Operator Learning for Scientific Computing

Thesis by
Margaret K. Trautner

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2025
Defended May 23, 2025

Margaret K. Trautner
ORCID: 0000-0001-9937-8393

ACKNOWLEDGEMENTS

My Ph.D. would not have been possible without many individuals. First, thank you to my advisor Andrew Stuart for his endless support, patience, kindness, and mentorship. There have been many moments during the past five years when I have been struck by how lucky I was to be his student. I cannot imagine having had a better advisor.

Thank you to my thesis committee members Kaushik Bhattacharya, Franca Hoffmann, and Houman Owhadi for their insights and support of my research. Discussions with Kaushik Bhattacharya in the first few years of my Ph.D. greatly shaped my research interests.

I am grateful to have been supported by the Department of Energy Computational Graduate Fellowship for four years. This fellowship allowed me flexibility as a researcher and served to broaