a(z)} dz \right), \tag{4.3.2}$$

*for all $a \in H_\lambda$ and $x \in [0,1]$. Then, $F$ is Gateaux-differentiable.*

*Proof.* Due to the definition of the Gateaux derivative at $a \in H_\lambda$ in the direction $\delta a \in C([0,1])$, we have for $x \in [0,1]$ that

$$\begin{aligned}
D_{\delta a}F(a)(x) &= \lim_{t\to 0} \frac{F(a+t\delta a)(x) - F(a)(x)}{t} \\
&= \lim_{t\to 0} t^{-1} \cdot \left( \int_0^x \frac{\int_0^z f(s)ds}{a(z)+t\delta a(z)} dz - \frac{\int_0^z f(s)ds}{a(z)} dz \right) \\
&= \lim_{t\to 0} t^{-1} \cdot \left( \int_0^x \int_0^z f(s)ds \left( \frac{1}{a(z)+t\delta a(z)} - \frac{1}{a(z)} \right) dz \right) \\
&= \lim_{t\to 0} t^{-1} \cdot \left( t \int_0^x \int_0^z f(s)ds \left( \frac{-\delta a(z)}{a(z)(a(z)+t\delta a(z))} \right) dz \right) \\
&= \lim_{t\to 0} \int_0^x \frac{-\delta a(z) \int_0^z f(s)ds}{a(z)(a(z)+t\delta a(z))} dz \\
&= -\int_0^x \frac{\delta a(z) \int_0^z f(s)}{a(z)^2} dz,
\end{aligned}$$

where the last equality is an application of the dominated convergence theorem.

□

**Proposition 4.3.2.** *Under the assumptions of Proposition 4.3.1, the operator $F$ defined by* (4.3.2) *is Fréchet-differentiable.*

*Proof.* Now, we prove that the Gateaux derivative at $a$ is indeed the Fréchet derivative of the operator at $a \in H_\lambda$. By the properties of the absolute value

$$
\begin{aligned}
&\left|\int_0^x \frac{\int_0^z f(s)ds}{a(z)+\delta a(z)}dz - \left(\int_0^x \frac{\int_0^z f(s)ds}{a(z)}dz\right) - \left(-\int_0^x \frac{\delta a(z)\int_0^z f(s)ds}{a(z)^2}dz\right)\right| \\
&= \left|\int_0^x \int_0^z f(s)ds \left(\frac{-\delta a(z)}{a(z)(a(z)+\delta a(z))}\right)dz + \int_0^x \frac{\delta a(z)\int_0^z f(s)ds}{a(z)^2}dz\right| \\
&= \left|\int_0^x \left(\int_0^z f(s)ds\right)\left(\frac{(\delta a(z))^2}{a(z)^2(a(z)+\delta a(z))}\right)dz\right| \\
&\leq \|\delta a\|_\infty^2 \sup_{x\in[0,1]}\left|\int_0^x \left(\int_0^z f(s)ds\right)\left(\frac{dz}{a(z)^2(a(z)+\delta a(z))}\right)\right|,
\end{aligned}
$$

and therefore

$$
\begin{aligned}
&\lim_{\delta a\to 0}\frac{\|F(a+\delta a)-F(a)-D_{\delta a}F(a)\|_\infty}{\|\delta a\|_\infty} \\
&\leq \lim_{\delta a\to 0}\|\delta a\|_\infty^2 \sup_{x\in[0,1]}\left|\int_0^x \left(\int_0^z f(s)ds\right)\left(\frac{dz}{a(z)^2(a(z)+\delta a(z))}\right)\right| / \|\delta a\|_\infty = 0,
\end{aligned}
$$

since for $\|\delta a\|_\infty$ small enough, we have that

$$
\sup_{x\in[0,1]}\left|\int_0^x \left(\int_0^z f(s)ds\right)\left(\frac{dz}{a(z)^2(a(z)+\delta a(z))}\right)\right|
$$

is bounded. Therefore, the Fréchet derivative for the operator $F$ at $a$ denoted by $F_a' : C([0,1]) \to C([0,1])$ is given by

$$
F_a'(\delta a) = \left(x \mapsto -\int_0^x \frac{\delta a(z)\int_0^z f(s)ds}{a(z)^2}dz\right) \quad \text{for} \quad \delta a \in C([0,1]).
$$

□

**Proposition 4.3.3.** *Under the assumptions of Proposition 4.3.1, let us consider the operator $F$ defined by (4.3.2). Let $F_a'$ denote the Fréchet derivative of $F$ at $a \in H_\lambda$. Then, $F_a'$is injective for all $a \in H_\lambda$.*

*Proof.* As $F_a'$ is a linear operator, it is enough to prove that if $F_a'(\delta a) = 0$ for $\delta a \in C([0,1])$, then $\delta a = 0$. If for all $x \in [0,1]$

$$
F_a'(\delta a)(x) = -\int_0^x \frac{\delta a(z)\int_0^z f(s)ds}{a(z)^2}dz = 0,
$$

then, due to the continuity of the integrand and by the fundamental theorem of calculus, we have that

$$
\frac{\delta a(z)\int_0^z f(s)ds}{a(z)^2} = 0
$$

which shows that $\delta a(z) = 0$ for all $z \in [0,1]$.

□

We have observed above that the operator $F$ is injective and Fréchet differentiable and $F'_a$ is injective for all $a \in H_\lambda$.

Next, we introduce the operator $P_{W,\lambda} : [a,b]^d \subset \mathbb{R}^d \to W + \lambda$ given by $P_{W,\lambda}(\alpha) = \lambda + \sum_{k=1}^d \alpha_k \phi_k$, where $(\phi_k)_{k=1}^d$ is a basis for a finite-dimensional space $W$ where $\phi_i > 0$ for all $i \in [d]$ and $\alpha \in [a,b]^d$ for $0 < a < b$ is an isomorphism onto its range. Let $(Q_N)_{N\in\mathbb{N}}$ be a sequence of operators converging strongly to the identity operator as in the beginning of Section 4.3.

Let $W_0 \coloneqq W \oplus \lambda$, then, according to Theorem 4.1.1 there is a $D \in \mathbb{N}$ and $C > 0$ such that

$$\|a_1 - a_2\|_\infty \leq C\|Q_D(F(a_1)) - Q_D(F(a_2))\|_\infty,$$

for all $a_1, a_2$ in an arbitrary compact and convex $K \subset W_0 \cap H_\lambda$. Notice that this result implies that, for an isomorphism $T_D$ such that $T_D \circ Q_D$ is a point sampling operator, we have that $T_D \circ Q_D \circ F$ has a Lipschitz continuous inverse.

We conclude that the operator $\widetilde{F} = T_D \circ Q_D \circ F \circ P_{W,\lambda}$ is an invertible and bi-Lipschitz function. Additionally, it is clear by construction that for every $x \in [0,1]$

$$\mathbb{R}^d \ni \alpha \mapsto \int_0^x \frac{\int_0^z f(s)ds}{(\lambda + \sum_{k=1}^d \alpha_k \phi_k(z))} dz$$

is smooth. This implies that $\widetilde{F}$ is smooth as well. Notice that $\widetilde{F}$ returns a $D$-dimensional vector which corresponds to the values of the functions at the sampling points. The details behind this approach can be found on [4, Page 7]. We conclude with the inverse function theorem that $\mathcal{M} = \widetilde{F}((a,b)^d)$ is a relatively compact manifold.

### 4.3.2 Identification of the Coefficient in the Euler-Bernoulli Equation

A beam is a long rigid element of a complex structure which is subject to a usually perpendicular external force causing the beam to bend. For simplicity, we think of a beam as the set of points $B = [0,1] \times \{0\}$. The deflection of the point $(x,0)$ is the measure of how much it is moved to a new position $(x',y') \in \mathbb{R}^2$. The model where deflections occur only in the $y$-coordinate under the action of a perpendicular force acting on $(x,0)$ is called the *Euler-Bernoulli* model. Such deflections also depend on a material property called *flexural rigidity* denoted by $a$. By simplicity, we identify the point $(x,0)$ with $x \in [0,1]$, the $y$-coordinate of the deflection acting on $x$ as $u(x) = u(x,0)$, and the $y$-coordinate of the force acting on $x$ as $f(x)$. We call $u : [0,1] \to \mathbb{R}$ the *deflection function* and $f : [0,1] \to \mathbb{R}$ the *force*. If $u \in C^4([0,1])$, $a \in C([0,1])$ and $f \in C([0,1])$, then, under some additional conditions that are described in [97], the deflection $u(x)$ of the beam at the point $x$ is modeled by the differential equation

$$\frac{d^2}{dx^2}\left(a(x)u''(x)\right) = f(x) \quad \text{for } x \in [0,1],$$

which is called the *Euler–Bernoulli* equation. There is a direct problem associated to this phenomenon consisting of identifying the deflection $u(x)$ at every point $x \in [0,1]$ when $f$ and $a$ are given and the following boundary conditions hold

$$\begin{aligned}(au'')'(0) &= c_3,\\ a(0)u''(0) &= c_2,\\ u'(0) &= c_1,\\ u(0) &= c_0,\end{aligned}$$

where $c_0$, $c_1$, $c_2$ $c_3 \in \mathbb{R}$. There are two inverse problems associated to this forward problem. First, the problem of reconstructing the coefficient $a$ leading to the so-called *coefficient identification problem*. This problem is particularly relevant when studying structures eroded by the action of natural agents such as storms and earthquakes and if it is impossible to dismantle the whole structure. Second, we have the (inverse) problem of reconstruction of the source $f$. In this work, we focus only on the first of the two inverse problems.

We remark that for the coefficient identification problem as described above the solution can be exactly reconstructed by integrating the differential equation. Concretely, if $u''(x) \neq 0$ for all $x \in [0,1]$ an explicit formula for the reconstruction of the coefficient $a$ at the point $x \in [0,1]$ is given by

$$a(x) = \frac{\int_0^x \int_0^t f(s)dsdt + c_3 x + c_2}{u''(x)}. \tag{4.3.3}$$

However, when dealing with real-life problems, the exact value of $u$ is unknown and instead, all that is provided is an approximation with noise. For our method to be applicable, we assume that $a$ belongs to a finite-dimensional space. Straightforward calculations lead to the forward operator $F : H_\lambda \subset C([0,1]) \to C([0,1])$

$$F(a) \coloneqq \left( x \mapsto \int_0^x \int_0^y \frac{\left(\int_0^s \int_0^w f(z)dzdw + c_3 s + c_2\right)}{a(s)} dsdy + c_1 x + c_0 \right), \tag{4.3.4}$$

where $H_\lambda \coloneqq \{a \in C([0,1]) : a > \lambda > 0\}$ for a fixed $\lambda$ is an open set. The uniqueness of $a$ for $u \in \mathrm{Im}(F)$ such that $u'' > 0$ and for initial conditions $c_0 = c_1 = c_2 = c_3 = 0$ is guaranteed by the explicit formula (4.3.3), i.e., $F$ is injective. Similar calculations to the example of Subsection 4.3.1 can be carried out to obtain the derivative of $F$ as

$$F'(a)(\delta a) = -\int_0^x \int_0^y \frac{\delta a(s)\left(\int_0^s \int_0^w f(z)dzdw\right)}{a(s)^2} dsdy, \text{ for } \delta a \in C([0,1]).$$

Moreover, by similar arguments as in the previous subsection, it can be shown that $F'(a)$ is injective for a fixed $a$ and continuous if $f > 0$.

We define again $P_{W,\lambda} : \mathbb{R}^d \to W$ given by $P_{W,\lambda}(\alpha) = \lambda + \sum_{k=1}^d \alpha_k \phi_k$, where $(\phi_k)_{k=1}^d$ is a basis for a subspace $W$, $\phi_i > 0$ for all $i \in [d]$, and $\alpha \in [a,b]^d$ for $0 < a < b$. Let $(Q_N)_{N\in\mathbb{N}}$ be

a sequence of operators converging strongly to the identity operator as in the beginning of Section 4.3. Let $W_0 \coloneqq W \oplus \lambda$. Then, Theorem 4.1.1 can be applied again. This shows that there exists a $D \in \mathbb{N}$ and $C > 0$ such that

$$\|a_1 - a_2\|_\infty \leq C\|Q_D(F(a_1)) - Q_D(F(a_2))\|_\infty,$$

for all $a_1, a_2$ in an arbitrary compact and convex $K \subset W_0 \cap H_\lambda$. We conclude that $Q_D \circ F|_K$ is bi-Lipschitz and injective. In addition, for an isomorphism $T_D$ such that $T_D \circ Q_D$ is a point sampling operator, we have that $T_D \circ Q_D \circ F$ has a Lipschitz continuous inverse.

Hence, the operator $\widetilde{F} = T_D \circ Q_D \circ F \circ P_{W,\lambda}$ is an invertible, bi-Lipschitz, and smooth function. As previously discussed, $\mathcal{M} = \widetilde{F}((a,b)^d)$ is a relatively compact manifold.

### 4.3.3 Solution of a Nonlinear Volterra-Hammerstein Integral Equation

Volterra-Hammerstein equations have been widely studied in the literature [103, 136, 144]. Let $F : L^2([0,1]) \to L^2([0,1])$ be the following (forward) operator

$$u \mapsto F(u) \coloneqq \left( t \mapsto \int_0^t (u(s))^2 ds \right), \tag{4.3.5}$$

for $u \in L^2([0,1])$ and $t \in [0,1]$. The direct problem is to determine the function $F(u) \in L^\infty([0,1]) \subset L^2([0,1])$ for a given $u \in L^2([0,1])$. Let us notice that $F(u)$ always exists for $u \in L^2([0,1])$. Let us consider the set $H_\lambda = \{u \in L^2([0,1]) : u > \lambda > 0\}$ for fixed $\lambda$. Notice that $F(H_\lambda) \subset C^1([0,1]) \cap \{v \in C^1([0,1]) : v' > 0\} \subset L^2([0,1])$. A solution to the inverse problem is found via the fundamental theorem of calculus. For every $v \in C^1([0,1])$, such that $v' > 0$ the equation $F(u) = v$ has two continuous solutions $u_+ = \sqrt{v'}$ and $u_- = -\sqrt{v'}$. As $u_- \notin H_\lambda$, we conclude that the operator $F|_{H_\lambda}$ is injective and this inverse problem has a unique solution. It is a well-known result that integral linear operators on infinite-dimensional spaces are compact operators and therefore, their inverses are not bounded. Hence, the inverse operator of (4.3.5) is not continuous. To overcome the instability of the inverse problem, we use Corollary 1 in [4]. By similar calculations to the previous examples, the first Fréchet derivative at $u$ of the forward operator denoted by $F'_u : L^2([0,1]) \to L^2([0,1])$ is the linear operator

$$\delta u \mapsto F'_u(\delta u) = \left( t \mapsto 2 \int_0^t u(s)\delta u(s) ds \right),$$

for all $\delta u \in L^2([0,1])$. Notice that under the assumption $u(s) > 0$, we have that $F'_u(\delta u) = 0$ if and only if $\delta u = 0$. Indeed, we have that if $F'_u(\delta u) = F'_u(\delta u')$ for $\delta u, \delta u' \in L^2([0,1])$, then

$$\int_0^x u(s)(\delta u(s) - \delta u'(s)) ds = 0,$$

by the Lebesgue differentiation theorem we have that for almost every $t \in [0,1]$

$$\begin{aligned} u(t)(\delta u(t) - \delta u'(t)) &= \lim_{h\to 0} h^{-1} \int_{t-h}^{t+h} u(s)(\delta u(s) - \delta u'(s))ds \\ &= \lim_{h\to 0} h^{-1} \cdot \left( \int_0^{t+h} u(s)(\delta u(s) - \delta u'(s))ds - \int_0^{t-h} u(s)(\delta u(s) - \delta u'(s))ds \right) \\ &= 0, \end{aligned}$$

which implies $\delta u = \delta u'$ and therefore $F'_u$ is injective.

Since the operator $F$ is Fréchet differentiable, injective, and $F'_u$ is injective for all $u \in C([0,1])$, Theorem 4.1.1 can be applied as in the previous examples. The discussion now follows that of Subsection 4.3.1.

We define again $P_{W,\lambda} : \mathbb{R}^d \to W + \lambda$ given by $P_{W,\lambda}(\alpha) = \sum_{k=1}^d \alpha_k \phi_k + \lambda$, such that $\lambda > 0$ and $(\phi_k)_{k=1}^d$ is a basis of a subspace $W$ where $\phi_k > 0$ for $k \in [d]$, $\alpha \in [a,b]^d$ for $0 < a < b$. Let $W_0 = W \oplus \lambda$, then Theorem 4.1.1 yields that for any sequence $(Q_N)_{N\in\mathbb{N}}$ converging strongly to the identity operator as $N \to \infty$, there is $D \in \mathbb{N}$ and $C > 0$ such that

$$\|u_1 - u_2\|_2 \leq C\|Q_D(F(u_1)) - Q_D(F(u_2))\|_2,$$

for all $u_1, u_2$ in a compact and convex $K \subset W_0 \cap H_\lambda$. Following a similar discussion as in Section 4.3.1, it follows that the operator $\widetilde{F} = T_D \circ Q_D \circ F|_K \circ P_W$ is an invertible, bi-Lipschitz, and smooth function between Euclidean spaces and $\mathcal{M} = \widetilde{F}((a,b)^d)$ is a relatively compact manifold.

### 4.3.4 Inverse Gravimetric Problem

An object with shape $\mathcal{E} \subset \mathbb{R}^2$ and mass density $\rho : \mathcal{E} \to \mathbb{R}$ admits a gravitational potential $U_{\mathcal{E},\rho} : \mathbb{R}^2 \setminus \mathcal{E} \to \mathbb{R}$ at the position $y \in \mathbb{R}^2$ given by

$$U_{\mathcal{E},\rho}(y) \coloneqq C \int_{\mathcal{E}} \rho(x) \ln(|x - y|)dx, \tag{4.3.6}$$

where $C$ is called the *gravitational constant* which is set to be 1 for simplicity (see [81]). The direct problem associated to this operator consists of determining the gravitational potential $U_{\mathcal{E},\rho}$ when $\rho$ and $\mathcal{E}$ are known. Two inverse problems are linked to this equation. The first problem is called the *linear inverse gravimetric problem*. Since it is physically impossible to measure the gravitational potential at every point, measurements are only taken at points on a surface $S$ away from $\bar{\mathcal{E}}$. We assume that the distance between an element $y \in S$ and $\mathcal{E}$ is $d(y,\mathcal{E}) > \delta$ for a fixed $\delta > 0$. For simplicity, we assume that $S \subset \mathbb{R}^2$ is a closed piecewise smooth surface which is the boundary of a solid body $B \subset \mathbb{R}^2$. The linear inverse gravimetric problem consist of finding only the mass density $\rho$ such that

$$U_{\mathcal{E},\rho}|_S = g,$$

when $U_{\mathcal{E},\rho}|_S$, $\mathcal{E}$ and $g \in L^2(S)$ are known. The second inverse gravimetric problem is also known as *the nonlinear inverse gravimetric problem* and is to find a shape $\mathcal{E}$ such that $\bar{\mathcal{E}} \subset \mathrm{int}(B)$ and

$$U_{\mathcal{E},\rho}|_S = g.$$

This work focuses on the linear inverse problem. We consider the operator

$$\begin{aligned} F\colon L^2(\mathcal{E}) &\to L^2(S), \\ \rho &\mapsto F(\rho) \coloneqq (y \mapsto U_{\mathcal{E},\rho}(y)). \end{aligned}$$

To demonstrate the smoothness of $F$ showing the continuity will suffice since $F$ is linear (in $\rho$). To prove the continuity of $F$, it will be shown that the operator is compact, which means that for every bounded family $\mathcal{F}$ of functions in $L^2(\mathcal{E})$, the family of functions $F(\mathcal{F})$ is relatively compact in $L^2(S)$. This will be established by invoking the theorem of Arzela-Ascoli (see [107]): every non-empty family of functions $\mathcal{G}$ where the elements are pointwise bounded and $\mathcal{G}$ is equicontinuous is relatively compact. First, the equicontinuity condition for $F(\mathcal{F})$ will be analyzed. Given $y, z \in S$ and $\rho \in \mathcal{F}$ it follows

$$|F(\rho)(y) - F(\rho)(z)| \leq C \int_{\mathcal{E}} |\rho(x)| \, |\ln|y-x| - \ln|z-x|| \, dx$$

Since $f(t) = \ln(t)$ is Lipschitz continuous in every interval $[\delta, \infty)$ with constant $C' \leq 1/\delta$, for $t < s$ it holds that

$$|\ln t - \ln s| \leq C'|t-s|.$$

Choosing $\delta > 0$ such that $d(y, \mathcal{E}) > \delta$ for all $y \in S$, and setting $t = |y-x|$ and $s = |z-x|$, we have by the inverse triangle inequality that

$$|\ln|y-x| - \ln|z-x|| \leq C' \, ||y-x| - |z-x|| \leq C'|y-z|,$$

and therefore

$$\begin{aligned} |F(\rho)(y) - F(\rho)(z)| &\leq C \int_{\mathcal{E}} |\rho(x)| \, |\ln|y-x| - \ln|z-x|| \, dx \\ &\leq C'C|z-y| \int_{\mathcal{E}} |\rho(x)| dx \\ &\leq \hat{C}\|\rho\|_{L^1(\mathcal{E})}|z-y| \\ &\leq \hat{C}C^*\|\rho\|_{L^2(\mathcal{E})}|z-y| \\ &\leq \widetilde{C}\|\rho\|_{L^2(\mathcal{E})}|z-y|, \end{aligned}$$

where $\hat{C} \coloneqq C'C$, we used that $\|\rho\|_{L^1(\mathcal{E})} \leq C^*\|\rho\|_{L^2(\mathcal{E})}$ for a universal constant $C^* > 0$ since $\mathcal{E}$ is bounded, and $\widetilde{C} \coloneqq \hat{C}C^*$. Since $\mathcal{F}$ was a bounded subset of $L^2(\mathcal{E})$, we conclude that $F(\mathcal{F})$ is equicontinuous.

Now, the point-wise boundedness of $F(\mathcal{F})$ will be derived. For $y \in \mathcal{E}$, the function $f_y(x) = |\ln|x-y||$ is continuous on the compact set $S$. Then, for $y \in \mathcal{E}$, the function $f_y(x) = |\ln(|y-x|)|$ is bounded with $f_y(x) \leq M_y$ where $M_y > 0$ is a constant that depends on $y$. Hence,

$$\begin{aligned} |F(\rho)(y)| &\leq C \int_{\mathcal{E}} |\rho(x)||\ln|y-x||dx \\ &\leq CM_y \int_{\mathcal{E}} |\rho(x)|dx \\ &= CC^* M_y \|\rho\|_{L^1(\mathcal{E})} \\ &\leq CM_y C^* \|\rho\|_{L^2(\mathcal{E})} \\ &\leq \widetilde{C} M_y, \end{aligned}$$

where $\widetilde{C} = C^*C$. Hence, $F(\mathcal{F})$ is pointwise bounded and therefore relatively compact. We conclude that the inverse problem associated to this operator is ill-posed.

Next, we introduce the operator $P_{W,\lambda} : (a,b)^d \subset \mathbb{R}^d \to W$ given by $P_{W,\lambda}(\alpha) = \sum_{k=1}^d \alpha_k \phi_k$, where $(\phi_k)_{k=1}^d$ is a basis for a finite-dimensional space $W$ where $\phi_i > 0$ for all $i \in [d]$ and $\alpha \in [a,b]^d$ for $0 < a < b$ is an isomorphism onto its range. It follows that the conditions of Theorem 4.1.1 are satisfied. For any sequence $(Q_N)_{N\in\mathbb{N}}$ of finite-rank operators, where $Q_N : L^2(S) \to L^2(S)$ and $Q_N \to I$ strongly as $N \to \infty$, there is $D \in \mathbb{N}$ and $C > 0$ such that

$$\|\rho_1 - \rho_2\|_2 \leq C\|Q_D(F(\rho_1)) - Q_D(F(\rho_2))\|_2,$$

for all $\rho_1, \rho_2$ in an arbitrary compact and convex $K$ subset of $W$. We can see that the operator $\widetilde{F} = T_D \circ Q_D \circ F|_K \circ P_W$ is an invertible, bi-Lipschitz, and smooth function between Euclidean spaces and $\mathcal{M} = \widetilde{F}((a,b)^d)$ is a relatively compact manifold.

## 4.4 Numerical Experiments

For all inverse problems described in Section 4.3, we present numerical examples illustrating how robust approximations to the inverse operators can be learned by neural networks. We will proceed similarly for all cases: First, given the forward operator $F : \mathcal{X} \to \mathcal{Y}$ between Banach spaces, a basis $B$ that defines a finite-dimensional space $W$ is chosen and the domain of $F$ will be restricted to this subspace. Second, as it was shown in Section 4.3 for the list of admissible inverse problems, given a compact and convex subset $K$ of $W$ and a sequence of finite-rank operators $(\widehat{Q}_N)_{N\in\mathbb{N}}$ where $\widehat{Q}_N : \mathcal{Y} \to Y_N$ and $Y_N$ are linear subspaces of $Y$, there is a $D \in \mathbb{N}$ such that

$$\widehat{F} \coloneqq \widehat{Q}_D \circ F|_K$$

has a Lipschitz continuous inverse. Third, we identify the space $Y_D$ with a Euclidean space $\mathbb{R}^D$ via an isomorphism $T_D : Y_D \to \mathbb{R}^D$ and $W$ with $\mathbb{R}^d$ through the isomorphism $P : \mathbb{R}^d \to W$. Then, the function

$$\widetilde{F} \coloneqq T_D \circ \widehat{F} \circ P|_{P^{-1}(\Gamma)}$$

is Lipschitz continuous. Next, $\widetilde{F}^{-1} : \mathbb{R}^D \supset \mathcal{M} \to \mathbb{R}^d$, where $\mathcal{M} \coloneqq \widetilde{F}(P^{-1}(\Gamma))$ and $\Gamma \subset K$, will be approximated by realizations of NNs for different values of $D$ since in practice, we do not know the exact value of $D$. In addition, we will suppose that we know the operator $T_D \circ \widehat{Q}_D$ for every $D \in \mathbb{N}$ and we will denote it as $Q_D$. For the training of the NN, and for fixed ambient dimension $D$, intrinsic dimension $d$, and noise level $\delta$, a data set of 40.000 noisy data points $(\mathbf{y}_i, x_i)_{i=1}^{40.000}$ is generated. To analyze the practical visibility of the bounds identified in Theorems 4.2.5 and 4.2.10 every component is perturbed with normally-distributed random noise $\eta = (\eta_i)_{i=1}^{40.000}$. Then $(\mathbf{y}_i, \mathbf{x}_i)_{i=1}^{40.000} = (\widetilde{F}(\mathbf{x}_i) + \eta_i, \mathbf{x}_i)_{i=1}^{40.000}$. This training set was identified in Theorem 4.2.10 as being the right one to generate robust-to-noise NNs. When the training stage is completed, each NN is evaluated on a test set of 8.000 randomly generated data points $(\mathbf{y}_i, \mathbf{x}_i)_{i=1}^{8.000}$ with the same distribution as the training data.

The predictions $\tilde{x}_i$ returned by the NN for each $\mathbf{y}_i$ will be compared with the elements $\mathbf{x}_i$. For each inverse problem and fixed $D$, $d$, and $\delta$, the training and testing process are performed 10 times and the mean squared test errors are presented. We recall that the mean squared test error is

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \tilde{\mathbf{x}}_i)^2.$$

Notice that the expected squared norm of the added noise, scales linearly in $D$ and linear in $\mathbb{E}(|\eta|^2)$. We observe that the parameter $D$ does not negatively affect the test error, even though the norm of the noise grows. This mirrors the prediction of Theorem 4.2.5, especially Equation (4.2.4). Moreover, the behavior of the error in all examples also reflects the observation in Remark 4.2.6.

### 4.4.1 Identification of the Transmissivity Coefficient

In this subsection, two examples will be presented. They differ only in the ambient dimension $D$. Given the forward operator $F : C([0,1]) \supset H_\lambda \to C([0,1])$ of (4.3.1), we set $f = 1$, $c_0, c_1 = 0$. The domain of $F$ will be restricted to the subspace $W_0$ of $C([0,1])$ given as the span of the four inner Lagrangian finite elements $\phi_1, \ldots, \phi_4$ on the grid $\{0, 1/6, 2/6, 3/6, 4/6, 5/6, 1\}$ and the constant 1. For $W \coloneqq \operatorname{span}\{\phi_1, \ldots, \phi_4\}$, $\lambda \coloneqq 0.1$, let $P_{W,\lambda}$ be defined as in Subsection 4.3.1. Additionally, $(a, b) = (0, 1)$ is chosen. Finally, let $Q_D$ be the sampling operator that samples continuous functions on $D$ equidistant points in $[0, 1]$. The inverse of the function

$$\widetilde{F} \coloneqq Q_D \circ F \circ P_{W,\lambda} : (0,1)^4 \to \mathbb{R}^D$$

will be approximated by the realization of a four-layer NN with 100 neurons per layer on 40.000 perturbed data points $(\mathbf{y}_i, x_i)_{i=1}^{40.000}$. Notice that $\mathcal{M} = \widetilde{F}((0,1)^4)$ is a $(M, \delta)-$covered submanifold and Theorem 4.2.5 can be applied. The mean squared error for different values of $D$ and noise level $\delta$ is displayed in Figure 4.3.

Next, a second example is discussed where the intrinsic dimension will be altered while the ambient dimension will remain unchanged. Under the same initial condition $c_0 = c_1 = 0$, $f = 1$, the operators $P_{W,\lambda}$ and $Q_D$ as previously stated with fixed $D = 100$ and the forward

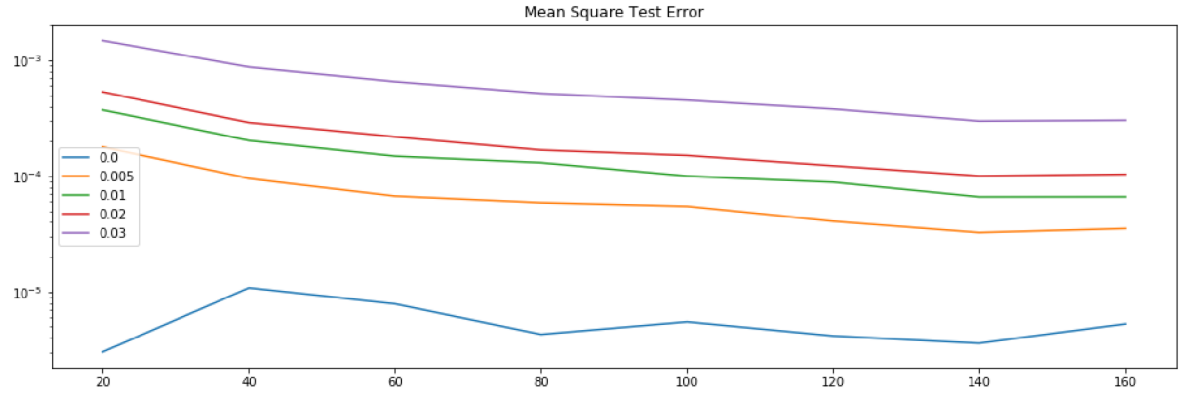


Figure 4.3: Mean squared errors of the approximation of the solution operator for the transmissivity identification problem on the test set for five different noise levels: $0.00, 0.005, 0.01, 0.02, 0.03$. The $x$-axis shows the ambient dimension which corresponds to the number of measurements $D$, the $y$-axis records the mean squared error of the reconstruction over the test set.

operator (4.3.1), the discretized operator

$$\widetilde{F} \coloneqq Q_{100} \circ F \circ P_{W,\lambda} : (0,1)^d \to \mathbb{R}^{100},$$

will be approximated by the realization of an NN for different noise levels and the values $d = 4, 8, 12, 16, 20$. In this case, the $d$-dimensional space $W$ is generated by the inner Lagrangian finite elements $(\phi_i)_{i=1}^d$ on the mesh $\{x_0, x_1, \ldots, x_{d+1}\}$ where $x_i = i/(d+1)$ and $i \in \{0, \ldots, d+1\}$ for $d = 4, 8, 12, 16, 20$. In Figure 4.4. As stated in Equation (4.2.4), the error grows as the intrinsic dimension increases.

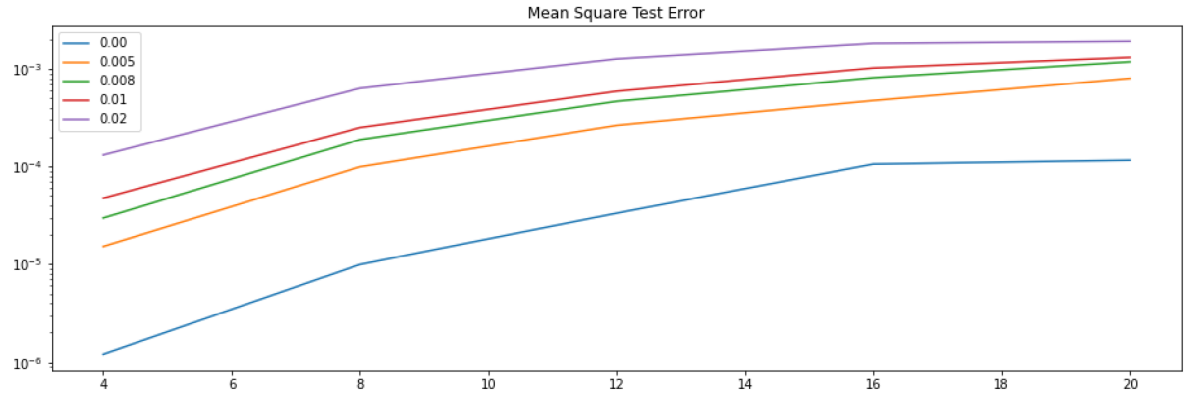


Figure 4.4: Mean squared errors of the approximation of the solution operator for the transmissivity identification problem on the test set for five different noise levels: $0.00, 0.005, 0.008, 0.01, 0.02$. The $x$-axis shows the intrinsic dimension $d$, the $y$-axis records the mean squared errors of the reconstruction over the test set.

### 4.4.2 Identification of the Coefficient in the Euler-Bernoulli Equation

In this subsection, an example of approximation of the inverse operator to the forward operator $F$ of (4.3.4) with $c_0, c_1, c_2, c_3 = 0$ and $f = 1$ will be discussed. We assume that the coefficient $a$ belongs to the finite-dimensional space $W$ defined as the span of the functions $(a_k)_{k=1}^4$, where

$$a_k(x) = \frac{\cos(2\pi k x)}{10} + 0.11 \text{ for } x \in [0,1].$$

Now, we define $P_W$ as in Subsection 4.3.1, $(a,b) = (0,1)$ and the sampling operator $Q_D$ that samples continuous functions on $D$ equidistant points in $[0,1]$. Then, the inverse of the function

$$\widetilde{F} := Q_D \circ F \circ P_W : (0,1)^4 \to \mathbb{R}^D$$

is approximated by the realization of an NN. Notice that $\mathcal{M} = \widetilde{F}((0,1)^4)$ is a $(M,\delta)-$covered submanifold and Theorem 4.2.5 can be applied. Again, a four-layer NN with 100 neurons per layer is trained on a set of 40.000 perturbed data points $(\mathbf{y}_i, x_i)_{i=1}^{40.000}$. In Figure 4.5, we present the mean squared error for various values of $D$ and noise levels $\delta$.


Figure 4.5: Mean squared errors of the approximation to the inverse problem of the Euler-Bernoulli equation on the test set for five different noise levels: $0.00, 0.005, 0.008, 0.01, 0.02$. The $x$-axis shows the ambient dimension which corresponds to the number of measurements $D$, the $y$-axis records the mean squared error of the reconstruction over the test set.

### 4.4.3 Solution of a Nonlinear Volterra-Hammerstein Integral Equation

For the approximation of the inverse of the nonlinear operator (4.3.5), the unknown function $u$ is assumed to belong to the finite-dimensional vector space $W$ generated by the basis

functions $(u_k)_{k=1}^d$ such that

$$u_k(x) = \frac{1}{4}\cos\left(2\pi k \cdot \left(x + \frac{1}{4}\right)\right) + 1, \text{ for } x \in [0,1].$$

Once again, we set $(a,b) = (0,1)$, $P_{W,\lambda}$ as in Section 4.3.3 along with $Q_D$ the sampling operator on $D$ equidistant points in $[0,1]$. Then, the inverse of the operator

$$\widetilde{F} := Q_D \circ F \circ P_{W,\lambda} : (0,1)^d \to \mathbb{R}^D$$

will be approximated by the realization of a four-layer NN with 100 neurons per layer. Notice that $\mathcal{M} = \widetilde{F}((0,1)^d)$ is a $(M,\delta)-$covered submanifold and Theorem 4.2.5 can be applied. For different values of $\delta$ and $D$ and for $d = 4$, the mean squared errors are visualized in Figure 4.6. In addition, in Figure 4.7 the mean squared errors for $d = 8$ are depicted. As previously stated, the increase in the intrinsic dimension $d$ yields a substantial increment in the error.

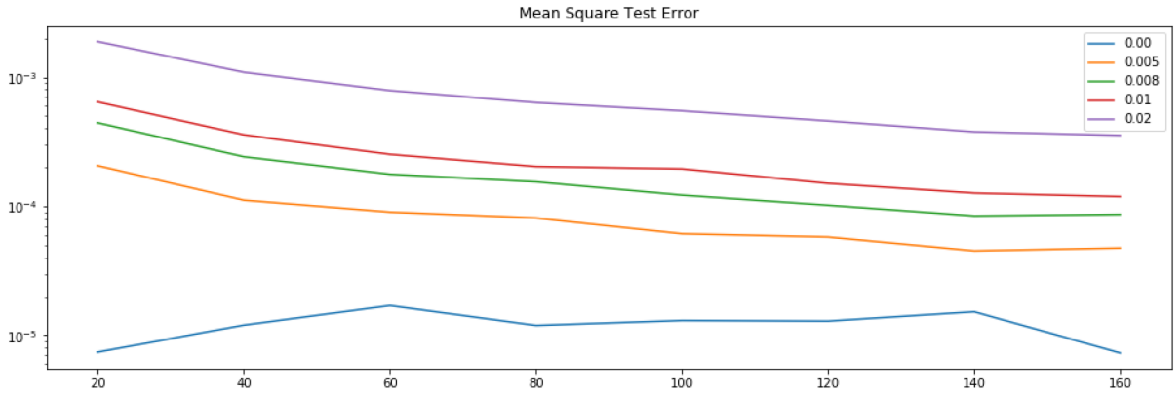


Figure 4.6: Mean squared error of the approximation to the inverse problem of the Volterra-Hammerstein integral equation on the test set for five different noise levels: $0.00, 0.005, 0.008, 0.01, 0.02$ and intrinsic dimension $d = 4$. The $x$-axis shows the ambient dimension which corresponds to the number of measurements $D$, the $y$-axis records the mean squared error of the reconstruction over the test set.

### 4.4.4 Solution of the Linear Inverse Gravimetric Problem

For the solution of the linear inverse gravimetric problem with forward operator $F$ given by (4.3.6), the density $\rho$ is assumed to belong to the four-dimensional subspace of $L^2([-1,1]^2)$ generated by the indicator functions:

$$\begin{aligned}
\phi_1(x) &= \mathbb{1}_{[-0.5,0]\times[0,0.5]}(x) \\
\phi_2(x) &= \mathbb{1}_{[0,0.5]\times[0,0.5]}(x) \\
\phi_3(x) &= \mathbb{1}_{[-0.5,0]\times[-0.5,0]}(x) \\
\phi_4(x) &= \mathbb{1}_{[0,0.5]\times[-0.5,0]}(x)
\end{aligned}$$

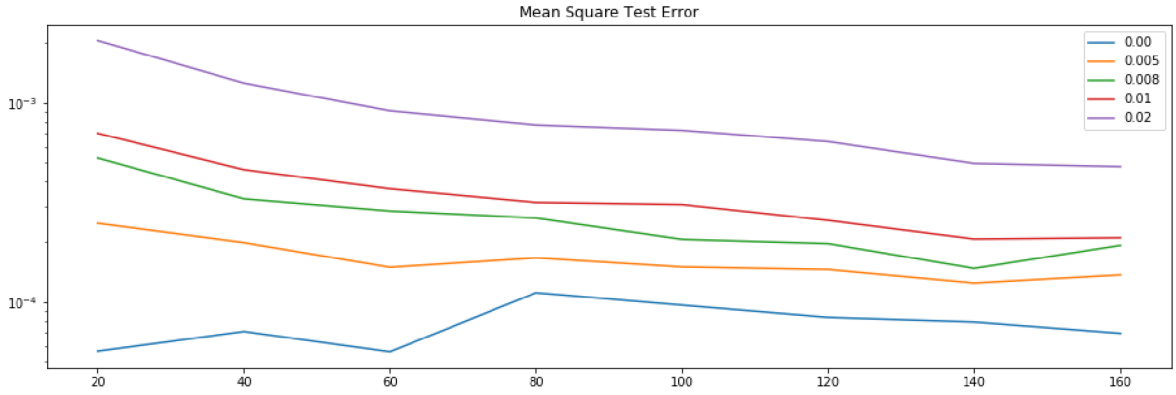


Figure 4.7: Mean squared errors of the approximation to the inverse problem of the Volterra-Hammerstein integral equation on the test set for five different noise levels: $0.00, 0.005, 0.008, 0.01, 0.02$ and intrinsic dimension $d = 8$. The $x$-axis shows the ambient dimension which corresponds to the number of measurements $D$, the $y$-axis records the mean squared error of the reconstruction over the test set.

where $x \in [-1,1]^2$. Again, let $P_W : (0,1)^4 \to L^2([-1,1]^2)$ be the operator given by $P_W(\alpha) = \sum_{k=1}^4 \alpha_k \phi_k$ with $\alpha = (\alpha_k)_{k=1}^4 \in \mathbb{R}^4$. To determine $\mathbf{y}_i$, we measure the values of $F(\rho)$ at the following points on the boundary of $[-1,1]^2$: the first sample is taken at the point $(-1,1) \in [-1,1]^2$ and the successive samples are taken counterclockwise at points with distance $1/D$. We can see that the number of samples is $4D$. Again, an NN is trained to approximate the inverse of the operator

$$\widetilde{F} \coloneqq Q_{4D} \circ F \circ P_W : (0,1)^4 \to \mathbb{R}^{4D}.$$

Figure 4.8 depicts the mean squared errors for the solution of the problem.

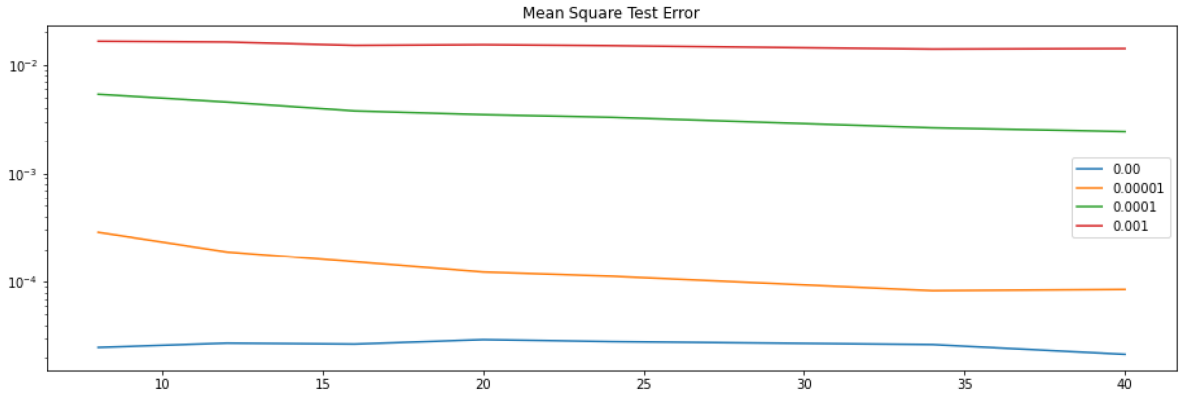


Figure 4.8: Mean squared errors of the approximation to the inverse problem of the gravimetric problem on the test set for four different noise levels: $0.00, 0.00001, 0.0001, 0.001$ and intrinsic dimension $d = 4$. The $x$-axis the value $D$ such that the ambient dimension is $4D$, the $y$-axis records the mean squared errors of the reconstruction over the test set.

# Chapter 5

# A PET for the Approximation of Frequency-localized Functions

Our last chapter is dedicated to the study of data-driven methods for the reconstruction of frequency-localized functions. To define the notion of *frequency-localized functions*, we recall the *Fourier transform*. Given a function $f : \mathbb{R}^d \to \mathbb{C}$, $f \in L^1(\mathbb{R}^d)$, its Fourier transform is the function

$$\hat{f}(\mathbf{v}) := \int_{\mathbb{R}^d} f(\mathbf{x}) e^{-i\mathbf{v}\cdot\mathbf{x}} \, \mathrm{d}\mathbf{x}, \qquad \mathbf{v} \in \mathbb{R}^d,$$

with inverse transform

$$f^{\vee}(\mathbf{x}) := \int_{\mathbb{R}^d} \hat{f}(\mathbf{v}) e^{i\mathbf{v}\cdot\mathbf{x}} \, \mathrm{d}\mathbf{x}, \qquad \mathbf{v} \in \mathbb{R}^d.$$

The domain of $\hat{f}$ is usually referred to as the *frequency domain*. The Fourier transform operator plays an important role in many applications and can be extended to $L^2(\mathbb{R}^d)$ via the Plancherel identity [126].

Before we proceed to extend the concept of Fourier transform, we require the following proposition.

**Proposition 5.0.1.** *[24, Proposition 4.17] Let $d \in \mathbb{N}$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function and let $(A_i)_{i\in I}$ be the family of all open sets in $\mathbb{R}^d$ such that for each $i \in I$, $f = 0$ a.e. on $A_i$. Let $A = \bigcup_{i\in I} A_i$. Then, $f = 0$ a.e. on $A$. The support of $f$ denoted as $supp(f)$ is the complement of $A$.*

If the energy of the function $\hat{f}$ is mostly contained in a compact set, we say that is frequency-localized. The Gaussian function is one of the most renowned frequency-localized functions. Another example of frequency-localized functions is the set of *bandlimited* [159] functions introduced in Definition 5.0.2.

**Definition 5.0.2.** *Let $\mathsf{w} > 0$. A function $f \in L^2(\mathbb{R}^d)$ is said to be $\mathsf{w}$-bandlimited if $supp(\hat{f}) \subset [-\mathsf{w}, \mathsf{w}]^d$. Equivalently,*

$$f(\mathbf{x}) = \frac{1}{(2\pi)^d} \int_{[-\mathsf{w},\mathsf{w}]^d} \hat{f}(\mathbf{v}) e^{i\mathbf{v}\cdot\mathbf{x}} \, \mathrm{d}\mathbf{v} \text{ for } \mathbf{x} \in \mathbb{R}^d. \tag{5.0.1}$$

When $d = 1$, the space of $\mathsf{w}$-bandlimited functions is referred to as the *Paley–Wiener space* [174, Subsection 2.2] of functions, denoted

$$\mathrm{PW}_{\mathsf{w}} := \{f \in L^2(\mathbb{R}) : \mathrm{supp}(\hat{f}) \subset [-\mathsf{w}, \mathsf{w}]\}.$$

The uncertainty principle asserts that for all nontrivial $f \in L^2(\mathbb{R})$, at least one of the sets $\{\mathbf{x} : f(\mathbf{x}) \neq 0\}$ or $\{\mathbf{v} : \hat{f}(\mathbf{v}) \neq 0\}$ has infinite measure ([57, Chapter 2]). Thus, a bandlimited function cannot be simultaneously *timelimited*, i.e. it cannot be compactly supported in the physical space ($\mathbf{x} \in \mathbb{R}^d$). Nevertheless, a natural and compelling resolution to this "time-frequency problem" emerges through the identification of *essentially timelimited* functions, a concept from the celebrated "Bell Labs theory," developed by Slepian, Landau, and Pollack [89, 156, 87, 88, 153]. We proceed to review this theory, beginning with dimension one and concluding with its extension to the multidimensional space.

## 5.1 The Slepian Basis

We introduce the Slepian basis in $1D$ first and proceed to generalize it for an arbitrary dimension $d$.

**The Slepian Basis in Dimension One.** For $T, \mathsf{w} > 0$, we define the following restriction operators on $L^2(\mathbb{R})$, as

$$P_{\mathsf{w}} f = (\hat{f}\mathbb{1}_{[-\mathsf{w},\mathsf{w}]})^{\vee} \quad \text{and} \quad Q_T f = f\mathbb{1}_{[-T,T]}.$$

where $\mathbb{1}_A$ denotes the indicator function for a set $A \subset \mathbb{R}$. By $\chi_A$, we denote the indicator function of $A \subset \mathbb{R}$. It is easy to verify that the range of $P_{\mathsf{w}}$ is exactly the Paley-Wiener space: $P_{\mathsf{w}} L^2(\mathbb{R}) = \mathrm{PW}_{\mathsf{w}}$. Since no nontrivial $f \in L^2(\mathbb{R})$ is fixed by both $P_{\mathsf{w}}$ and $Q_T$ due to the uncertainty principle, an alternative approach is to measure how much of the energy of a function in $\mathrm{PW}_{\mathsf{w}}$ is concentrated on $[-T, T]$. To this end, we consider the operator

$$P_{\mathsf{w}} Q_T f(x) = \frac{1}{2\pi} \int_{-\mathsf{w}}^{\mathsf{w}} \int_{-T}^{T} f(t) e^{-itv} e^{ivx} \, \mathrm{d}t\mathrm{d}v.$$

The operator $P_{\mathsf{w}} Q_T$ is compact, since it is a Hilbert-schmidt operator, and is self-adjoint. A quick calculation shows that $P_{\mathsf{w}} Q_T f(x) = P_{T\mathsf{w}} Q_1 f(Tx)$ for all $f \in L^2(\mathbb{R})$. Hence, without loss of generality, we may set $T = 1$, in which case we define $\mathcal{Q}_{\mathsf{w}} := P_{\mathsf{w}} Q_1$.

In [155], it is proved that

$$\mathcal{Q}_{\mathsf{w}} f(x) = \int_{-1}^{1} f(t) \, \frac{\sin \mathsf{w}(x-t)}{\pi(x-t)} \, \mathrm{d}t, \qquad x \in \mathbb{R}. \tag{5.1.1}$$

The compactness of $P_{\mathsf{w}}$ and $Q_1$ ensures that the spectrum of $\mathcal{Q}_{\mathsf{w}}$ is countable and accumulates at 0. We denote these eigenvalues as

$$1 > \mu_{\mathsf{w},0} > \mu_{\mathsf{w},1} > \cdots > 0, \tag{5.1.2}$$

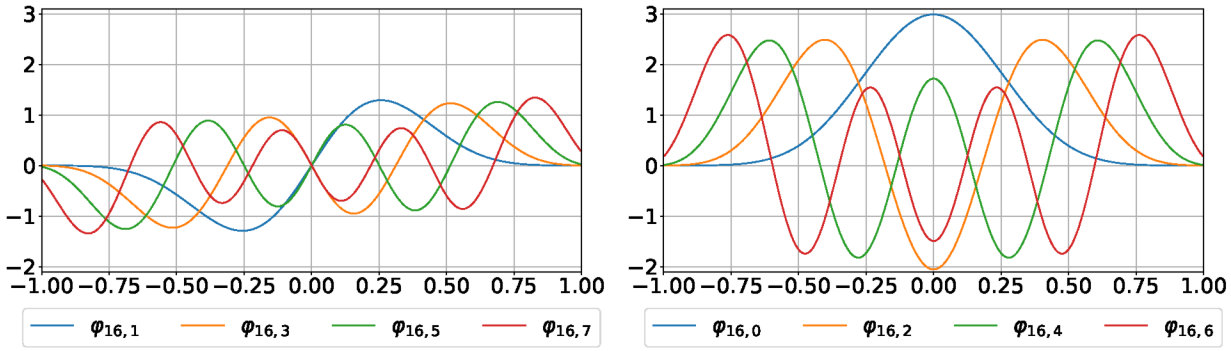


Figure 5.1: Plot of Slepian functions (PSWFs) $\varphi_{16,j}$ with $\mathsf{w} = 16$ for odd (left) and even (right) indices $j = 0, 1, 2, 3, 4, 5, 6, 7$.

where $\lim_{j\to\infty} \mu_{\mathsf{w},j} = 0$ [120, Subsection 3.2.1]. The eigenfunctions of $\mathcal{Q}_\mathsf{w}$ form an orthogonal basis for the space $L^2_{\mathbf{u}}([-1,1])$, which is the $L^2$-Lebesgue space on $[-1,1]$, with norm calibrated to the uniform measure $\mathbf{u}$ on $[-1,1]$. From this point on, we use the symbol $\mathbf{u}$ to refer to the uniform measure in $[-1,1]$. For $j \geq 0$, let $\varphi_{\mathsf{w},j} \in \mathrm{PW}_\mathsf{w} \cap L^2_{\mathbf{u}}([-1,1])$ to be the ($\mathbf{u}$-)*normalized* eigenfunction of $\mu_{\mathsf{w},j}$ [1], i.e. $\mathcal{Q}_\mathsf{w}\varphi_{\mathsf{w},j} = \mu_{\mathsf{w},j}\varphi_{\mathsf{w},j}$ and

$$\|\varphi_{\mathsf{w},j}\|^2_{L^2_{\mathbf{u}}[-1,1]} = \int_{-1}^{1} |\varphi_{\mathsf{w},j}(x)|^2 \, \mathrm{d}\mathbf{u}(x) = 1. \tag{5.1.3}$$

The set $\{\varphi_{\mathsf{w},j}\}_{j\in\mathbb{N}_0}$ is referred to in the literature as the Prolate Spheroidal Wave Functions or the Slepian basis. We illustrate the Slepian basis functions $\varphi_{\mathsf{w},j}$ for $\mathsf{w} = 16$ and $j = 0, 1 \ldots, 7$ in Figure 5.1.

An alternative way to arrive at the Slepian basis comes in the form of solutions to a Sturm-Louville equation. Precisely, consider the second-order boundary value problem

$$\mathcal{L}_\mathsf{w} f(x) := -\frac{d}{dx}\left((1-x^2)\frac{df}{dx}(x)\right) + \mathsf{w}^2 x^2 f(x), \qquad \forall x \in (-1,1). \tag{5.1.4}$$

The normalized eigenfunctions of this problem are exactly the PSWFs, $\mathcal{L}_\mathsf{w}\varphi_{\mathsf{w},j} = \chi_{\mathsf{w},j}\varphi_{\mathsf{w},j}$, but now with the eigenvalues satisfying the ordering

$$0 < \chi_{\mathsf{w},0} < \chi_{\mathsf{w},1} < \cdots \tag{5.1.5}$$

with $\lim_{j\to\infty} \chi_{\mathsf{w},j} = \infty$.

Understanding the pointwise magnitude of the Slepian basis functions is fundamental to our analysis of the least-squares approximation problem. We present two established lemmata.

**Lemma 5.1.1.** *Let $\mathcal{Q}_\mathsf{w}$ be defined in* (5.1.1)*. Then the eigenvalues $\mu_{\mathsf{w},j}$ of $\mathcal{Q}_\mathsf{w}$ satisfy*

$$\mu_{\mathsf{w},\lfloor 4\mathsf{w}\rfloor - 1} \geq \frac{1}{2} \geq \mu_{\mathsf{w},\lceil 4\mathsf{w}\rceil}. \tag{5.1.6}$$

[1] In this chapter, a function is referred to as normalized if its $L^2_{\mathbf{u}}$ over $[-1,1]$ equals to one.

Lemma 5.1.2 below is a version of an estimate commonly referenced in the literature but seldom proved. For the convenience of the reader, we offer a brief proof sketch, supplemented with references for more rigorous details.

**Lemma 5.1.2.** *Let $j \in \mathbb{N}_0$. Then for every $j \geq \frac{2\mathsf{w}}{\pi}$, the following holds*

$$\|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])} \leq \frac{12}{5}\sqrt{j+1}. \tag{5.1.7}$$

*Sketch of proof.* Recall that the Slepian basis functions $\varphi_{\mathsf{w},j}$ serve as eigenfunctions for the operator $\mathcal{L}_{\mathsf{w}}$ (5.1.4), with the corresponding eigenvalues denoted by $\chi_{\mathsf{w},j}$ (5.1.5). Let $j \geq \frac{2\mathsf{w}}{\pi}$. Then it follows from [120, Theorem 4.4] that

$$\chi_{\mathsf{w},j} > \mathsf{w}^2. \tag{5.1.8}$$

On the one hand, when (5.1.8) is satisfied, we can employ [120, Theorem 3.39] to obtain

$$\|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])} = |\varphi_{\mathsf{w},j}(1)|. \tag{5.1.9}$$

On the other hand, according to [120, Theorem 3.39], (5.1.8) also yields

$$|\varphi_{\mathsf{w},j}(1)| \leq \frac{12}{5}\sqrt{j+1}. \tag{5.1.10}$$

Combining (5.1.8), (5.1.9), (5.1.10), we acquire (5.1.7), as desired.

□

**Proposition 5.1.3.** *Let $\mathsf{w} \geq 1$. Let $j^\star(\mathsf{w}) := \lfloor 4\mathsf{w} \rfloor - 1$. Then it holds for all $j \in \mathbb{N}_0$ that*

$$\|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])} \leq \begin{cases} 2\sqrt{\frac{\mathsf{w}}{\pi}} & \text{if } j \leq j^\star(\mathsf{w}), \\ \\ \frac{12}{5}\sqrt{j+1} & \text{if } j > j^\star(\mathsf{w}). \end{cases}$$

*Proof of Proposition 5.1.3.* Extend the uniform measure $\mathbf{u}$ on $[-1,1]$ to a measure on $\mathbb{R}$ by setting $\mathrm{d}\mathbf{u}(t) = \frac{1}{2}\mathrm{d}t$. We show that $\{\varphi_{\mathsf{w},j}\}_{j\in\mathbb{N}_0}$ forms an orthogonal system in $L^2_{\mathbf{u}}(\mathbb{R})$. Since $\varphi_{\mathsf{w},j}$ is an eigenfunction of the operator $\mathcal{Q}_{\mathsf{w}}$ (5.1.1) corresponding to the eigenvalue $\mu_{\mathsf{w},j}$,

$$\mathcal{Q}_{\mathsf{w}}\varphi_{\mathsf{w},j}(x) = \int_{-1}^{1} \varphi_{\mathsf{w},j}(t)\,\frac{\sin \mathsf{w}(x-t)}{\pi(x-t)}\,\mathrm{d}t = \mu_{\mathsf{w},j}\varphi_{\mathsf{w},j}(x), \qquad \forall x \in \mathbb{R}. \tag{5.1.11}$$

Hence, we deduce that

$$\begin{aligned}
&\int_{\mathbb{R}} \varphi_{\mathsf{w},j}(x)\varphi_{\mathsf{w},k}(x)\,\mathrm{d}\mathbf{u}(x) \\
&= \frac{1}{2\mu_{\mathsf{w},j}\mu_{\mathsf{w},k}} \int_{[-1,1]\times[-1,1]} \varphi_{\mathsf{w},j}(t)\varphi_{\mathsf{w},k}(u)\left(\int_{\mathbb{R}} \frac{\sin \mathsf{w}(x-t)}{\pi(x-t)}\,\frac{\sin \mathsf{w}(x-u)}{\pi(x-u)}\,\mathrm{d}x\right)\mathrm{d}t\mathrm{d}u \\
&= \frac{1}{2\mu_{\mathsf{w},j}\mu_{\mathsf{w},k}} \int_{[-1,1]\times[-1,1]} \varphi_{\mathsf{w},j}(t)\varphi_{\mathsf{w},k}(u)\frac{\sin \mathsf{w}(t-u)}{\pi(t-u)}\,\mathrm{d}t\mathrm{d}u \\
&= \frac{1}{\mu_{\mathsf{w},j}} \int_{[-1,1]} \varphi_{\mathsf{w},j}(t)\varphi_{\mathsf{w},k}(t)\,\mathrm{d}\mathbf{u}(t).
\end{aligned} \tag{5.1.12}$$

Above, we have applied a *reproducing kernel trick* [18, Subsection 3.1.2] at the second inequality and (5.1.11) at the first and last. Due to the orthonormality of $\{\varphi_{\mathsf{w},j}\}_{j\in\mathbb{N}_0}$ in $L^2_{\mathbf{u}}([-1,1])$, (5.1.12) evaluates to zero if $j \neq k$ and to $\frac{1}{\mu_{\mathsf{w},j}}$ if $j = k$, we conclude that $\{\varphi_{\mathsf{w},j}\}_{j\in\mathbb{N}_0}$ indeed constitutes an orthogonal system in $L^2_{\mathbf{u}}(\mathbb{R})$[2]. In particular,

$$\|\varphi_{\mathsf{w},j}\|_{L^2_{\mathbf{u}}(\mathbb{R})} = \sqrt{\frac{1}{\mu_{\mathsf{w},j}}}. \tag{5.1.13}$$

Hence, by applying Lemma 5.1.1 to (5.1.13), we obtain for $j \leq \lfloor 4\mathsf{w} \rfloor - 1$ that

$$\|\varphi_{\mathsf{w},j}\|_{L^2_{\mathbf{u}}(\mathbb{R})} \leq \sqrt{2}. \tag{5.1.14}$$

Since $\varphi_{\mathsf{w},j} \in \mathrm{PW}_{\mathsf{w}}$, (5.0.1) holds. It follows that

$$|\varphi_{\mathsf{w},j}(x)| = \left| \frac{1}{2\pi} \int_{-\mathsf{w}}^{\mathsf{w}} \widehat{\varphi_{\mathsf{w},j}}(t) e^{ixt}\, dt \right| \leq \frac{\sqrt{\mathsf{w}}}{\sqrt{2}\pi} \|\widehat{\varphi_{\mathsf{w},j}}\|_{L^2(\mathbb{R})} = \sqrt{\frac{\mathsf{w}}{\pi}} \|\varphi_{\mathsf{w},j}\|_{L^2(\mathbb{R})}, \qquad \forall x \in \mathbb{R},$$

and subsequently, for $j \in \mathbb{N}_0$,

$$\|\varphi_{\mathsf{w},j}\|_{L^\infty(\mathbb{R})} \leq \sqrt{\frac{2\mathsf{w}}{\pi}} \|\varphi_{\mathsf{w},j}\|_{L^2_{\mathbf{u}}(\mathbb{R})}. \tag{5.1.15}$$

Then combining (5.1.14) with (5.1.15) for $j \leq \lfloor 4\mathsf{w} \rfloor - 1$ yields

$$\|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])} \leq \|\varphi_{\mathsf{w},j}\|_{L^\infty(\mathbb{R})} \leq \sqrt{\frac{2\mathsf{w}}{\pi}} \|\varphi_{\mathsf{w},j}\|_{L^2_{\mathbf{u}}(\mathbb{R})} \leq 2\sqrt{\frac{\mathsf{w}}{\pi}}.$$

For the remaining case of $j \geq \lfloor 4\mathsf{w} \rfloor$, we recall from Lemma 5.1.2 that (5.1.7) holds whenever $j \geq \frac{2\mathsf{w}}{\pi}$. The proof is now completed. □

Proposition 5.1.3 can be reformulated in the following result.

**Proposition 5.1.4.** *For any $\mathsf{w} \geq 1$, let $\{\varphi_{\mathsf{w},j}\}_{j\in\mathbb{N}_0}$ be the Slepian basis in $L^2_{\mathbf{u}}([-1,1])$. Then we have*

$$\|\varphi_{\mathsf{w},j}\|^2_{L^\infty([-1,1])} \leq \mathbb{1}_{\{j=0\}}\Big(\frac{4\mathsf{w}}{\pi}\Big) + \mathbb{1}_{\{j\in\mathbb{N}\}}(j+1)^{\gamma(\mathsf{w})}. \tag{5.1.16}$$

**Remark 5.1.5.** *The distinction between Proposition 5.1.3 and Proposition 5.1.4 lies in the latter providing a different upper bound for $\|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])}$ as a function of the index $j$, without relying on a critical division $j^\star(\mathsf{w})$ tied to the bandwidth $\mathsf{w}$. Such a broader bound is needed in the proof of Theorem 5.3.5. A frequently cited estimate indicates that, for all $j \in \mathbb{N}_0$*

$$\|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])} \leq C\sqrt{j}; \tag{5.1.17}$$

[2]In other words, the Slepian functions are bi-orthogonal in the sense that they are orthogonal over both a given finite interval and $\mathbb{R}$.

*see for example [148, Remark 2.4]. However,* (5.1.17) *holds only asymptotically (and a rectified version of* (5.1.17) *is given in [170, Theorem 2.7]). Specifically, for sufficiently large values of $j$, beyond the critical index $j^\star(\mathsf{w})$, the global maximum of $|\varphi_{\mathsf{w},j}|$ on $[-1,1]$ shifts to the boundary of the interval, becoming equal to $|\varphi_{\mathsf{w},j}(1)|$, which is of order $\mathcal{O}(\sqrt{j})$, as numerically demonstrated in [170, Theorem 8.4]. This shifting behavior, as discussed in both [170, Theorem 3.38], [20, Theorem 3.1], occurs when $\mathsf{w}^2/\chi_{\mathsf{w},j} \leq 1$. The second theorem will be later utilized in our proof of Proposition 5.1.3. For low indices $j$, [20, Remark 3.1] implies that $\|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])} \leq C\sqrt{\mathsf{w}}$. Notably, $\varphi_{\mathsf{w},0}$ is even, positive on the interval $(-1,1)$ and attains its global maximum at the origin. Yet, for low indices $j$, including $j=0$, the behavior of $\varphi_{\mathsf{w},j}(0)$ is poorly understood.*

*Proof of Proposition 5.1.4.* Leveraging Proposition 5.1.3, the claim $\|\varphi_{\mathsf{w},0}\|^2_{L^\infty([-1,1])} \leq 4\mathsf{w}/\pi$ becomes clear, leaving only the proof of (5.1.16) to be addressed for $j \in \mathbb{N}$. Toward this end, we write

$$\|\varphi_{\mathsf{w},j}\|^2_{L^\infty([-1,1])} \leq \Big(\frac{4\mathsf{w}}{\pi}\Big)\mathbb{1}_{\{1\leq j\leq j^\star(\mathsf{w})\}} + \Big(\frac{12}{5}\Big)^2(j+1)\mathbb{1}_{\{j>j^\star(\mathsf{w})\}}. \tag{5.1.18}$$

Let $\tilde{c} = (\frac{12}{5})^2$, and let $h : [1,\infty) \to \mathbb{R}$ given by

$$h(x) := \frac{\log\big(30\mathsf{w}\mathbb{1}_{\{x\leq j^\star(\mathsf{w})\}} + \tilde{c}(x+1)\mathbb{1}_{\{x>j^\star(\mathsf{w})\}}\big)}{\log(x+1)}. \tag{5.1.19}$$

We claim that the function $h$ is nonincreasing on $[1,\infty)$. Observe that the claim is evident for either the case $1 \leq x \leq j^\star(\mathsf{w})$ or for the case $x > j^\star(\mathsf{w})$. Indeed, by definition (5.1.19), on $[1, j^\star(\mathsf{w})]$,

$$h(x) = \frac{\log(30\mathsf{w})}{\log(x+1)}$$

is decreasing, and on $(j^\star(\mathsf{w}),\infty)$,

$$h(x) = \frac{\log(\tilde{c}\,(x+1))}{\log(x+1)} = \frac{\log(\tilde{c})}{\log(x+1)} + 1$$

is also decreasing. Hence it suffices to show, if $1 \leq x_1 \leq j^\star(\mathsf{w}) < x_2 \leq j^\star(\mathsf{w})+1$, then $h(x_2) \leq h(x_1)$. Write $x_2 = j^\star(\mathsf{w}) + \varepsilon = \lfloor 4\mathsf{w}\rfloor - 1 + \varepsilon$ for some $\varepsilon \in (0,1]$. Given that $\mathsf{w} \geq 1$, and so $30\mathsf{w} \geq 6(4\mathsf{w}+1) \geq \tilde{c}(j^\star(\mathsf{w})+\varepsilon+1)$, we gather

$$h(x_1) = \frac{\log(30\mathsf{w})}{\log(x_1+1)} \geq \frac{\log(\tilde{c}(j^\star(\mathsf{w})+\varepsilon+1))}{\log(j^\star(\mathsf{w})+\varepsilon+1)} = h(x_2)-$$

As a consequence, for all $x \geq 1$,

$$\frac{\log\big(30\mathsf{w}\mathbb{1}_{\{x\leq j^\star(\mathsf{w})\}} + \tilde{c}(x+1)\mathbb{1}_{\{x>j^\star(\mathsf{w})\}}\big)}{\log(x+1)} \leq \frac{\log(30\mathsf{w})}{\log(2)} = \gamma(\mathsf{w})$$

or equivalently,

$$30\mathsf{w}\mathbb{1}_{\{1\leq x\leq j^\star(\mathsf{w})\}} + \tilde{c}(x+1)\mathbb{1}_{\{x>j^\star(\mathsf{w})\}} \leq (x+1)^{\gamma(\mathsf{w})}. \tag{5.1.20}$$

Combining (5.1.20), (5.1.18), we can infer that

$$\|\varphi_{\mathsf{w},j}\|^2_{L^\infty([-1,1])} \leq 30\mathsf{w}\mathbb{1}_{\{1\leq j\leq j^\star(\mathsf{w})\}} + \tilde{c}(j+1)\mathbb{1}_{\{j>j^\star(\mathsf{w})\}} \leq (j+1)^{\gamma(\mathsf{w})},$$

as wanted. □

**Extension to Higher Spatial Dimensions.** We now consider the case of Slepian functions in higher spatial dimensions $d \geq 2$. We continue to use $\mathbf{u}$ to denote the uniform measure on $[-1,1]^d$, with $L^p_{\mathbf{u}}([-1,1]^d)$ the space of $p$-integrable functions on $[-1,1]^d$ with respect to the measure $\mathbf{u}$. Then, for multi-indices $\nu = (\nu_1, \ldots, \nu_d) \in \mathbb{N}_0^d$ we construct a basis for $L^2_{\mathbf{u}}([-1,1]^d)$ comprising normalized Slepian basis functions as

$$\varphi_{\mathsf{w},\nu} := \varphi_{\mathsf{w},\nu_1} \otimes \cdots \otimes \varphi_{\mathsf{w},\nu_d}. \tag{5.1.21}$$

It follows by definition of the $\varphi_{\mathsf{w},j}$ above that $\{\varphi_{\mathsf{w},\nu}\}_{\nu\in\mathbb{N}_0^d}$ is an orthonormal basis of $L^2_{\mathbf{u}}([-1,1]^d)$, which we refer to as the Slepian basis in dimension $d$.

## 5.2 Hyperbolic Cross Sets

We now formally define the concept of hyperbolic cross set.

**Definition 5.2.1.** *Let $d, n \in \mathbb{N}$. The $d$-dimensional* **hyperbolic cross** *index set of order $n$ is defined to be*

$$\Lambda^{\mathrm{HC}}_{n-1} := \left\{\nu \in \mathbb{N}_0^d : \prod_{k=1}^d (\nu_k + 1) \leq n\right\}.$$

For the sake of compact presentation we have omitted the dimension $d$ in the notation $\Lambda^{\mathrm{HC}}_{n-1}$, allowing for this information to be inferred from the context. We also use the notation $\Lambda = \Lambda^{\mathrm{HC}}_{n-1}$. From [2, Lemma 5.15], the hyperbolic cross index set satisfies the slicing property:

$$\Lambda^{\mathrm{HC}}_{n-1} = \bigsqcup_{k=0}^{n-1} \{k\} \times \Lambda_k, \tag{5.2.1}$$

where $\bigsqcup$ denotes the disjoint union, and $\Lambda_k := \{(\nu_2, \ldots, \nu_d) : (k, \nu_2, \ldots, \nu_d) \in \Lambda\}$. Moreover, for $0 \leq k \leq j \leq n-1$, we have the nesting property:

$$\#\Lambda_j \leq \#\Lambda_k. \tag{5.2.2}$$

The hyperbolic cross index set also comes with the following convenient and notable estimates on its cardinality, which will play a key role in our analysis.

**Lemma 5.2.2** (See [34, Theorem 3.5] and [2, Lemma B.3])**.** *Let $d, n \in \mathbb{N}$ and suppose $\Lambda^{\mathrm{HC}}_{n-1}$ is the $d$-dimensional hyperbolic cross index set of order $n$. Then there exists $n_\star(d) \in \mathbb{N}$, depending only on $d$, such that if $n \geq n_\star(d)$,*

$$\#\Lambda^{\mathrm{HC}}_{n-1} < \frac{n(\log n + d\log 2)^{d-1}}{(d-1)!}.$$

**Remark 5.2.3.** *An estimate on $n_\star(d)$ can be derived, for example, from the calculations presented in the proof of [34, Theorem 2.4]. In particular, we can take $n_\star(2) = 1$.*

When $\Lambda$ is a hyperbolic cross in low dimensions, an upper bound for $\kappa(\mathcal{S}_\Lambda)$ can be provided by the next proposition.

Later, we us the auxiliary result stated below, which compares the sizes of horizontal slices in two- and three-dimensional hyperbolic cross index sets.

**Proposition 5.2.4.** *Let $n \in \mathbb{N}$ and $d = 2, 3$. Let $\Lambda = \Lambda_{n-1}^{\mathrm{HC}}$ be the $d$-dimensional hyperbolic cross of order $n$ where $\Lambda = \bigsqcup_{k=0}^{n-1} \{k\} \times \Lambda_k$ as in (5.2.1), (5.3.9). Then for $d = 2$ and all $n \geq 2$, or for $d = 3$ and all $n \geq 26$,*

$$\sum_{k=1}^{n-1} (2k+1) \, (\#\Lambda_k)^2 \geq \frac{(\#\Lambda_0)^2}{n}. \tag{5.2.3}$$

*Proof.* For this proof, we rely on Lemma 5.2.2. An estimate on $n_\star(d)$ can be derived, for example, from the calculations presented in the proof of [34, Theorem 2.4]. In particular, we can take $n_\star(2) = 1$. We proceed to prove Proposition 5.2.4 for $d = 2$. In this case, $\Lambda = \Lambda_{n-1}^{\mathrm{HC}}$ is a two-dimensional hyperbolic cross where each slice $\Lambda_k$, for $k = 0, \ldots, n-1$, is a one-dimensional hyperbolic cross. It also follows from (5.3.10) that $\#\Lambda_0 = n$. Therefore, proving the proposition amounts to proving

$$\sum_{k=1}^{n-1} (2k+1) \, (\#\Lambda_k)^2 \geq n.$$

However, it can be seen that the right-hand side term above is at least

$$\sum_{k=1}^{n-1} (2k+1) \, (\#\Lambda_k)^2 \geq \sum_{k=1}^{n-1} (2k+1) = n(n-1) + n - 1 = n^2 - 1 \geq n,$$

when $n \geq 2$, and we are done with $d = 2$.

Turning to $d = 3$, we obtain from the Cauchy-Schwarz inequality that

$$\sum_{k=1}^{n-1} (\#\Lambda_k)^2 (2k+1) \geq \frac{1}{n} \Big( \sum_{k=1}^{n-1} (\#\Lambda_k) \sqrt{2k+1} \Big)^2. \tag{5.2.4}$$

This can be seen if we consider sequences $a_n = 1$ and $b_n = (\#\Lambda_k)\sqrt{2k+1}$. Note that in this case, each $\Lambda_k$ is a two-dimensional hyperbolic cross. Further, we claim that

$$\sum_{k=1}^{n-1} (\#\Lambda_k) \sqrt{2k+1} \geq \#\Lambda_0 \text{ if } n \geq 26. \tag{5.2.5}$$

We demonstrate this. First, since $\#\Lambda_k \geq 1$ for all $k \leq n-1$, we have

$$\sum_{k=1}^{n-1} (\#\Lambda_k) \sqrt{2k+1} \geq \sum_{k=1}^{n-1} \sqrt{2k+1} \geq \int_0^{n-1} (2t+1)^{1/2} \, \mathrm{d}t \geq \frac{(2n-1)^{3/2} - 1}{3}. \tag{5.2.6}$$

Second, from Lemma 5.2.2 and the fact that $n_\star(2) = 1$, we have $\#\Lambda_0 \leq n(\log n + 2\log 2)$, whenever $n \geq 1$. Third,

$$\frac{(2n-1)^{3/2} - 1}{3} \geq n(\log n + 2\log 2), \tag{5.2.7}$$

as soon as $n \geq 26$. It is now apparent that (5.2.5) follows from (5.2.6), (5.2.7).

Combining (5.2.4) and (5.2.5), we deduce for $d = 3$ and $n \geq 26$,

$$\sum_{k=1}^{n-1} (\#\Lambda_k)^2 (2k+1) \geq \frac{1}{n} \Big( \sum_{k=1}^{n-1} (\#\Lambda_k) \sqrt{2k+1} \Big)^2 \geq \frac{(\#\Lambda_0)^2}{n},$$

which is the conclusion we aimed to obtain. □

## 5.3 Least-squares Approximation

The problem of reconstructing a function from a given set of samples has been extensively studied and numerically implemented. In this section, we present the theoretical foundation of the classical and renowned technique of *least squares approximation* for which we will derive a certain PET. We mainly follow results from [2, Section 5].

The main goal of the least squares approximation method is to learn a function $f \in L^2_{\mathbf{u}}([-1,1]^d) \cap C([-1,1]^d)$ from a known set of noisy samples

$$\{(\mathbf{x}_i, (f(\mathbf{x}_i) + \eta_i)\}_{i=1}^m,$$

where each $\mathbf{x}_i \in [-1,1]^d$ is drawn independently from the uniform measure $\mathbf{u}$. We call $\mathbf{e} := \frac{1}{\sqrt{m}}(\eta_j)_{j=1}^m$ the *noise vector*, which satisfies the following

$$\|\mathbf{e}\|_2 = \frac{1}{\sqrt{m}} \sqrt{\sum_{j=1}^m |\eta_j|^2} < \infty.$$

Now, we assume $m \geq L$ and $V \subset C([-1,1])$ to be a finite-dimensional space with dimension $v$. The *least squares approximation* of a function $f \in C([-1,1]^d)$ in the subspace $V$, and denoted by $\widetilde{f} \in V$, is the solution of the minimization problem

$$\widetilde{f} \in \operatorname*{arg\,min}_{g \in V} \frac{1}{m} \sum_{j=1}^m |g(\mathbf{x}_j) - f(\mathbf{x}_j) - \eta_j|^2. \tag{5.3.1}$$

For a basis $\{\psi_j\}_{i=1}^v$ of $V$, we define the *measurement matrix* as

$$A_{ij} := \frac{1}{\sqrt{m}} \psi_i(\mathbf{x}_j). \tag{5.3.2}$$

where $i = 1, \dots, m$ and $j = 1 \dots, dim(V)$, and the *measurement vector* as

$$b_i := \frac{1}{\sqrt{m}}(f(\mathbf{x}_i) + \eta_i). \tag{5.3.3}$$

for $i = 1, \dots, m$. It follows that the least square approximation problem is equivalent to the problem

$$\mathbf{c} \in \arg\min_{\mathbf{d} \in \mathbb{C}^v} \|A\mathbf{d} - \mathbf{b}\|_2. \tag{5.3.4}$$

To compute the approximation error obtained by the least squares problem (5.3.1), we introduce the *discrete stability constant* associated with $V$ and the sample size $m$, as the quantity

$$\alpha = \alpha(V, \{\mathbf{x}_j\}_{j=1}^m) := \inf \left\{ \frac{1}{m} \sum_{j=1}^m |g(\mathbf{x}_j)|^2 : g \in V \text{ and } \|g\|_{L^2_{\mathbf{u}}([-1,1]^d)} = 1 \right\}. \tag{5.3.5}$$

By the Courant-Fisher Min-Max theorem [62, Theorem 4.2.6], it holds

$$\alpha = \sigma_{\min}(A) = \sqrt{\lambda_{\min}(A^*A)}.$$

This notion allows to a first result guaranteeing the accuracy and stability of the least squares approximation, which we present as Theorem 5.3.1.

**Theorem 5.3.1.** *[2, Theorem 5.3] Let $d \in \mathbb{N}$. Let $V \subset L^2([-1,1]^d) \cap C([-1,1]^d)$ be a finite-dimensional space with dimension $v$ and $\mathbf{x}_1, \dots, \mathbf{x}_m$ be distinct points with $m \geq L$. Let $\alpha$ defined as in 5.3.5. Then for all $f \in L^2([-1,1]^d) \cap \mathbb{C}([-1,1]^d)$ and $e \in \mathbb{C}^m$, the problem 5.3.1 as a unique solution $\tilde{f}$ satisfying*

$$\|f - \tilde{f}\|_{L^2_{\mathbf{u}}([-1,1]^d)} \leq (1 + \alpha^{-1}) \inf_{g \in \mathcal{S}_\Lambda} \|f - g\|_{L^\infty_{\mathbf{u}}([-1,1]^d)} + \frac{1}{\alpha}\|\mathbf{e}\|_2.$$

Estimating the constant $\alpha > 0$ might be a challenging task. In many cases, an estimation of this number can be performed via *Monte Carlo sampling*. We now assume that the sample points are randomly and independently drawn from the uniform measure on $[-1, 1]$. For an orthogonal basis $\{\phi_i\}_{i=1}^v$ in $V$, we define

$$\kappa := \kappa(V) := \|K\|_{L^\infty_{\mathbf{u}}([-1,1]^d}$$

where $K : Y \to \mathbb{R}$ is called the *Christoffel function* and is defined as

$$K(\mathbf{x}) := \sum_{i=1}^v |\phi_i(\mathbf{x})|.$$

for all $y \in [-1, 1]$. The knowledge of $\kappa$ allows us to obtain the following result. This is formalized in the following theorem, adapted from [2, Theorem 5.7].

**Theorem 5.3.2.** *Let $\beta, \delta \in (0,1)$ and set*[3]

$$\kappa(\mathcal{S}_\Lambda) := \max_{\mathbf{y}\in[-1,1]^d} \sum_{k=1,\dots,K} |\varphi_{\mathsf{w},\Pi(k)}(\mathbf{y})|^2. \tag{5.3.6}$$

*If $m \geq c_\delta \kappa(\mathcal{S}_\Lambda)\log(K/\beta)$, then with probability at least $1-\beta$, the discrete stability constant $\alpha = \alpha(\mathcal{S}_\Lambda, \{\mathbf{y}_j\}_{j=1}^m)$ satisfies*

$$\alpha > \sqrt{1-\delta}. \tag{5.3.7}$$

We can use Theorem 5.3.2 to modify Theorem 5.3.1 as follows.

**Theorem 5.3.3.** *[2, Corollary 5.3] Let $d \in \mathbb{N}$ and $\beta, \delta \in (0,1)$. Let $V \subset L^2([-1,1]^d) \cap C([-1,1]^d)$ be a finite dimensional space with dimension $v$ and $\mathbf{x}_1, \dots, \mathbf{x}_m$ be distinct points with $m \geq L$. Let $\mathbf{x}_1, \dots, \mathbf{x}_m$ with $m \geq c\kappa\log(v/\beta)$ for a constant $c > 0$. Then for all $f \in L^2([-1,1]^d) \cap C([-1,1]^d)$ and $e \in \mathbb{C}^m$, the problem 5.3.1 as a unique solution $\tilde{f}$ satisfying*

$$\|f - \tilde{f}\|_{L^2_{\mathbf{u}}([-1,1]^d)} \leq (1 + \frac{1}{\sqrt{1-\delta}}) \inf_{g\in\mathcal{S}_\Lambda} \|f-g\|_{L^\infty_{\mathbf{u}}([-1,1]^d)} + \frac{1}{\sqrt{1-\delta}}\|\mathbf{e}\|_2.$$

In order to apply Theorem 5.3.3, we compute an upper bound for $\kappa$ as the following result shows.

**Proposition 5.3.4.** *Let $n \in \mathbb{N}$, and let $\Lambda = \Lambda^{\mathrm{HC}}_{n-1}$ be of dimension $d = 1, 2, 3$. If $\mathsf{w} \geq 1$, and $\mathcal{S}_\Lambda$ denotes the linear span of $\{\varphi_{\mathsf{w},\nu}\}_{\nu\in\Lambda}$, then it holds that*

$$\kappa(\mathcal{S}_\Lambda) \leq (2\,\#\Lambda)^{2\gamma(\mathsf{w})}, \tag{5.3.8}$$

*for the following cases:*

1. *for $d = 1, 2$ and all $n \in \mathbb{N}$,*
2. *for $d = 3$ and all $n \geq 26$.*

*Proof.* To derive an estimate for $\kappa(\mathcal{S}_\Lambda)$, where $\Lambda = \Lambda^{\mathrm{HC}}_{n-1} = \{\nu \in \mathbb{N}_0^d : \prod_{k=1}^d (\nu_k + 1) \leq n\}$ and $d = 1, 2, 3$, we utilize an induction strategy on dimension, as employed in the proof of [2, Proposition 5.13] (also [35, Lemma 3.3]). Our induction step will end at $d = 3$. For the base case $d = 1$, we make use of Proposition 5.1.4 to provide an upper bound for $\kappa(\mathcal{S}_\Lambda)$. For the induction step, we observe from the slicing property (5.2.1) that each $\Lambda_k$ is a $(d-1)$-dimensional hyperbolic cross of order $\lfloor \frac{n}{k+1} \rfloor$, i.e.

$$\Lambda_k = \left\{ \nu' = (\nu_2, \dots, \nu_d) \in \mathbb{N}_0^{d-1} : \prod_{k=2}^{d} (\nu_k + 1) \leq \lfloor \frac{n}{k+1} \rfloor \right\}, \tag{5.3.9}$$

[3] The quantity $\kappa(\mathcal{S}_\Lambda)$ in (5.3.6) is known as the $L^\infty$-norm of the *Christoffel function* of $\mathcal{S}_\Lambda$ [2, Definition 5.4].

and particularly,

$$\Lambda_0 = \left\{ \nu' = (\nu_2, \ldots, \nu_d) \in \mathbb{N}_0^{d-1} : \prod_{k=2}^{d} (\nu_k + 1) \leq n \right\} \tag{5.3.10}$$

is a $(d-1)$-dimensional hyperbolic cross of exactly order $n$.

Returning to the proof of Proposition 5.3.4, we note that $\gamma(\mathsf{w}) > 1$ for $\mathsf{w} \geq 1$. Moreover

$$2^{\gamma(\mathsf{w})-1} \geq \frac{4\mathsf{w}}{\pi}. \tag{5.3.11}$$

When $d = 1$, $\Lambda = \Lambda_{n-1}^{\mathrm{HC}} = \{0, \ldots, n-1\}$. On the one hand, if $n = 1$, then it follows from Proposition 5.1.4 and (5.3.11) that

$$\kappa(\mathcal{S}_\Lambda) = \|\varphi_{\mathsf{w},0}\|_{L^\infty([-1,1])}^2 \leq \frac{4\mathsf{w}}{\pi} \leq 2^{\gamma(\mathsf{w})} = (\sqrt{2}\,\#\Lambda)^{2\gamma(\mathsf{w})}, \tag{5.3.12}$$

which satisfies (5.3.8). On the other hand, if $n \geq 2$, then we can again apply Proposition 5.1.4 to obtain

$$\kappa(\mathcal{S}_\Lambda) = \sum_{k=0}^{n-1} \|\varphi_{\mathsf{w},k}\|_{L^\infty([-1,1])}^2 \leq \frac{4\mathsf{w}}{\pi} + \sum_{k=1}^{n-1} (k+1)^{\gamma(\mathsf{w})} \leq \frac{4\mathsf{w}}{\pi} + \sum_{k=1}^{n-1} (2k+1)^{\gamma(\mathsf{w})}. \tag{5.3.13}$$

We claim that

$$\frac{4\mathsf{w}}{\pi} + \sum_{k=1}^{n-1} (2k+1)^{\gamma(\mathsf{w})} \leq \Big( \sum_{k=0}^{n-1} 2k+1 \Big)^{\gamma(\mathsf{w})} = (n^2-1)^{\gamma(\mathsf{w})} \leq (2n^2)^{\gamma(\mathsf{w})} = (\sqrt{2}\,\#\Lambda)^{2\gamma(\mathsf{w})}. \tag{5.3.14}$$

It is enough to validate the first inequality in (5.3.14). Then by Pascal's identity [138, Chapter 6.4, Theorem 2] and (5.3.11), we have

$$\begin{aligned}
\Big( \sum_{k=0}^{n-1} 2k+1 \Big)^{\gamma(\mathsf{w})} &= \Big( \Big( \sum_{k=1}^{n-1} 2k+1 \Big) + 1 \Big)^{\gamma(\mathsf{w})} \\
&= \sum_{l=0}^{\gamma(\mathsf{w})-1} \binom{\gamma(\mathsf{w})}{l} \Big( \sum_{k=1}^{n-1} 2k+1 \Big)^{l} + \Big( \sum_{k=1}^{n-1} 2k+1 \Big)^{\gamma(\mathsf{w})} \\
&\geq \sum_{l=0}^{\gamma(\mathsf{w})-1} \binom{\gamma(\mathsf{w})}{l} \Big( \sum_{k=1}^{n-1} 2k+1 \Big)^{l} + \sum_{k=1}^{n-1} (2k+1)^{\gamma(\mathsf{w})} \\
&\geq \sum_{l=0}^{\gamma(\mathsf{w})-1} \binom{\gamma(\mathsf{w})-1}{l} 3^l + \sum_{k=1}^{n-1} (2k+1)^{\gamma(\mathsf{w})} \\
&= 4^{\gamma(\mathsf{w})-1} + \sum_{k=1}^{n-1} (2k+1)^{\gamma(\mathsf{w})} \\
&\geq \frac{4\mathsf{w}}{\pi} + \sum_{k=1}^{n-1} (2k+1)^{\gamma(\mathsf{w})}.
\end{aligned}$$

Therefore, (5.3.14) holds. By combining (5.3.14) and (5.3.13), we deduce (5.3.8) for $d = 1$ and $n \geq 2$.

Turning to the case $d = 2$ or $d = 3$, we have that $\Lambda = \Lambda_{n-1}^{\mathrm{HC}}$ is a two-dimensional or three-dimensional hyperbolic cross, where each $\Lambda_k$ is a one-dimensional or two-dimensional hyperbolic cross, respectively. In what follows, we will employ an argument applicable for both the case $d = 2$ when $n \geq 2$ and for $d = 3$ when $n \geq 26$. For the latter case, we additionally assume that (5.3.8) already holds for $d = 2$ for all $n \in \mathbb{N}$. The separate case of $d = 2$ and $n = 1$ will be addressed at the end.

Let $d = 2$ and $n \geq 2$. By applying (5.3.8) to each one-dimensional hyperbolic cross $\Lambda_k$ and invoking Proposition 5.1.4, we obtain

$$\begin{aligned}
\kappa(\mathcal{S}_\Lambda) &= \sum_{k=0}^{n-1} \|\varphi_{\mathsf{w},k}\|_{L^\infty([-1,1])}^2 \sum_{j \in \Lambda_k} \|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])}^2 \\
&= \|\varphi_{\mathsf{w},0}\|_{L^\infty([-1,1])}^2 \sum_{j \in \Lambda_0} \|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])}^2 + \sum_{k=1}^{n-1} \|\varphi_{\mathsf{w},k}\|_{L^\infty([-1,1])}^2 \sum_{j \in \Lambda_k} \|\varphi_{\mathsf{w},j}\|_{L^\infty([-1,1])}^2 \\
&\leq \frac{4\mathsf{w}}{\pi} (2\,\#\Lambda_0)^{2\gamma(\mathsf{w})} + \sum_{k=1}^{n-1} (k+1)^{\gamma(\mathsf{w})} (2\,\#\Lambda_k)^{2\gamma(\mathsf{w})}. \qquad (5.3.15)
\end{aligned}$$

Similarly, let $d = 3$ and $n \geq 26$. Assuming that (5.3.8) holds for all two-dimensional hyperbolic cross index sets $\Lambda_k$, we can write

$$\begin{aligned}
\kappa(\mathcal{S}_\Lambda) &= \sum_{k=0}^{n-1} \|\varphi_{\mathsf{w},k}\|_{L^\infty([-1,1])}^2 \sum_{\nu \in \Lambda_k} \|\varphi_{\mathsf{w},\nu}\|_{L^\infty([-1,1]^2)}^2 \\
&\leq \frac{4\mathsf{w}}{\pi} (2\,\#\Lambda_0)^{2\gamma(\mathsf{w})} + \sum_{k=1}^{n-1} (k+1)^{\gamma(\mathsf{w})} (2\,\#\Lambda_k)^{2\gamma(\mathsf{w})}. \qquad (5.3.16)
\end{aligned}$$

As with the case $d = 1$, we aim to dominate the right-hand sides of (5.3.15), (5.3.16) using the quantity $(\sum_{k=0}^{n-1}(2k+1)\,(2\,\#\Lambda_k)^2)^{\gamma(\mathsf{w})}$. For brevity, we denote $A_k := 2\,\#\Lambda_k$. Then

$$\begin{aligned}
&\Big(\sum_{k=0}^{n-1} (2k+1)\, A_k^2\Big)^{\gamma(\mathsf{w})} \\
&= \Big(\sum_{k=1}^{n-1} (2k+1)\, A_k^2\Big)^{\gamma(\mathsf{w})} + \sum_{l=1}^{\gamma(\mathsf{w})} \binom{\gamma(\mathsf{w})}{l} \Big(\sum_{k=1}^{n-1} (2k+1)\, A_k^2\Big)^{\gamma(\mathsf{w})-l} A_0^{2l} \\
&\geq \sum_{k=1}^{n-1} (2k+1)^{\gamma(\mathsf{w})} A_k^{2\gamma(\mathsf{w})} + \sum_{l=1}^{\gamma(\mathsf{w})} \binom{\gamma(\mathsf{w})}{l} \Big(\sum_{k=1}^{n-1} (2k+1)\, A_k^2\Big)^{\gamma(\mathsf{w})-l} A_0^{2l}. \qquad (5.3.17)
\end{aligned}$$

An application of Proposition 5.2.4 gives

$$\Big(\sum_{k=1}^{n-1} (2k+1)\, A_k^2\Big)^{\gamma(\mathsf{w})-l} \geq \frac{A_0^{2(\gamma(\mathsf{w})-l)}}{n^{\gamma(\mathsf{w})-l}}.$$

Therefore

$$\sum_{l=1}^{\gamma(\mathsf{w})}\binom{\gamma(\mathsf{w})}{l}\Big(\sum_{k=1}^{n-1}(2k+1)\,A_k^2\Big)^{\gamma(\mathsf{w})-l}A_0^{2l} \geq \frac{A_0^{2\gamma(\mathsf{w})}}{n^{\gamma(\mathsf{w})}}\sum_{l=1}^{\gamma(\mathsf{w})}\binom{\gamma(\mathsf{w})}{l}n^l$$
$$= A_0^{2\gamma(\mathsf{w})}\big(1+n+n^2+\cdots+n^{\gamma(\mathsf{w})-1}\big)$$
$$\geq A_0^{2\gamma(\mathsf{w})}2^{\gamma(\mathsf{w})-1}$$
$$\geq A_0^{2\gamma(\mathsf{w})}\frac{4\mathsf{w}}{\pi}, \tag{5.3.18}$$

where the last inequality is due to (5.3.11). Putting back $A_k = 2\,\#\Lambda_k$, and combining (5.3.17), (5.3.18) together with (5.3.15) or (5.3.16) yields

$$\kappa(\mathcal{S}_\Lambda) \leq \frac{4\mathsf{w}}{\pi}(2\,\#\Lambda_0)^{2\gamma(\mathsf{w})} + \sum_{k=1}^{n-1}(k+1)^{\gamma(\mathsf{w})}(2\,\#\Lambda_k)^{2\gamma(\mathsf{w})}$$
$$\leq 4^{\gamma(\mathsf{w})}\Big(\sum_{k=0}^{n-1}(2k+1)\,(\#\Lambda_k)^2\Big)^{\gamma(\mathsf{w})}. \tag{5.3.19}$$

Next, by the monotonicity property (5.2.2)

$$k(\#\Lambda_k)^2 = \sum_{j=1}^{k}(\#\Lambda_k)^2 \leq \sum_{j=0}^{k-1}(\#\Lambda_k)(\#\Lambda_j),$$

we can conclude from (5.3.19) that

$$\kappa(\mathcal{S}_\Lambda) \leq 4^{\gamma(\mathsf{w})}\Big(\sum_{k=0}^{n-1}(2k+1)\,(\#\Lambda_k)^2\Big)^{\gamma(\mathsf{w})}$$
$$\leq 4^{\gamma(\mathsf{w})}\Big(\sum_{k=0}^{n-1}(\#\Lambda_k)^2 + 2\sum_{j<k}(\#\Lambda_k)(\#\Lambda_j)\Big)^{\gamma(\mathsf{w})}$$
$$= 4^{\gamma(\mathsf{w})}\Big(\sum_{k=0}^{n-1}\#\Lambda_k\Big)^{2\gamma(\mathsf{w})} = (2\,\#\Lambda)^{2\gamma(\mathsf{w})},$$

which is (5.3.8) for $d=2$ when $n\geq 2$ and for $d=3$ when $n\geq 26$.

It remains to check the validity of (5.3.8) when $d=2$ and $n=1$. In this case, $\Lambda=\{(0,0)\}$. Then performing a similar calculation as in (5.3.12) yields

$$\kappa(\mathcal{S}_\Lambda) = \|\varphi_{\mathsf{w},(0,0)}\|^2_{L^\infty([-1,1]^2)} = \|\varphi_{\mathsf{w},0}\|^4_{L^\infty([-1,1])} \leq \frac{16\mathsf{w}^2}{\pi^2} \leq 2^{2\gamma(\mathsf{w})} = (2\,\#\Lambda)^{2\gamma(\mathsf{w})}.$$

With this, we close the induction loop and conclude (5.3.8) for $d=1,2$ for all $n\in\mathbb{N}$ and for $d=3$ for all $n\geq 26$. □

We are ready to present our result on least squares approximation when we consider the Slepian basis.

**Theorem 5.3.5** (Recovery guarantee for least squares)**.** *Let $\beta, \delta \in (0,1)$, $d = 1,2,3$, and $\Lambda = \Lambda_{n-1}^{\mathrm{HC}}$ be the $d$-dimensional hyperbolic cross of order $n$, where $n \in \mathbb{N}$ if $d = 1,2$, and $n \geq 26$ if $d = 3$. For $\mathsf{w} \geq 1$, let $\mathcal{S}_\Lambda$ be the linear span of $\{\varphi_{\mathsf{w},\nu}\}_{\nu\in\Lambda}$ and let $\mathbf{y}_1, \dots, \mathbf{y}_m$ be random sample points drawn independently from the uniform measure $\mathbf{u}$ on $[-1,1]^d$. If $m \geq c_\delta \kappa^\star \log(\#\Lambda/\beta)$, where*

$$\kappa^\star := (2\#\Lambda)^{2\gamma(\mathsf{w})} \quad \text{and} \quad c_\delta := ((1-\delta)\log(1-\delta)+\delta)^{-1}, \tag{5.3.20}$$

*then, with probability at least $1-\beta$, for every $f \in C([-1,1]^d)$ and every noise vector $\mathbf{e} = \frac{1}{\sqrt{m}}(\eta_j)_{j=1}^m \in \mathbb{C}^m$, the least squares problem (5.3.1) has a unique solution $f^\natural \in \mathcal{S}_\Lambda$ satisfying*

$$\|f - f^\natural\|_{L^2_{\mathbf{u}}([-1,1]^d)} \leq \left(1 + \frac{1}{\sqrt{1-\delta}}\right) \inf_{g\in\mathcal{S}_\Lambda} \|f-g\|_{L^\infty([-1,1]^d)} + \frac{1}{\sqrt{1-\delta}}\|\mathbf{e}\|_2. \tag{5.3.21}$$

*Proof.* Assuming the validity of (5.3.7), the theory outlined in [2, Section 5], particularly [2, Corollary 5.9], demonstrates that every $f \in C([-1,1]^d)$ has a unique least squares approximation $f^\natural$ in (5.3.1) that satisfies

$$\begin{aligned}\|f - f^\natural\|_{L^2_{\mathbf{u}}([-1,1]^d)} &\leq (1+\alpha^{-1}) \inf_{g\in\mathcal{S}_\Lambda} \|f-g\|_{L^\infty([-1,1]^d)} + \frac{1}{\alpha}\|\mathbf{e}\|_2 \\ &< \left(1 + \frac{1}{\sqrt{1-\delta}}\right) \inf_{g\in\mathcal{S}_\Lambda} \|f-g\|_{L^\infty([-1,1]^d)} + \frac{1}{\sqrt{1-\delta}}\|\mathbf{e}\|_2. \end{aligned} \tag{5.3.22}$$

Since the index set $\Lambda = \Lambda_{n-1}^{\mathrm{HC}}$ contains $\mathbf{0}$, the function $\varphi_{\mathsf{w},\mathbf{0}}$ will always be included in the span $\mathcal{S}_\Lambda$. Proposition 5.1.4 and Remark 5.1.5 suggest that estimating $\|\varphi_{\mathsf{w},\mathbf{0}}\|_{L^\infty([-1,1]^d)}$ may pose complications, as an upper bound for this value could reach $\mathsf{w}^d$, leading to an exponential growth of $\kappa(\mathcal{S}_\Lambda)$ in high dimensions. Consequently, we restrict the selection of $\Lambda = \Lambda_{n-1}^{\mathrm{HC}}$ to dimensions up to $d = 3$.

If we accept Proposition 5.3.4, then the rationale behind the choice of $\kappa^\star = (2\,\#\Lambda)^{2\gamma(\mathsf{w})}$ in (5.3.20) can be seen from (5.3.8). Specifically, by ensuring $m \geq c_\delta \kappa^\star \log(\#\Lambda/\beta)$, the proof of Theorem 5.3.5 can now be completed through a combination of Theorem 5.3.2 and (5.3.22). □

### 5.3.1 Discussion on Further Recovery Guarantees for Least Squares

We now turn to two additional recovery guarantees for least squares approximation. The reason we present these results is that the recovery guarantee in Theorem 5.3.5 comes with the drawback of bounding the error in terms of the best approximation of $f$ in the $L^\infty$-norm, which is a stronger and more restrictive norm compared to the $L^2_{\mathbf{u}}$-norm used to measure the recovery error. Thus, here we present two results aimed at addressing this shortcoming. They can be seen as direct corollaries of Theorem 5.3.2 and Proposition 5.3.4, while their proofs are omitted as they are identical to other published results.

The first result is an $L^2_{\mathbf{u}}$-$L^2_{\mathbf{u}}$ uniform recovery guarantee in probability. It is stated as follows.

**Corollary 5.3.6** ($L^2_{\mathbf{u}}$-$L^2_{\mathbf{u}}$ recovery guarantee in probability for least squares)**.** *In the same setting of Theorem 5.3.5, let $\beta, \delta \in (0,1)$ and $f \in \mathcal{C}([-1,1]^d)$. Assume that the number of samples $m$ satisfies the condition $m \geq c\kappa^\star \log(2\#\Lambda/\beta)$, where $c > 0$ is a universal constant, and $\kappa^\star$ is defined as in Theorem 5.3.5. Then for every noise vector $e \in \mathbb{C}^m$, with probability at least $1-\beta$, the least squares solution $f^\natural$ obtained in* (5.3.1) *satisfies*

$$\|f - f^\natural\|_{L^2_{\mathbf{u}}([-1,1]^d)} \leq \left(1 + \sqrt{\frac{2}{\beta}}\frac{1}{\sqrt{1-\delta}}\right) \inf_{g \in \mathcal{S}_\Lambda} \|f - g\|_{L^2_{\mathbf{u}}([-1,1]^d)} + \frac{1}{\sqrt{1-\delta}}\|\vec{e}\|_2.$$

This result can be obtained from Theorem 5.3.2 and Proposition 5.3.4 by applying the same argument as in [2, Corollary 5.10]. The error bound now contains the best approximation error of $f$ with respect to the $L^2_{\mathbf{u}}$-norm as desired. However, as opposed to Theorem 5.3.5, Corollary 5.3.6 is a *nonuniform recovery guarantee*, i.e., the error bound holds with probability $1-\beta$ for a fixed $f$. In other words, one single draw of sample points is not sufficient for the validity of the recovery guarantee for all $f \in C([-1,1]^d)$ simultaneously. Another limitation of Corollary 5.3.6 is the poor scaling of the error bound with respect to the failure probability $\beta$. The following result addresses these limitations by providing a recovery guarantee in expectation.

**Corollary 5.3.7** ($L^2_{\mathbf{u}}$-$L^2_{\mathbf{u}}$ recovery guarantee in expectation for least squares)**.** *Let $\delta, \beta \in (0,1)$ and $f \in C([-1,1]^d)$ be such that $\|f\|_{L^2_{\mathbf{u}}([-1,1]^d)} \leq L$ for some $L > 0$. Define the operator $\tau_L : L^2_{\mathbf{u}}([-1,1]^d) \to L^2_{\mathbf{u}}([-1,1]^d)$ as*

$$\tau_L(g) := \min\left\{1, \frac{L}{\|g\|_{L^2_{\mathbf{u}}([-1,1]^d)}}\right\} g.$$

*Then, under the same assumptions as Theorem 5.3.5, if $f^\natural$ is the least squares solution of* (5.3.1)*, for every noise vector $e \in \mathbb{C}^m$ we have*

$$\mathbb{E}\|f - \tau_L(f^\natural)\|^2_{L^2_{\mathbf{u}}([-1,1]^d)} \leq \left(\frac{3-\delta}{1-\delta}\right) \inf_{g \in \mathcal{S}_\Lambda} \|f - g\|_{L^2_{\mathbf{u}}([-1,1]^d)} + \frac{2}{1-\delta}\|\vec{e}\|_2^2 + 4L^2\beta.$$

This result can obtained from Theorem 5.3.2 and Proposition 5.3.4 by applying the same argument as in [2, Corollary 5.11] (originally proposed in [39, Theorem 2]). Notably, the poor scaling with respect to $\beta$ in the error bound of Corollary 5.3.6 is not there anymore. However, this comes at the price of having a recovery guarantee in expectation, as opposed to high probability, while also requiring a priori knowledge of a constant $L$ such that $\|f\|_{L^2_{\mathbf{u}}([-1,1]^d)} \leq L$.

We conclude with a brief remark that the proof of Theorem 5.5.1 can also be modified to incorporate $L^2_{\mathbf{u}}$-$L^2_{\mathbf{u}}$ recovery guarantees by employing Corollary 5.3.6 or 5.3.7 in place of Theorem 5.3.5. For the sake of conciseness, we omit the statements of the resulting corollaries.

## 5.4 Approximation of the Slepian Basis by NNs

It is known that the normalized Legendre polynomials $\{P_k\}_{k\in\mathbb{N}_0}$ in one dimension can be effectively approximated by ReLU neural networks. A concrete example is provided in [119, Proposition 2.11] (see also Proposition 2.2.10). In the case of $d$ dimensions, we frequently rely on the tensorization definition (5.1.21) and the fundamental network calculus operations from Chapter 2, such as concatenation, parallelization, and product approximation, to develop approximating NNs for normalized Legendre polynomials. For a bandwidth $\mathsf{w} \geq 1$ we define

$$\gamma(\mathsf{w}) := \lceil \log_2(30\mathsf{w}) \rceil, \tag{5.4.1}$$

to be used in the ensuing statements. Using the definition of $\gamma(\mathsf{w})$ in (5.4.1) for $\mathsf{w} \geq 1$, for $n \in \mathbb{N}$ we further define

$$B(d,n) := \begin{cases} n^{\gamma(\mathsf{w})} + 1 & \text{if } d = 1, \\ 3n^{2\gamma(\mathsf{w})} + 4n^{\gamma(\mathsf{w})} + 2 & \text{if } d = 2, \\ 7n^{3\gamma(\mathsf{w})} + 12n^{2\gamma(\mathsf{w})} + 8n^{\gamma(\mathsf{w})} + 3 & \text{if } d = 3, \end{cases} \tag{5.4.2}$$

and

$$M(d,n) := \begin{cases} 1 & \text{if } d = 1, \\ 2n^{\gamma(\mathsf{w})} + 1 & \text{if } d = 2, \\ 4n^{2\gamma(\mathsf{w})} + 4n^{\gamma(\mathsf{w})} + 2 & \text{if } d = 3. \end{cases} \tag{5.4.3}$$

We are now ready to present Proposition 5.4.1.

**Proposition 5.4.1.** *Let $\varepsilon \in (0,1)$, and $\mathsf{w} \geq 1$. Let $\Lambda = \Lambda_{n-1}^{\mathrm{HC}}$ be the $d$-dimensional hyperbolic cross of order $n \geq 2$, with $d = 1,2,3$. Let $B(d,n)$, $M(d,n)$ be given (5.4.2), (5.4.3), respectively. Then there exist a universal constant $C > 0$ and $N_\star = N_\star(\Lambda, \mathsf{w}, \varepsilon) \in \mathbb{N}$ for which the following holds. For every Slepian basis function $\varphi_{\mathsf{w},\nu}$ where $\nu \in \Lambda_{n-1}^{\mathrm{HC}}$, there exists an NN $\Psi_{\varepsilon,N_\star}^{\nu}$ such that*

$$\|\varphi_{\mathsf{w},\nu} - \mathrm{R}(\Psi_{\varepsilon,N_\star}^{\nu})\|_{L^\infty([-1,1]^d)} \leq B(d,n)\varepsilon,$$

*with*

$$\begin{aligned} L(\Psi_{\varepsilon,N_\star}^{\nu}) &\leq C\big((1+\log_2 N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2(M(d,n)/\varepsilon)\big), \\ W(\Psi_{\varepsilon,N_\star}^{\nu}) &\leq C\big((N_\star^2 + N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2(M(d,n)/\varepsilon)\big). \end{aligned}$$

Before we prove Proposition 5.4.1, we need the following two supporting results. The first, Lemma 5.4.2, establishes that every $\varphi_{\mathsf{w},k}$ can be well-approximated by a linear combination of normalized Legendre polynomials, with a controllable number of terms. The second, Lemma 5.4.3, establishes that any such linear combination can be efficiently modeled by an NN of manageable length and complexity.

**Lemma 5.4.2.** *Let $\mathsf{w} \geq 1$. Let $\Lambda \subset \mathbb{N}$ be a finite set. For $k \in \Lambda$, let $\varphi_{\mathsf{w},k}$ be the $k$th Slepian basis function associated with the eigenvalue $\mu_{\mathsf{w},k}$ (5.1.2). For every $\varepsilon \in (0,1)$, let*

$$N_\star := \left\lceil \max\left\{2\lfloor e\mathsf{w}\rfloor + 1, \frac{\log(3/(c_\star\varepsilon))}{\log(3/2)}\right\}\right\rceil, \tag{5.4.4}$$

*where $c_\star := \min_{k\in\Lambda} \mu_{\mathsf{w},k}$. For every normalized Legendre polynomial $P_j$ of degree $j \in \mathbb{N}$, let*

$$\beta_j^k := \frac{1}{2}\int_{-1}^{1} \varphi_{\mathsf{w},k}(x)\overline{P_j}(x)\,\mathrm{d}x. \tag{5.4.5}$$

*Then for every $N \geq N_\star$ and for every $k \in \Lambda$, it holds that*

$$\left\|\varphi_{\mathsf{w},k} - \sum_{j=0}^{N} \beta_j^k P_j\right\|_{L^\infty([-1,1])} \leq \varepsilon. \tag{5.4.6}$$

**Lemma 5.4.3.** *Let $\varepsilon \in (0,1)$, and $N \in \mathbb{N}$. Let $\{\beta_j\}_{j=0}^N$ be a set of real numbers such that $|\beta_j| \leq c_0$ for some $c_0 > 0$. Let $P_j$ be the normalized Legendre polynomial of degree $j$. Then there exists an NN $\widetilde{\Phi}_{\varepsilon,N}$ such that*

$$\left\|\sum_{j=0}^{N} \beta_j P_j - \mathrm{R}(\widetilde{\Phi}_{\varepsilon,N})\right\|_{L^\infty([-1,1])} \leq c_0\varepsilon.$$

*Moreover,*

$$\begin{aligned} L(\widetilde{\Phi}_{\varepsilon,N}) &\leq C(1+\log_2 N)(N+\log_2(N/\varepsilon)),\\ W(\widetilde{\Phi}_{\varepsilon,N}) &\leq C(N^2+N)(N+\log_2(N/\varepsilon)), \end{aligned}$$

*for some universal constant $C > 0$.*

The proofs of Lemmas 5.4.2, 5.4.3 are given at the end of this subsection, in the order in which they are presented.

*Proof of Proposition 5.4.1.* We begin with the case $\Lambda = \Lambda_{n-1}^{\mathrm{HC}} \subset \mathbb{N}$, i.e. when $\{\varphi_{\mathsf{w},k}\}_{k\in\Lambda}$ is a finite subset of the Slepian orthonormal basis of $L^2_{\mathbf{u}}([-1,1])$. Let $N_\star$ be as given in (5.4.4), with $\varepsilon \in (0,1)$. Note, when $\Lambda = \Lambda_{n-1}^{\mathrm{HC}} = \{0,\dots,n-1\}$,

$$c_\star = \min_{k\in\Lambda} \mu_{\mathsf{w},k} = \mu_{\mathsf{w},n-1}. \tag{5.4.7}$$

By Lemma 5.4.2, (5.4.6) holds with $N = N_\star$. To apply Lemma 5.4.3 to our context, we need an estimate for $\max_{k\in\Lambda}\max_{j=0,\dots,N_\star}|\beta_j^k|$, which will act as the value $c_0$. Since $n \geq 2$, we obtain from Proposition 5.1.4 that

$$\|\varphi_{\mathsf{w},k}\|_{L^\infty([-1,1])} \leq \max\left\{\frac{4\mathsf{w}}{\pi}, (k+1)^{\gamma(\mathsf{w})}\right\} \leq n^{\gamma(\mathsf{w})}, \tag{5.4.8}$$

and subsequently, from (5.4.5),

$$|\beta_j^k| = \frac{1}{2}\Big|\int_{-1}^{1} \varphi_{\mathsf{w},k}(x)\overline{P_j}(x)\,\mathrm{d}x\Big| \le \|P_j\|_{L^2_{\mathbf{u}}([-1,1])}\|\varphi_{\mathsf{w},k}\|_{L^\infty_{\mathbf{u}}([-1,1])}$$
$$= \|P_j\|_{L^2_{\mathbf{u}}([-1,1])}\|\varphi_{\mathsf{w},k}\|_{L^\infty([-1,1])} \le n^{\gamma(\mathsf{w})}.$$

Thus, Lemma 5.4.3 ensures the existence of an NN $\widetilde{\Phi}^k_{\varepsilon,N_\star}$ satisfying

$$\Big\|\sum_{j=0}^{N_\star}\beta_j^k P_j - \mathrm{R}(\widetilde{\Phi}^k_{\varepsilon,N_\star})\Big\|_{L^\infty([-1,1])} \le n^{\gamma(\mathsf{w})}\varepsilon, \tag{5.4.9}$$

with the following respective numbers of layers and weights

$$L(\widetilde{\Phi}^k_{\varepsilon,N_\star}) \le C(1+\log_2 N_\star)(N_\star + \log_2(N_\star/\varepsilon)), \tag{5.4.10}$$
$$W(\widetilde{\Phi}^k_{\varepsilon,N_\star}) \le C(N_\star^2 + N_\star)(N_\star + \log_2(N_\star/\varepsilon)). \tag{5.4.11}$$

By setting $N = N_\star$ in (5.4.6) and combining it with (5.4.9), we derive

$$\Big\|\varphi_{\mathsf{w},k} - \mathrm{R}(\widetilde{\Phi}^k_{\varepsilon,N_\star})\Big\|_{L^\infty([-1,1])} \le \big(1+n^{\gamma(\mathsf{w})}\big)\varepsilon, \tag{5.4.12}$$

and we conclude for the case $\Lambda^{\mathrm{HC}}_{n-1} \subset \mathbb{N}$ by letting $\Psi^k_{\varepsilon,N_\star} = \widetilde{\Phi}^k_{\varepsilon,N_\star}$.

For the case $\Lambda = \Lambda^{\mathrm{HC}}_{n-1} \subset \mathbb{N}^2$, we write, $\varphi_{\mathsf{w},\nu} = \varphi_{\mathsf{w},\nu_1}\otimes\varphi_{\mathsf{w},\nu_2}$, for each $\nu = (\nu_1,\nu_2) \in \Lambda^{\mathrm{HC}}_{n-1}$. Note, $\nu_j \in \{0,\dots,n-1\}$. Let $N_\star$ remain as previously defined with the choice $c_\star$ in (5.4.7). Then by (5.4.12), for $j = 1,2$, there exists an NN $\widetilde{\Phi}^{\nu_j}_{\varepsilon,N_\star}$, such that

$$\|\varphi_{\mathsf{w},\nu_j} - \mathrm{R}(\widetilde{\Phi}^{\nu_j}_{\varepsilon,N_\star})\|_{L^\infty([-1,1])} \le (1+n^{\gamma(\mathsf{w})})\varepsilon. \tag{5.4.13}$$

This, together with (5.4.8), implies

$$\|\mathrm{R}(\widetilde{\Phi}^{\nu_j}_{\varepsilon,N_\star})\|_{L^\infty([-1,1])} \le \|\varphi_{\mathsf{w},\nu_j}\|_{L^\infty([-1,1])} + (1+n^{\gamma(\mathsf{w})})\varepsilon \le 2n^{\gamma(\mathsf{w})}+1. \tag{5.4.14}$$

Combining (5.4.8), (5.4.13), (5.4.14), we derive for $\nu = (\nu_1,\nu_2) \in \Lambda^{\mathrm{HC}}_{n-1}$ that

$$\|\varphi_{\mathsf{w},\nu_1}\otimes\varphi_{\mathsf{w},\nu_2} - \mathrm{R}(\widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star})\otimes\mathrm{R}(\widetilde{\Phi}^{\nu_2}_{\varepsilon,N_\star})\|_{L^\infty([-1,1]^2)} \le a+b, \tag{5.4.15}$$

where

$$a = \|\varphi_{\mathsf{w},\nu_1}\otimes\varphi_{\mathsf{w},\nu_2} - \varphi_{\mathsf{w},\nu_1}\otimes\mathrm{R}(\widetilde{\Phi}^{\nu_2}_{\varepsilon,N_\star})\|_{L^\infty([-1,1]^2)}$$
$$\le \|\varphi_{\mathsf{w},\nu_1}\|_{L^\infty([-1,1])}\|\varphi_{\mathsf{w},\nu_2} - \mathrm{R}(\widetilde{\Phi}^{\nu_2}_{\varepsilon,N_\star})\|_{L^\infty([-1,1])} \le n^{\gamma(\mathsf{w})}(1+n^{\gamma(\mathsf{w})})\varepsilon,$$

and

$$b = \|\varphi_{\mathsf{w},\nu_1}\otimes\mathrm{R}(\widetilde{\Phi}^{\nu_2}_{\varepsilon,N_\star}) - \mathrm{R}(\widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star})\otimes\mathrm{R}(\widetilde{\Phi}^{\nu_2}_{\varepsilon,N_\star})\|_{L^\infty([-1,1]^2)}$$
$$\le \|\mathrm{R}(\widetilde{\Phi}^{\nu_2}_{\varepsilon,N_\star})\|_{L^\infty([-1,1])}\|\varphi_{\mathsf{w},\nu_1} - \mathrm{R}(\widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star})\|_{L^\infty([-1,1])} \le (2n^{\gamma(\mathsf{w})}+1)(1+n^{\gamma(\mathsf{w})})\varepsilon.$$

Therefore, (5.4.15) yields

$$\|\varphi_{\mathsf{w},\nu_1} \otimes \varphi_{\mathsf{w},\nu_2} - \mathrm{R}(\widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star}) \otimes \mathrm{R}(\widetilde{\Phi}^{\nu_2}_{\varepsilon,N_\star})\|_{L^\infty([-1,1])} \leq (3n^{2\gamma(\mathsf{w})} + 4n^{\gamma(\mathsf{w})} + 1)\varepsilon. \tag{5.4.16}$$

Next, we utilize the network construction $\tilde{\times}_{\varepsilon,B}$ in Proposition 2.2.7, with $B = 2n^{\gamma(\mathsf{w})} + 1$ (obtained from (5.4.14)), as well as the concatenation and parallelization operations, denoted as $\odot$ and P, from Definitions 2.0.5 and 2.0.6 respectively, to construct

$$\Psi^{\nu}_{\varepsilon,N_\star} := \tilde{\times}_{\varepsilon,B} \odot \mathrm{P}(\widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star}, \widetilde{\Phi}^{\nu_2}_{\varepsilon,N_\star}).$$

It follows that

$$\|\mathrm{R}(\widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star}) \otimes \mathrm{R}(\widetilde{\Phi}^{\nu_2}_{\varepsilon,N_\star}) - \mathrm{R}(\Psi^{\nu}_{\varepsilon,N_\star})\|_{L^\infty([-1,1]^2)} \leq \varepsilon. \tag{5.4.17}$$

Combining (5.4.16), (5.4.17), we obtain

$$\|\varphi_{\mathsf{w},\nu_1} \otimes \varphi_{\mathsf{w},\nu_2} - \mathrm{R}(\Psi^{\nu}_{\varepsilon,N_\star})\|_{L^\infty([-1,1]^2)} \leq (3n^{2\gamma(\mathsf{w})} + 4n^{\gamma(\mathsf{w})} + 2)\varepsilon. \tag{5.4.18}$$

Furthermore, by applying (5.4.10), (5.4.11), Proposition 2.2.7, and Remark 2.0.7, we conclude

$$\begin{aligned} L(\Psi^{\nu}_{\varepsilon,N_\star}) &\leq C\big((1+\log_2 N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2((2n^{\gamma(\mathsf{w})}+1)/\varepsilon)\big) \\ W(\Psi^{\nu}_{\varepsilon,N_\star}) &\leq C\big((N_\star^2 + N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2((2n^{\gamma(\mathsf{w})}+1)/\varepsilon)\big), \end{aligned}$$

and thus, we are done with the case $\Lambda^{\mathrm{HC}}_{n-1} \subset \mathbb{N}^2$.

For the case $d = 3$, let $\nu = (\nu_1, \nu_2, \nu_3) \in \Lambda^{\mathrm{HC}}_{n-1}$ and denote $\varphi_{\mathsf{w},\nu} = \varphi_{\mathsf{w},\nu_1} \otimes \varphi_{\mathsf{w},\nu_2} \otimes \varphi_{\mathsf{w},\nu_3}$. By (5.4.12), there exists an NN $\widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star}$ such that

$$\|\varphi_{\mathsf{w},\nu_1} - \mathrm{R}(\widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star})\|_{L^\infty([-1,1])} \leq (1 + n^{\gamma(\mathsf{w})})\varepsilon. \tag{5.4.19}$$

Denote $\nu^* = (\nu_2, \nu_3)$. Then by (5.4.18), there also exists an NN $\Psi^{\nu^*}_{\varepsilon,N_\star}$ satisfying

$$\|\varphi_{\mathsf{w},\nu_2} \otimes \varphi_{\mathsf{w},\nu_3} - \mathrm{R}(\Psi^{\nu^*}_{\varepsilon,N_\star})\|_{L^\infty([-1,1]^2)} \leq (3n^{2\gamma(\mathsf{w})} + 4n^{\gamma(\mathsf{w})} + 2)\varepsilon. \tag{5.4.20}$$

Altogether, using (5.4.8), (5.4.14), (5.4.19), (5.4.20), via similar arguments to those leading to (5.4.16), we conclude

$$\begin{aligned} \|\varphi_{\mathsf{w},\nu_1} \otimes \varphi_{\mathsf{w},\nu_2} \otimes \varphi_{\mathsf{w},\nu_3} - \mathrm{R}(\widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star}) \otimes \mathrm{R}(\Psi^{\nu^*}_{\varepsilon,N_\star})\|_{L^\infty([-1,1]^3)} \\ \leq (7n^{3\gamma(\mathsf{w})} + 12n^{2\gamma(\mathsf{w})} + 8n^{\gamma(\mathsf{w})} + 2)\varepsilon. \end{aligned} \tag{5.4.21}$$

Let $\Phi_I$ be the NN formulation in Proposition 2.0.8 (with $d = 1$) such that $\Psi^{\nu_1}_{\varepsilon,N_\star} := \Phi_I \odot \widetilde{\Phi}^{\nu_1}_{\varepsilon,N_\star}$ and $\Psi^{\nu^*}_{\varepsilon,N_\star}$ have the same number of layers. We construct

$$\Psi^{\nu}_{\varepsilon,N_\star} := \tilde{\times}_{\varepsilon,B'} \odot \mathrm{P}(\Psi^{\nu_1}_{\varepsilon,N_\star}, \Psi^{\nu^*}_{\varepsilon,N_\star})$$

where $B' = 4n^{2\gamma(\mathsf{w})} + 4n^{\gamma(\mathsf{w})} + 2$. Then from Proposition 2.2.7 again and (5.4.21),

$$\|\varphi_{\mathsf{w},\nu_1} \otimes \varphi_{\mathsf{w},\nu_2} \otimes \varphi_{\mathsf{w},\nu_3} - \mathrm{R}(\Psi^{\nu}_{\varepsilon,N_\star})\|_{L^\infty([-1,1]^3)} \leq (7n^{3\gamma(\mathsf{w})} + 12n^{2\gamma(\mathsf{w})} + 8n^{\gamma(\mathsf{w})} + 3)\varepsilon.$$

Since it is readily verified from the construction of $\Psi^{\nu_1}_{\varepsilon,N_\star}$ and (5.4.8), (5.4.19), (5.4.20) that

$$\max\big\{|\mathrm{R}(\Psi^{\nu_1}_{\varepsilon,N_\star})|, |\mathrm{R}(\Psi^{\nu^*}_{\varepsilon,N_\star})|\big\} \leq 4n^{2\gamma(\mathsf{w})} + 4n^{\gamma(\mathsf{w})} + 2,$$

employing similar reasoning and computations analogous to those before, delivers

$$\begin{aligned} L(\Psi^{\nu}_{\varepsilon,N_\star}) &\leq C\big((1+\log_2 N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2((4n^{2\gamma(\mathsf{w})} + 4n^{\gamma(\mathsf{w})} + 2)/\varepsilon)\big) \\ W(\Psi^{\nu}_{\varepsilon,N_\star}) &\leq C\big((N_\star^2 + N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2((4n^{2\gamma(\mathsf{w})} + 4n^{\gamma(\mathsf{w})} + 2)/\varepsilon)\big). \end{aligned}$$

The proof is now complete. □

*Proof of Lemma 5.4.2.* The set of all normalized Legendre polynomials is an orthonormal basis for $L^2_{\mathbf{u}}([-1,1])$. Therefore, for every Slepian basis function $\varphi_{\mathsf{w},k}$, there exist $\{\beta^k_j\}_{j\in\mathbb{N}_0} \in \ell^2(\mathbb{N}_0)$ satisfying (5.4.5) such that

$$\lim_{N\to\infty} \Big\| \sum_{j=0}^{N} \beta^k_j P_j - \varphi_{\mathsf{w},k} \Big\|_{L^2_{\mathbf{u}}([-1,1])} = 0; \tag{5.4.22}$$

see also [120, Equation (2.47)]. Since $L^\infty([-1,1])$ is a Banach space, the absolute convergence of the series $\sum_{j=0}^{\infty} \beta^k_j P_j$, i.e.

$$\lim_{N\to\infty} \sum_{j=N+1}^{\infty} |\beta^k_j| \|P_j\|_{L^\infty([-1,1])} = 0, \tag{5.4.23}$$

will guarantee the existence of the limit $\sum_{j=0}^{\infty} \beta^k_j P_j$ as a function in $L^\infty([-1,1])$ [51, Theorem 5.1]. Once held, (5.4.23) together with (5.4.22), and the continuity of $\sum_{j=0}^{N} \beta^k_j P_j$ and $\varphi_{\mathsf{w},k}$, yields

$$\sum_{j=0}^{\infty} \beta^k_j P_j = \varphi_{\mathsf{w},k}, \tag{5.4.24}$$

on $[-1,1]$. We proceed to demonstrate (5.4.23) for all $k \in \Lambda$. By [120, Theorem 7.2], if $j \geq 2(\lfloor e\mathsf{w} \rfloor + 1)$, then

$$|\beta^k_j| = \frac{1}{2}\Big| \int_{-1}^{1} \varphi_{\mathsf{w},k}(x)\overline{P_j}(x)\,\mathrm{d}x \Big| \leq \frac{1}{\mu_{\mathsf{w},k}} \Big(\frac{1}{2}\Big)^j \leq \frac{1}{c_\star}\Big(\frac{1}{2}\Big)^j. \tag{5.4.25}$$

Using (2.2), (5.4.25), we find for $N \geq 2\lfloor e\mathsf{w} \rfloor + 1$

$$\begin{aligned}\sum_{j=N+1}^{\infty} |\beta_j^k| \|P_j\|_{L^\infty([-1,1])} &\leq \frac{1}{c_\star} \sum_{j=N+1}^{\infty} \frac{\sqrt{2j+1}}{2^j} \\ &\leq \frac{1}{c_\star} \sum_{j=N}^{\infty} \frac{\sqrt{j}}{2^j} \\ &\leq \frac{1}{c_\star} \sum_{j=N}^{\infty} \Big(\frac{2}{3}\Big)^j \\ &= \frac{3}{c_\star} \Big(\frac{2}{3}\Big)^N, \end{aligned} \tag{5.4.26}$$

which is (5.4.23), valid for all $k \in \Lambda$. Hence, (5.4.24) holds, and we conclude from (5.4.26) that,

$$\Big\| \sum_{j=0}^{N} \beta_j^k P_j - \varphi_{\mathsf{w},k} \Big\|_{L^\infty([-1,1])} = \Big\| \sum_{j=N+1}^{\infty} \beta_j^k P_j \Big\|_{L^\infty([-1,1])} \leq \frac{3}{c_\star} \Big(\frac{2}{3}\Big)^N, \tag{5.4.27}$$

whenever $N \geq 2\lfloor e\mathsf{w} \rfloor - 1$. For the final step, we require in addition that

$$N \geq \max\Big\{1, \frac{\log(3/(c_\star \varepsilon))}{\log(3/2)}\Big\}$$

in (5.4.27), to get (5.4.6), as desired. □

*Proof of Lemma 5.4.3.* Let $j = 0, \ldots, N$. A result derived from [119, Proposition 2.11], and stated as Proposition 2.2.10, guarantees for $j = 1, \ldots, N$, the existence of an NN $\Phi_\varepsilon^j$ with both input dimension and output dimension $d_{L_j}^j$ equal to 1, such that

$$\|P_j - \mathrm{R}(\Phi_\varepsilon^j)\|_{L^\infty([-1,1])} \leq \varepsilon, \tag{5.4.28}$$

and for $j = 0$, an NN $\Phi_\varepsilon^0$ satisfying $\mathrm{R}(\Phi_\varepsilon^0) = P_0 = 1$ on $[-1, 1]$. Moreover,

$$L(\Phi_\varepsilon^j) \leq C(1 + \log_2 j)(j + \log_2(1/\varepsilon)) \quad \text{and} \quad W(\Phi_\varepsilon^j) \leq Cj(j + \log_2(1/\varepsilon)), \tag{5.4.29}$$

and $L(\Phi_\varepsilon^0) = 2$, $W(\Phi_\varepsilon^0) = 1$. Therefore, by Proposition 2.0.9, there exists an NN $\widetilde{\Phi}_{\varepsilon,N}$ that fulfills

$$\mathrm{R}(\widetilde{\Phi}_{\varepsilon,N}) = \sum_{j=0}^{N} \beta_j \mathrm{R}(\Phi_\varepsilon^j), \tag{5.4.30}$$

with, from (5.4.29), the number of layers being at most

$$L(\widetilde{\Phi}_{\varepsilon,N}) = \max_{0 \leq j \leq N} L(\Phi_\varepsilon^j) \leq C(1 + \log_2 N)(N + \log_2(1/\varepsilon)), \tag{5.4.31}$$

and the number of weights being at most,

$$\begin{aligned} W(\widetilde{\Phi}_{\varepsilon,N}) &\leq 2\sum_{j=0}^{N} W(\Phi_\varepsilon^j) + 2\sum_{j=0}^{N} d_{L_j}^j \Big( \max_{0\leq j\leq N} L(\Phi_\varepsilon^j) \Big) \\ &\leq C\sum_{j=1}^{N} j(j+\log_2(1/\varepsilon)) + CN(1+\log_2 N)(N+\log_2(1/\varepsilon)) \\ &\leq C(N^2+N)(N+\log_2(1/\varepsilon)) + CN(1+\log_2 N)(N+\log_2(1/\varepsilon)) \\ &\leq C(N^2+N)(N+\log_2(1/\varepsilon)). \end{aligned} \tag{5.4.32}$$

Now, from (5.4.28), (5.4.30), we obtain

$$\begin{aligned} \Big\| \sum_{j=0}^{N} \beta_j P_j - \mathrm{R}(\widetilde{\Phi}_{\varepsilon,N}) \Big\|_{L^\infty([-1,1])} &= \Big\| \sum_{j=0}^{N} \beta_j P_j - \sum_{j=0}^{N} \beta_j \mathrm{R}(\Phi_\varepsilon^j) \Big\|_{L^\infty([-1,1])} \\ &\leq \sum_{j=0}^{N} |\beta_j| \Big\| P_j - \mathrm{R}(\Phi_\varepsilon^j) \Big\|_{L^\infty([-1,1])} \\ &\leq c_0 N\varepsilon. \end{aligned} \tag{5.4.33}$$

By replacing $N\varepsilon$ with $\varepsilon$ in (5.4.31), (5.4.32), (5.4.33), we arrive at the desired conclusion. □

## 5.5 Practical Existence Theorem

Now, we present the practical existence theorem of this section and its proof. We observe that this is stated in a broader context that includes frequency-localized functions.

**Theorem 5.5.1** (Practical existence theorem)**.** *Let $\varepsilon, \delta, \beta \in (0,1)$, $d = 1,2,3$, and $\Lambda = \Lambda_{n-1}^{\mathrm{HC}}$ be the $d$-dimensional hyperbolic cross of order $n$, where $n \in \mathbb{N}$ if $d = 1,2$, and $n \geq 26$ if $d = 3$. Let $\mathsf{w} \geq 1$ and $B(d,n)$, $M(d,n)$ be as in (5.4.2), (5.4.3), respectively. Let $\mathbf{y}_1, \ldots, \mathbf{y}_m$ be independent sample points drawn from the uniform measure $\mathbf{u}$ on $[-1,1]^d$, with $m \geq c_\delta \kappa^\star \log(\#\Lambda/\beta)$, where $\kappa^\star$, $c_\delta$ are given in (5.3.20). Then for every $\varepsilon > 0$ sufficiently small satisfying*

$$\varepsilon \leq \frac{\sqrt{1-\delta}}{2\sqrt{\#\Lambda}B(d,n)}, \tag{5.5.1}$$

*there exists a class of NNs, denoted $\mathcal{N}_{\Lambda,\mathsf{w},\varepsilon}$, such that the following holds with probability at least $1-\beta$. For every $f \in \mathcal{C}([-1,1]^d)$ and every noise vector $\mathbf{e} = \frac{1}{\sqrt{m}}(\eta_j)_{j=1}^m \in \mathbb{C}^m$, the problem*

$$\Psi^\natural \in \operatorname*{arg\,min}_{\Psi\in\mathcal{N}_{\Lambda,\mathsf{w},\varepsilon}} \frac{1}{m}\sum_{j=1}^{m} |f(\mathbf{y}_j) + \eta_j - \mathrm{R}(\Psi)(\mathbf{y}_j)|^2 \tag{5.5.2}$$

*has a unique solution* $\Psi^\natural \in \mathcal{N}_{\Lambda,\mathsf{w},\varepsilon}$ *satisfying*

$$\begin{aligned}&\|f - \mathrm{R}(\Psi^\natural)\|_{L^2_{\mathbf{u}}([-1,1]^d)}\\ &\leq \Big(1 + \frac{2}{\sqrt{1-\delta}}\Big)\Big(\inf_{g\in\mathcal{S}_\Lambda} \|f-g\|_{L^\infty([-1,1]^d)} + \sqrt{\#\Lambda}B(d,n)\|g\|_{L^2_{\mathbf{u}}([-1,1]^d)}\varepsilon\Big) + \frac{2\|\mathbf{e}\|_2}{\sqrt{1-\delta}}. \quad (5.5.3)\end{aligned}$$

*Moreover, it is guaranteed that*

$$\begin{aligned}L(\Psi^\natural) &\leq C\big((1+\log_2 N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2(M(d,n)/\varepsilon)\big),\\ W(\Psi^\natural) &\leq C\#\Lambda\big((N_\star^2 + N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2(M(d,n)/\varepsilon)\big),\end{aligned}$$

*for a universal constant* $C > 0$ *and* $N_\star = N_\star(\Lambda, \mathsf{w}, \varepsilon) \in \mathbb{N}$.

Similar to Theorem 5.3.5, Theorem 5.5.1 states that functions well-approximated by the Slepian system $\{\varphi_{\mathsf{w},\nu}\}_{\nu\in\Lambda}$ can be accurately recovered via deep learning using a training set of pointwise samples with size proportional to $\kappa^\star = (2\#\Lambda)^{2\gamma(\mathsf{w})}$ (up a constant and a log factor). Theorem 5.5.1 assumes training to be performed by solving a least squares-type optimization problem (5.5.2) that closely resembles (5.3.1), operating over a class of NNs $\mathcal{N}_{\Lambda,\mathsf{w},\varepsilon}$ with specific architectures, determined by $\Lambda$, $\mathsf{w}$, $\varepsilon$. The class $\mathcal{N}_{\Lambda,\mathsf{w},\varepsilon}$ is defined in (5.5.9) and is composed by NNs whose first-to-second-last layers are explicitly constructed to emulate the Slepian basis and whose last layer is made of trainable weights, optimized through (5.5.2). Theorem 5.5.1 provides explicit bounds on the depth $L(\Psi^\natural)$ and width $W(\Psi^\natural)$ of NNs in the class $\mathcal{N}_{\Lambda,\mathsf{w},\varepsilon}$, while the dependence of $N_\star$ on $\Lambda$, $\mathsf{w}$, $\varepsilon$ is detailed later in the proof of Theorem 5.5.1, particularly in (5.4.4). The error bound (5.5.3) of Theorem 5.5.1 is similar to (5.3.21) in Theorem 5.3.5, so that besides minor changes in the constant factors, the main difference between (5.5.3) and (5.3.21) is the presence of an extra term in the best approximation error. The impact of this extra term is mild, though, as it scales linearly with an auxiliary parameter $\varepsilon \in (0,1)$ that can be made arbitrarily close to 0 at the price of increasing the architecture bounds proportionally to $\log(1/\varepsilon)$.

*Proof.* To simplify the presentation, we adopt the convention introduced at the beginning of Subsection 5.2. Namely, we use $\Lambda$ to denote $\Lambda^{\mathrm{HC}}_{n-1}$, and let $K = \#\Lambda = \#\Lambda^{\mathrm{HC}}_{n-1}$. For $\nu \in \Lambda$, we write $\varphi_{\mathsf{w},\nu} = \varphi_{\mathsf{w},\Pi(k)}$, where $\Pi : \{1,\dots,K\} \to \Lambda$ is a bijection. Now let $g \in \mathcal{S}_\Lambda$, i.e.

$$g = \sum_{k=1}^K b_k \varphi_{\mathsf{w},\Pi(k)}, \quad (5.5.4)$$

for $b_k := \frac{1}{2}\int_{-1}^1 g(x)\varphi_{\mathsf{w},\Pi(k)}(x)\,\mathrm{d}x$. According to Proposition 5.4.1, for a given $\varepsilon > 0$ and each Slepian basis function $\varphi_{\mathsf{w},\Pi(k)}$, there exists an NN $\Psi^k_\varepsilon$ such that[4]

$$\|\varphi_{\mathsf{w},k} - \mathrm{R}(\Psi^k_\varepsilon)\|_{L^\infty([-1,1]^d)} \leq B(d,n)\varepsilon. \quad (5.5.5)$$

[4]Given that $N_\star$, in (5.4.4), depends on $\varepsilon$, we will omit its reference in the subscript and simply write $\Psi^k_\varepsilon$ going forward.

In turn, by Proposition 2.0.9, there exists an NN $\Psi_{g;\varepsilon}$ satisfying

$$\mathrm{R}(\Psi_{g;\varepsilon}) = \sum_{k=1}^{K} b_k \mathrm{R}(\Psi_\varepsilon^k). \tag{5.5.6}$$

Thus we obtain

$$\begin{aligned} \|g - \mathrm{R}(\Psi_{g;\varepsilon})\|_{L^\infty([-1,1]^d)} &\leq \sum_{k=1}^{K} |b_k| \|\varphi_{\mathsf{w},k} - \mathrm{R}(\Psi_\varepsilon^k)\|_{L^\infty_{\mathbf{u}}([-1,1]^d)} \\ &\leq \sqrt{K} B(d,n) \|g\|_{L^2_{\mathbf{u}}([-1,1]^d)} \varepsilon, \end{aligned} \tag{5.5.7}$$

from (5.5.5) and the fact that $\|g\|_{L^2_{\mathbf{u}}([-1,1]^d)} = (\sum_{k=1}^{K} |b_k|^2)^{1/2}$. It also follows from Proposition 5.4.1 and another application of Proposition 2.0.9 that

$$\begin{aligned} L(\Psi_{g;\varepsilon}) &\leq C\big((1+\log_2 N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2(M(d,n)/\varepsilon)\big), \\ W(\Psi_{g;\varepsilon}) &\leq CK\big((N_\star^2 + N_\star)(N_\star + \log_2(N_\star/\varepsilon)) + (1+\log_2(M(d,n)/\varepsilon)\big). \end{aligned} \tag{5.5.8}$$

Now, let

$$\mathcal{N}_{\Lambda,\mathsf{w},\varepsilon} := \{\Psi_{g;\varepsilon} : \Psi_{g;\varepsilon} \text{ satisfies (5.5.6) where } g \text{ takes the form of (5.5.4)}\}. \tag{5.5.9}$$

We will show in what follows that, given $\mathcal{N}_{\Lambda,\mathsf{w},\varepsilon}$ in (5.5.9), the optimization problem (5.5.2) admits a unique solution fulfilling (5.5.3).

Recall the matrix $A \in \mathbb{R}^{m\times K}$, introduced in (5.3.2), where $A_{jk} = \dfrac{1}{\sqrt{m}} \varphi_{\mathsf{w},\Pi(k)}(\mathbf{y}_j)$. Since the Slepian basis is an orthonormal set, the columns of $A$ are linearly independent. Consider $\tilde{A} \in \mathbb{R}^{m\times K}$, such that $\tilde{A}_{jk} = \dfrac{1}{\sqrt{m}} \mathrm{R}(\Psi_\varepsilon^k)(\mathbf{y}_j)$. Then it follows from (5.5.5) that the columns of $\tilde{A}$ are also linearly independent for sufficiently small $\varepsilon$. Indeed, let $\varepsilon \in (0, 1/(\sqrt{K}B(d,n)))$. If $0 = \sum_{k=1}^{K} b_k \mathrm{R}(\Psi_\varepsilon^k)$ for some not-all-zero scalars $\{b_k\}_{k=1}^K$, then

$$\sum_{k=1}^{K} b_k(\varphi_\varepsilon^k - \mathrm{R}(\Psi_\varepsilon^k)) = \sum_{k=1}^{K} b_k \varphi_\varepsilon^k,$$

which leads to

$$\begin{aligned} \Big(\sum_{k=1}^{K} |b_k|^2\Big)^{1/2} = \Big\| \sum_{k=1}^{K} b_k \varphi_\varepsilon^k \Big\|_{L^2_{\mathbf{u}}([-1,1]^d)} &= \Big\| \sum_{k=1}^{K} b_k(\varphi_\varepsilon^k - \mathrm{R}(\Psi_\varepsilon^k)) \Big\|_{L^2_{\mathbf{u}}([-1,1]^d)} \\ &\leq \sum_{k=1}^{K} |b_k| \|\varphi_\varepsilon^k - \mathrm{R}(\Psi_\varepsilon^k)\|_{L^2_{\mathbf{u}}([-1,1]^d)} \\ &\leq \varepsilon\sqrt{K} B(d,n) \Big(\sum_{k=1}^{K} |b_k|^2\Big)^{1/2} < \Big(\sum_{k=1}^{K} |b_k|^2\Big)^{1/2}, \end{aligned}$$

a contradiction. Since, due to (5.5.6), every element in $\mathcal{N}_{\Lambda,\mathsf{w},\varepsilon}$ is realized as a linear combination of $\mathrm{R}(\Psi_\varepsilon^k)$, the problem (5.5.2) is equivalent to finding $(\mathbf{z})^\natural \in \mathcal{C}^K$ such that

$$(\mathbf{z})^\natural \in \operatorname*{arg\,min}_{\mathbf{z}\in\mathcal{C}^K} \|\tilde{A}\mathbf{z} - \mathbf{b}\|_2^2$$

where $\mathbf{b} = \frac{1}{\sqrt{m}}(f(\mathbf{y}_j) + \eta_j)_{j=1}^m$. We look at the smallest singular value of $\tilde{A}$ next. By Weyl's Lemma [62, Theorem 4.3.1], we have

$$\max_{j=1,\dots,m} |\sigma_j(A) - \sigma_j(\tilde{A})| \le \|A - \tilde{A}\|_2 \tag{5.5.10}$$

where $\sigma_j$ denotes the $j$th singular value of $A$. Hence, for all unit vector $\mathbf{z} \in \mathcal{C}^K$,

$$\begin{aligned}
\|(A-\tilde{A})\mathbf{z}\|_2^2 &= \sum_{j=1}^m |\sum_{k=1}^K (A_{jk} - \tilde{A}_{jk}) z_k|^2 \\
&\le \sum_{j=1}^m \left( \sum_{k=1}^K |A_{jk} - \tilde{A}_{jk}||z_k| \right)^2 \\
&\le \sum_{j=1}^m \left( \sum_{k=1}^K \frac{1}{\sqrt{m}} \|\varphi_{\mathsf{w},k} - \mathrm{R}(\Psi_\varepsilon^k)\|_{L^\infty([-1,1]^d)} |z_k| \right)^2 \\
&= \left( \sum_{k=1}^K \|\varphi_{\mathsf{w},k} - \mathrm{R}(\Psi_\varepsilon^k)\|_{L^\infty([-1,1]^d)} |z_k| \right)^2 \\
&\le \sum_{k=1}^K \|\varphi_{\mathsf{w},k} - \mathrm{R}(\Psi_\varepsilon^k)\|_{L^\infty([-1,1]^d)}^2 \|\mathbf{z}\|_2^2 \\
&\le KB(d,n)^2\varepsilon^2,
\end{aligned}$$

where we have used the Cauchy-Schwarz inequality in the third inequality and (5.5.5) in the fourth. Therefore,

$$\|A - \tilde{A}\|_2 \le \sqrt{K} B(d,n)\varepsilon. \tag{5.5.11}$$

We have demonstrated in the proof of Theorem 5.3.5 that $\sigma_{\min}(A) > \sqrt{1-\delta}$ with probability at least $1-\beta$, as long as $m \ge c_\delta \kappa^\star \log(\#\Lambda/\beta)$. Hence, from (5.5.10), (5.5.11), we get

$$\sigma_{\min}(\tilde{A}) \ge \sigma_{\min}(A) - \|A - \tilde{A}\|_2 \ge \sigma_{\min}(A) - \sqrt{K}B(d,n)\varepsilon > \sqrt{1-\delta} - \sqrt{K}B(d,n)\varepsilon, \tag{5.5.12}$$

with the same probability. Thus, by choosing $\varepsilon \in (0, 1/(\sqrt{K}B(d,n)))$ additionally sufficiently small such that

$$\sqrt{K}B(d,n)\varepsilon \le \frac{1}{2}\sqrt{1-\delta},$$

which is (5.5.1), we further obtain from (5.5.12)

$$\sigma_{\min}(\tilde{A}) > \frac{1}{2}\sqrt{1-\delta}. \tag{5.5.13}$$

By applying the least squares approximation theory from Subsection 5.3, specifically leveraging (5.3.22) and the lower bound given in (5.5.13), we conclude that (5.5.2) has, with probability at least $1-\beta$, a unique solution $\Psi^\natural$ such that

$$\|f - \mathrm{R}(\Psi^\natural)\|_{L^2_{\mathbf{u}}([-1,1]^d)} \leq \left(1 + \frac{2}{\sqrt{1-\delta}}\right) \inf_{\Psi \in \mathcal{N}_{\Lambda,\mathsf{w},\varepsilon}} \|f - \mathrm{R}(\Psi)\|_{L^\infty([-1,1]^d)} + \frac{2}{\sqrt{1-\delta}}\|\mathbf{e}\|_2.$$

Additionally, by (5.5.7), for any $\Psi \in \mathcal{N}_\varepsilon$ and any $g \in \mathcal{S}_\Lambda$,

$$\begin{aligned}\|f - \mathrm{R}(\Psi)\|_{L^\infty([-1,1]^d)} &\leq \|f - g\|_{L^\infty([-1,1]^d)} + \|g - \mathrm{R}(\Psi)\|_{L^\infty([-1,1]^d)} \\ &\leq \|f - g\|_{L^\infty([-1,1]^d)} + \sqrt{K}B(d,n)\|g\|_{L^2_{\mathbf{u}}([-1,1]^d)}\varepsilon,\end{aligned}$$

and we arrive at (5.5.3). Finally, we note that the necessary upper bounds on the layers and weights for $\Psi^\natural$ have already been specified in (5.5.8).

□

## 5.6 Numerical Experiments

In this section, we provide numerical examples illustrating the performance of least squares approximation and deep learning in reconstructing frequency-localized functions from pointwise samples. For simplicity, we refer to these methods as "Least Squares" and "Deep Learning" throughout. We describe our numerical setup and discuss the results obtained in dimensions one and two. The Python code necessary to reproduce our experiments can be found in the GitHub repository: `https://github.com/andreslerma01/PSWF-`.

**Benchmark functions.** We assess the performance of Least Squares and Deep Learning for the reconstruction of the one-dimensional function

$$f_1(x) := \cos(10x)e^{-\pi x^2}, \qquad \forall x \in [-1,1],$$

and the two-dimensional function

$$f_2(x,y) := \cos(0.2x)\cos(0.2y)e^{-\pi(x^2+y^2)}, \qquad \forall (x,y) \in [-1,1]^2.$$

The one-dimensional function $f_1$, defined as the product of a trigonometric function and a Gaussian, was considered in [105] as a simple example in which the Slepian basis achieves better approximation accuracy than the standard Fourier basis. The two-dimensional function $f_2$ generalizes the one-dimensional function $f_1$ while reducing the cosine frequency from 10 to 0.2. Both functions are frequency-localized since most of their energy is concentrated within a compact interval in the frequency domain.

**Training set, test set, and error metric.** For a given target function $f$ that we seek to approximate using either Least Squares or Deep Learning, we generate data sets of the form $\{(x_j, f(x_j)\}_{j=1}^m$ with $x_j$ drawn independently from the uniform distribution on $[-1,1]^d$ and the sample size $m$ taking values up to 10000. For a set of $m$ training points, we also consider a test set $\{x_j^{\text{test}}\}_{j=1}^{m_{\text{test}}}$ of size $m_{\text{test}} = 0.2m$, sampled independently according to the uniform distribution on $[-1,1]^d$. The corresponding test error is given by the *Root Mean Square Error (RMSE)*, defined as

$$\mathcal{E}_{\text{test}} := \sqrt{\frac{1}{m_{\text{test}}} \sum_{j=1}^{m_{\text{test}}} |f(x_j^{\text{test}}) - \tilde{f}(x_j^{\text{test}})|^2},$$

where $f$ is the target function, and $\tilde{f}$ is the approximation obtained from either method. Specifically, adopting the notation in Section 5.3, $\tilde{f}$ for Least Squares, and $\tilde{f} = \mathrm{R}(\Psi^\natural)$ for Deep Learning, where $\Psi^\natural$ is the trained NN.

**Slepian basis functions.** To construct one-dimensional Slepian basis functions $\varphi_{\mathsf{w},j}$, we employ the Python package `scipy.special` [165], which is based on the angular solutions of the Helmholtz wave equation of the first kind. In fact, each function $\varphi_{\mathsf{w},j}$ is a scalar multiple of a corresponding angular solution [105]. To construct two-dimensional Slepian basis functions, we utilize the tensor-product definition (5.1.21) $\varphi_{\mathsf{w},\nu} = \varphi_{\mathsf{w},\nu_1} \otimes \varphi_{\mathsf{w},\nu_2}$. Finally, in both cases, we take the index set to be the hyperbolic cross $\Lambda_{n-1}^{\mathrm{HC}}$ (Definition 5.2.1) for various orders $n \in \mathbb{N}$.

**Neural network architecture and training.** Following standard deep learning practice, we consider fully trained NNs. Here, we deviate from the theoretical setting of Theorem 5.5.1, where only the last layer is trained. NN architectures are built according to the ratio $r = L/N = 0.1$, where $L$ represents the number of hidden layers and $N$ the number of neurons per layer. Specifically, we conduct experiments for $L = 1, \ldots, 10$, with $r = 0.1$, a choice empirically supported by [3]. Weights and biases are initialized from a normal distribution with mean 0 and standard deviation 0.1. Training is performed using Adam [77] with an exponential decay rate of 0.85 for 150 epochs. To ensure the robustness and reliability of the results, each experiment is repeated 20 times.

**One-dimensional results.** Figure 5.2 illustrates the results of our first experiment on approximating the one-dimensional function $f_1$. In accordance with our theoretical guarantees, both methods successfully approximate this frequency-localized function, with the RMSE converging as the number of sampling points increases. Moreover, the number of parameters in each reconstruction model significantly affects the RMSE. For Least Squares, increasing the number of Slepian basis functions (corresponding to increasing the order $n$) generally lowers test errors, with the only recorded exception being $n = 36$. Similarly, for Deep Learning, increasing the number of network layers $L$ results in a noticeable error reduction. However, note that Least Squares typically requires fewer samples as well as significantly

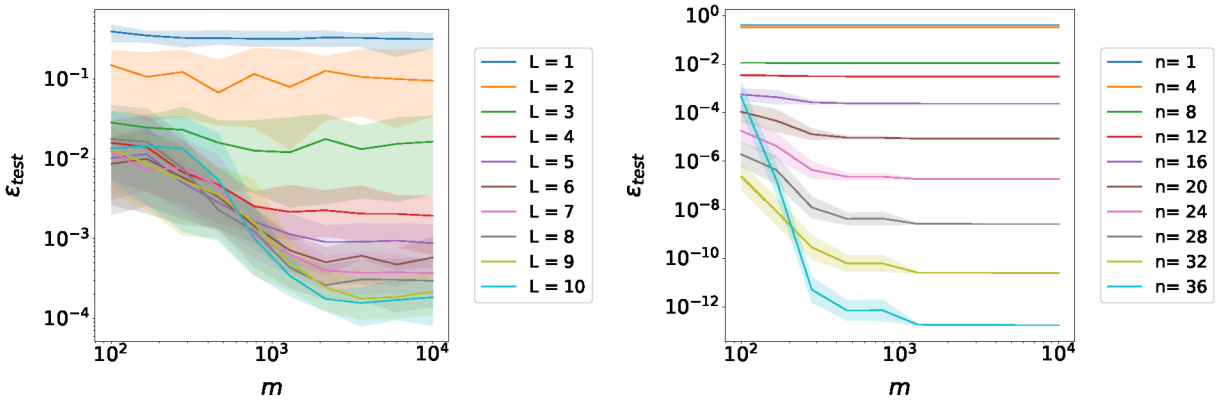


Figure 5.2: Comparison of reconstruction methods for $f_1(x) = \cos(10x)e^{-\pi x^2}$ from samples. The left panel shows results using Deep Learning, while the right panel shows results using Least Squares. Here, $L$ denotes the number of layers in each NN, and $n$ denotes the order of the hyperbolic cross $\Lambda^{\mathrm{HC}}_{n-1}$.

fewer parameters to achieve high accuracy. The best accuracy achieved by Least Squares (RMSE $\approx 10^{-12}$) is several orders of magnitude higher than that by Deep Learning (RMSE $\approx 10^{-4}$). In summary, in one dimension, Least Squares is much more effective than Deep Learning for the approximation of $f_1$.

**Two-dimensional results.** Figure 5.3 illustrates the results of our second experiment on approximating the two-dimensional function $f_2$. Both Least Squares and Deep Learning successfully approximate this frequency-localized function, in line with our theory. As in the one-dimensional case, we observe RMSE convergence as a function of the number of samples $m$. However, there are notable differences between the one- and two-dimensional cases. First, the best accuracy achieved by both methods is now much more similar. Both Least Squares and Deep Learning achieve a best mean RMSE of around $10^{-4}$. The accuracy gap between Least Squares and Deep Learning is much less pronounced in dimension two than in dimension one. Second, although the advantage of adding parameters to the model (for both Least Squares and Deep Learning) is still there, we observe some intriguing numerical phenomena. For Least Squares, considering a larger Slepian basis (i.e., larger order $n$) might not lead to a better accuracy (see the curve corresponding to $n = 26$). Meanwhile, for Deep Learning, the benefit of increasing $L$ for accuracy becomes limited after $L \geq 6$. In summary, dimensionality appears to play a crucial role when comparing Least Squares and Deep Learning. Our preliminary results suggest that the performance difference between the two reconstruction methods could increase in dimension three and higher.

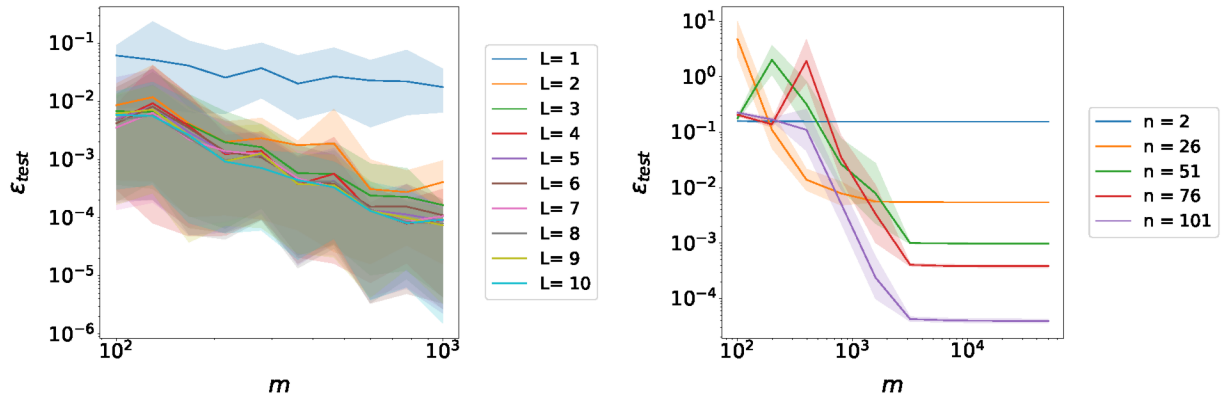


Figure 5.3: Comparison of reconstruction methods for $f_2(x) = \cos(0.2x)\cos(0.2y)e^{-\pi(x^2+y^2)}$ from samples. The left panel shows results using Deep Learning, while the right panel shows results using Least Squares. Here, $L$ denotes the number of layers in each NN, and $n$ denotes the order of the hyperbolic cross $\Lambda^{\mathrm{HC}}_{n-1}$.

# Bibliography


[1] B. Adcock, S. Brugiapaglia, N. Dexter, and S. Moraga. Deep neural networks are effective at learning high-dimensional hilbert-valued functions from limited data. *arXiv preprint arXiv:2012.06081*, 2020.

[2] B. Adcock, S. Brugiapaglia, and C. G. Webster. *Sparse polynomial approximation of high-dimensional functions*, volume 25. SIAM, 2022.

[3] B. Adcock and N. Dexter. The gap between theory and practice in function approximation with deep neural networks. *SIAM Journal on Mathematics of Data Science*, 3(2):624–655, 2021.

[4] G. S. Alberti and M. Santacesaria. Infinite-dimensional inverse problems with finite measurements. *Archive for Rational Mechanics and Analysis*, 243(1):1–31, 2022.

[5] M. T Ali, Y. A Elnakieb, A. Shalaby, A. Mahmoud, A. Switala, M. Ghazal, A. Khelifi, L. Fraiwan, G. Barnes, and A. El-Baz. Autism classification using smri: A recursive features selection based on sampling from multi-level high dimensional spaces. pages 267–270, 2021.

[6] U. Anders and O. Korn. Model selection in neural networks. *Neural networks*, 12(2):309–323, 1999.

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

[8] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen. On instabilities of deep learning in image reconstruction-does AI come at a cost? *arXiv preprint arXiv:1902.05300*, 2019.

[9] K. Armanious, C. Jiang, S. Abdulatif, T. Küstner, S. Gatidis, and B. Yang. Unsupervised medical image translation using cycle-MedGAN. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[10] S. Arridge, P. Maass, O. Öktem, and C.B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[11] V. Bacchelli and S. Vessella. Lipschitz stability for a stationary 2d inverse problem with unknown polygonal boundary. *Inverse problems*, 22(5):1627, 2006.

[12] F. Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

[13] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

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

[15] H. Bateman and A. Erdélyi. Higher transcendental functions, volume ii. *Bateman Manuscript Project) Mc Graw-Hill Book Company*, 1953.

[16] H. Bauer. *Measure and integration theory*, volume 26. Walter de Gruyter, 2001.

[17] R. Bellman. Dynamic programming. *science*, 153(3731):34–37, 1966.

[18] J. J. Benedetto and P. J. S. G. Ferreira. *Modern Sampling Theory: Mathematics and Applications*. Springer Science & Business Media, 2012.

[19] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[20] A. Bonami and A. Karoui. Uniform bounds of prolate spheroidal wave functions and eigenvalues decay. *Comptes Rendus. Mathématique*, 352(3):229–234, 2014.

[21] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[22] J. P. Boyd. Prolate spheroidal wavefunctions as an alternative to chebyshev and legendre polynomials for spectral element and pseudospectral algorithms. *Journal of Computational Physics*, 199(2):688–716, 2004.

[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[24] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*, volume 2. Springer, 2011.

[25] S. Brugiapaglia, N. Dexter, S. Karam, and W. Wang. Physics-informed deep learning and compressive collocation for high-dimensional diffusion-reaction equations: practical existence theory and numerics. *arXiv preprint arXiv:2406.01539*, 2024.

[26] S. L. Brunton and J. N. Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[27] S. Bubeck, V. Chadrasekaran, R. Eldan, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. arXiv preprint arXiv:2303.12712.

[28] L. Budach, M. Feuerpfeil, N Ihde, A. Nathansen, N. Noack, H. Patzlaff, F. Naumann, and H. Harmouch. The effects of data quality on machine learning performance. *Information Systems*, 132:102549, 2025.

[29] D. Cabric, S. M. Mishra, and R. W. Brodersen. Implementation issues in spectrum sensing for cognitive radios. In *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, volume 1, pages 772–776. Ieee, 2004.

[30] A. Caragea, P. Petersen, and F. Voigtlaender. Neural network approximation and estimation of classifiers with classification boundary in a Barron class. *arXiv preprint arXiv:2011.09363*, 2020.

[31] M. Chen, H. Jiang, W. Liao, and T. Zhao. Efficient approximation of deep ReLU networks for functions on low dimensional manifolds. *Advances in Neural Information Processing Systems*, 32:8174–8184, 2019.

[32] M Chen, H. Jiang, W. Liao, and T. Zhao. Nonparametric regression on low-dimensional manifolds using deep relu networks: Function approximation and statistical recovery. *Information and Inference: A Journal of the IMA*, 11(4):1203–1253, 2022.

[33] R. Chen and B. Cantrell. Highly bandlimited radar signals. In *Proceedings of the 2002 IEEE Radar Conference (IEEE Cat. No. 02CH37322)*, pages 220–226. IEEE, 2002.

[34] A. Chernov and D. Dũng. New explicit-in-dimension estimates for the cardinality of high-dimensional hyperbolic crosses and approximation of functions having mixed smoothness. *Journal of Complexity*, 32(1):92–121, 2016.

[35] A. Chkifa, A. Cohen, G. Migliorati, F. Nobile, and R. Tempone. Discrete least squares polynomial approximation with random evaluations- application to parametric and stochastic elliptic pdes. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 49(3):815–837, 2015.

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

[37] C. K. Chui and H. Mhaskar. Deep nets for local manifold learning. *Frontiers in Applied Mathematics and Statistics*, 4:12, 2018.

[38] A. Cloninger and T. Klock. A deep network construction that adapts to intrinsic dimensionality beyond the domain. *Neural Networks*, 141:404–419, 2021.

[39] A. Cohen, M. A Davenport, and D. Leviatan. On the stability and accuracy of least squares approximations. *Foundations of computational mathematics*, 13:819–834, 2013.

[40] M. Crawford, T. M. Khoshgoftaar, A. N. Prusa, J. D.and Richter, and H. Al Najada. Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 2:1–24, 2015.

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

[42] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[43] J. Daws and C. Webster. Analysis of deep neural networks with quasi-optimal polynomial approximation rates. *arXiv preprint arXiv:1912.02302*, 2019.

[44] T. De Ryck, S. Lanthaler, and S. Mishra. On the approximation of functions by tanh neural networks. *Neural Networks*, 143:732–750, 2021.

[45] F. Dinuzzo and B. Schölkopf. The representer theorem for hilbert spaces: a necessary and sufficient condition. *Advances in neural information processing systems*, 25, 2012.

[46] D. Dobrev. A definition of artificial intelligence. *arXiv preprint arXiv:1210.1568*, 2012.

[47] Y. Duan, J. S Edwards, and Y. K Dwivedi. Artificial intelligence for decision making in the era of big data–evolution, challenges and research agenda. *International journal of information management*, 48:63–71, 2019.

[48] J. C. Duchi. Introductory lectures on stochastic optimization. *The mathematics of data*, 25:99–186, 2018.

[49] W. E and S. Wojtowytsch. Representation formulas and pointwise properties for Barron functions. *arXiv preprint arXiv:2006.05982*, 2020.

[50] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[51] G. B. Folland. *Real analysis: modern techniques and their applications*, volume 40. John Wiley & Sons, 1999.

[52] N. R. Franco and S. Brugiapaglia. A practical existence theorem for reduced order models based on convolutional autoencoders. *Foundations of Data Science*, 7(1):72–98, 2025.

[53] G. Garrigos and R. M. Gower. Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*, 2023.

[54] N. M. Gottschling, V. Antun, B. Adcock, and A. C. Hansen. The troublesome kernel: why deep learning for inverse problems is typically unstable. *arXiv preprint arXiv:2001.01258*, 2020.

[55] F. A. Graybill and G. Marsaglia. Idempotent matrices and quadratic forms in the general linear hypothesis. *The Annals of Mathematical Statistics*, 28(3):678–686, 1957.

[56] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

[57] K. Gröchenig. *Foundations of time-frequency analysis*. Springer Science & Business Media, 2001.

[58] P. Grohs and F. Voigtlaender. Proof of the theory-to-practice gap in deep learning via sampling complexity bounds for neural network approximation spaces. *arXiv preprint arXiv:2104.02746*, 2021.

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

[60] S. Hayou, A. Doucet, and J. Rousseau. On the impact of the activation function on deep neural networks training. In *International conference on machine learning*, pages 2672–2680. PMLR, 2019.

[61] J. He, L. Li, J. Xu, and C. Zheng. ReLU deep neural networks and linear finite elements. *J. Comput. Math.*, 38:502–527, 2020.

[62] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

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

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

[65] Y. Huang, T. Würfl, K. Breininger, L. Liu, G. Lauritsch, and A. Maier. Some investigations on robustness of deep learning in limited angle tomography. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 145–153. Springer, 2018.

[66] C. M. Hyun, S. H. Baek, M. Lee, S. M. Lee, and J. K. Seo. Deep learning-based solvability of underdetermined inverse problems in medical imaging. *Medical Image Analysis*, 69:101967, 2021.

[67] M. Imaizumi and K. Fukumizu. Deep neural networks learn non-smooth functions effectively. In *The 22nd international conference on artificial intelligence and statistics*, pages 869–878. PMLR, 2019.

[68] M. Imaizumi and K. Fukumizu. Advantage of deep neural networks for estimating functions with singularity on hypersurfaces. *The Journal of Machine Learning Research*, 23(1):4772–4825, 2022.

[69] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[70] B. Irie and S. Miyake. Capabilities of three-layered perceptrons. In *ICNN*, pages 641–648, 1988.

[71] D. Jayswal, B. Y Panchal, B. Patel, N. Acharya, R. Nayak, and P. Goel. Study and develop a convolutional neural network for mnist handwritten digit classification. In *Proceedings of Third International Conference on Computing, Communications, and Cyber-Security: IC4S 2021*, pages 407–416. Springer, 2022.

[72] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[73] B. Kaltenbacher, A. Neubauer, and O. Scherzer. Iterative regularization methods for nonlinear ill-posed problems. In *Iterative Regularization Methods for Nonlinear Ill-Posed Problems*. de Gruyter, 2008.

[74] A. Khan, A. Laghari, and S. Awan. Machine learning in computer vision: A review. *EAI Endorsed Transactions on Scalable Information Systems*, 8(32), 2021.

[75] G. Khromov and S. P. Singh. Some fundamental aspects about lipschitz continuity of neural network functions. In *ICLR 2024*, 2024.

[76] Y. Kim, I. Ohn, and D. Kim. Fast convergence rates of deep neural networks for classification. *arXiv preprint arXiv:1812.03599*, 2018.

[77] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[78] P. Kirichenko, P. Izmailov, and A. G. Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

[79] A. Kirsch. *An introduction to the mathematical theory of inverse problems*, volume 120. Springer, 2011.

[80] M. Kohler, A. Krzyzak, and S. Langer. Estimation of a function of low local dimensionality by deep neural networks. *arXiv preprint arXiv:1908.11140*, 2019.

[81] M. Kontak and V. Michel. A greedy algorithm for nonlinear inverse problems with an application to nonlinear inverse gravimetry. *GEM-International Journal on Geomathematics*, 9(2):167–198, 2018.

[82] M. Köppen. The curse of dimensionality. In *5th online world conference on soft computing in industrial applications (WSC5)*, volume 1, pages 4–8, 2000.

[83] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

[84] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[85] M. Kubat. Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7. *The Knowledge Engineering Review*, 13(4):409–412, 1999.

[86] G. Lan. *First-order and stochastic optimization methods for machine learning*, volume 1. Springer, 2020.

[87] H. J. Landau and H. O. Pollak. Prolate spheroidal wave functions, fourier analysis and uncertainty—ii. *Bell System Technical Journal*, 40(1):65–84, 1961.

[88] H. J. Landau and H. O. Pollak. Prolate spheroidal wave functions, fourier analysis and uncertainty—iii: the dimension of the space of essentially time-and band-limited signals. *Bell System Technical Journal*, 41(4):1295–1336, 1962.

[89] H J. Landau and H. Widom. Eigenvalue distribution of time and frequency limiting. *Journal of Mathematical Analysis and Applications*, 77(2):469–481, 1980.

[90] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[91] J. M. Lee. Smooth manifolds. In *Introduction to smooth manifolds*, pages 1–31. Springer, 2013.

[92] K. Lee and K. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *arXiv preprint arXiv:1812.08373*, 2018.

[93] A. F. Lerma-Pineda, P. Petersen, S. Frieder, and T. Lukasiewicz. Dimension-independent learning rates for high-dimensional classification problems. *arXiv preprint arXiv:2409.17991*, 2024.

[94] C. K. Leung, R. K. MacKinnon, and Y. Wang. A machine learning approach for stock price prediction. In *Proceedings of the 18th international database engineering & applications symposium*, pages 274–277, 2014.

[95] Q. Li, T. Lin, and Z. Shen. Deep learning via dynamical systems: An approximation perspective. *Journal of the European Mathematical Society*, 2022.

[96] A. L. Maas, A. Y Hannun, A. Y. Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.

[97] T. T. Marinov and A. S. Vatsala. Inverse problem for coefficient identification in the Euler–Bernoulli equation. *Computers & Mathematics with Applications*, 56(2):400–410, 2008.

[98] W. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5:115–133, 1943.

[99] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[100] E. J. McShane. Extension of range of functions. *Bulletin of the American Mathematical Society*, 40(12):837–842, 1934.

[101] H. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural Comput.*, 8(1):164–177, 1996.

[102] H. N. Mhaskar. Eignets for function approximation on manifolds. *Applied and Computational Harmonic Analysis*, 29(1):63–87, 2010.

[103] R. K. Miller. Volterra integral equations in a Banach space. *Funkcial. Ekvac*, 18(2):163–193, 1975.

[104] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2012.

[105] I. C. Moore and M. Cada. Prolate spheroidal wave functions, an introduction to the slepian series and its properties. *Applied and Computational Harmonic Analysis*, 16(3):208–230, 2004.

[106] A. Mukherjee, A. Basu, R. Arora, and P. Mianjy. Understanding deep neural networks with rectified linear units. *International Conference on Learning Representations*, 2018.

[107] J. R. Munkres. *Topology*, volume 2. Prentice Hall Upper Saddle River, 2000.

[108] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[109] R. Nakada and M. Imaizumi. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *Journal of Machine Learning Research*, 21(174):1–38, 2020.

[110] N. Nandhakumar and J. Aggarwal. The artificial intelligence approach to pattern recognition—a perspective and an overview. *Pattern Recognition*, 18(6):383–389, 1985.

[111] F. Natterer. Regularisierung schlecht gestellter Probleme durch Projektionsverfahren. *Numerische Mathematik*, 28(3):329–341, 1977.

[112] F. Natterer. *The mathematics of computerized tomography*. SIAM, 2001.

[113] F. Natterer and F. Wübbeling. *Mathematical methods in image reconstruction*. SIAM, 2001.

[114] Y. Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[115] A M. Neuman, A. F. Lerma-Pineda, J. J. Bramburger, and S. Brugiapaglia. Reconstruction of frequency-localized functions from pointwise samples via least squares and deep learning. *arXiv preprint arXiv:2502.09794*, 2025.

[116] S. Niwas and D.C. Singhal. Aquifer transmissivity of porous media from resistivity data. *Journal of Hydrology*, 82(1-2):143–153, 1985.

[117] G. Ongie, A. Jalal, C. A Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

[118] J. A. A. Opschoor, P. C. Petersen, and Ch. Schwab. Deep ReLU networks and high-order finite element methods. *Analysis and Applications*, 18(05):715–770, 2020.

[119] Joost AA Opschoor, Ch Schwab, and Jakob Zech. Exponential relu dnn expression of holomorphic maps in high dimension. *Constructive Approximation*, 55(1):537–582, 2022.

[120] A. Osipov, V. Rokhlin, H. Xiao, et al. Prolate spheroidal wave functions of order zero. *Springer Ser. Appl. Math. Sci*, 187, 2013.

[121] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[122] R. Parhi and R. D. Nowak. Banach space representer theorems for neural networks and ridge splines. *The Journal of Machine Learning Research*, 22(1):1960–1999, 2021.

[123] R. Parhi and R. D. Nowak. Near-minimax optimal estimation with shallow ReLU neural networks. *IEEE Transactions on Information Theory*, 2022.

[124] R. Parhi and R. D. Nowak. What kinds of functions do deep neural networks learn? insights from variational spline theory. *SIAM Journal on Mathematics of Data Science*, 4(2):464–489, 2022.

[125] R. Parhi and R. D. Nowak. Deep learning meets sparse regularization: A signal processing perspective. *IEEE Signal Processing Magazine*, 40(6):63–74, 2023.

[126] R. L. Pego. Compactness in $l^2$ and the fourier transform. *Proceedings of the American Mathematical Society*, 95(2):252–254, 1985.

[127] P. Petersen, M. Raslan, and F. Voigtlaender. Topological properties of the set of functions generated by neural networks of fixed size. *Foundations of computational mathematics*, 21:375–444, 2021.

[128] P. Petersen and F. Voigtlaender. Optimal learning of high-dimensional classification problems using deep neural networks. *arXiv preprint arXiv:2112.12555*, 2021.

[129] P. Petersen and J. Zech. Mathematical theory of deep learning. *arXiv preprint arXiv:2407.18384*, 2024.

[130] P. C. Petersen. Neural network theory. *University of Vienna*, 2020.

[131] P. C. Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 180:296–330, 2018.

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

[133] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8:143–195, 1999.

[134] R. Plato and G. Vainikko. On the regularization of projection methods for solving ill-posed problems. *Numerische Mathematik*, 57(1):63–79, 1990.

[135] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[136] M. Razzaghi and Y. Ordokhani. Solution of nonlinear Volterra-Hammerstein integral equations via rationalized Haar functions. *Mathematical Problems in Engineering*, 7(2):205–219, 2001.

[137] L. Rondi. A remark on a paper by Alessandrini and Vessella. *Advances in Applied Mathematics*, 36(1):67–69, 2006.

[138] K. H. Rosen and K. Krithivasan. *Discrete Mathematics and its Applications*. McGraw-Hill New York, 1999.

[139] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[140] S. T Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[141] S. J Russell and P. Norvig. *Artificial intelligence: a modern approach*. pearson, 2016.

[142] J. Schmidt-Hieber. Deep ReLU network approximation of functions on a manifold. *arXiv preprint arXiv:1908.00695*, 2019.

[143] J. Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020.

[144] B. Sepehrian and M. Razzaghi. Solution of nonlinear Volterra-Hammerstein integral equations via single-term Walsh series method. *Mathematical Problems in engineering*, 2005(5):547–554, 2005.

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

[146] H. Shao, E. Ma, M. Zhu, X. Deng, and S. Zhai. Mnist handwritten digit classification based on convolutional neural network with hyperparameter optimization. *Intelligent Automation & Soft Computing*, 36(3), 2023.

[147] S. Sharma, M. Bhatt, and P Sharma. Face recognition system using machine learning algorithm. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pages 1162–1168. IEEE, 2020.

[148] Yoel Shkolnisky, Mark Tygert, and Vladimir Rokhlin. Approximation of bandlimited functions. *Applied and Computational Harmonic Analysis*, 21(3):413–420, 2006.

[149] J. W. Siegel. Optimal approximation of zonoids and uniform approximation by shallow neural networks. *arXiv preprint arXiv:2307.15285*, 2023.

[150] F. J. Simons. *Slepian Functions and Their Use in Signal Estimation and Spectral Analysis*, pages 891–923. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[151] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

[153] D. Slepian. Prolate spheroidal wave functions, fourier analysis and uncertainty—iv: extensions to many dimensions; generalized prolate spheroidal functions. *Bell System Technical Journal*, 43(6):3009–3057, 1964.

[154] D. Slepian. Some asymptotic expansions for prolate spheroidal wave functions. *Journal of mathematics and physics*, 44(1-4):99–140, 1965.

[155] D. Slepian. Some comments on fourier analysis, uncertainty and modeling. *SIAM review*, 25(3):379–393, 1983.

[156] D. Slepian and H. O. Pollak. Prolate spheroidal wave functions, fourier analysis and uncertainty—i. *Bell System Technical Journal*, 40(1):43–63, 1961.

[157] N. Stromberg, R. Ayyagari, S. Koyejo, R. Nock, and L. Sankar. Enhancing robustness of last layer two-stage fair model corrections. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[158] J. Sun, H. Li, and Z. Xu. Deep admm-net for compressive sensing MRI. *Advances in neural information processing systems*, 29, 2016.

[159] K. S. Suslick. Encyclopedia of physical science and technology. *Sonoluminescence and sonochemistry, 3rd edn. Elsevier Science Ltd, Massachusetts*, pages 753–771, 2003.

[160] J. B Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[161] J. A Tropp, J. N Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk. Beyond nyquist: Efficient sampling of sparse bandlimited signals. *IEEE transactions on information theory*, 56(1):520–544, 2009.

[162] V. N. Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[163] N. Vaswani, A.and Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[164] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

[165] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[166] U. Von Luxburg and B. Schölkopf. Statistical learning theory: Models, concepts, and results. In *Handbook of the History of Logic*, volume 10, pages 651–706. Elsevier, 2011.

[167] G. G. Walter and X.A. Shen. Sampling with prolate spheroidal wave functions. *Sampling Theory in Signal and Image Processing*, 2:25–52, 2003.

[168] P. Wang. On defining artificial intelligence. *Journal of Artificial General Intelligence*, 10(2):1–37, 2019.

[169] S. Wojtowytsch and W. E. A priori estimates for classification problems using neural networks. *arXiv preprint arXiv:2009.13500*, 2020.

[170] H. Xiao, V. Rokhlin, and N. Yarvin. Prolate spheroidal wavefunctions, quadrature and interpolation. *Inverse problems*, 17(4):805, 2001.

[171] J. Xin, T. Yi, V. P. Yi, Pu J. Yu, and Z. A. A. Salam. Convolutional neural network for fashion images classification (fashion-mnist). *Journal of Applied Technology and Innovation*, 7(4):11, 2023.

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

[173] W. W-G Yeh. Review of parameter identification procedures in groundwater hydrology: The inverse problem. *Water Resources Research*, 22(2):95–108, 1986.

[174] A. I. Zayed. *Advances in Shannon's sampling theory*. Routledge, 2018.

[175] B. Zhu, J. Z Liu, S. F Cauley, B. R. Rosen, and M. S. Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.

[176] J.Y Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.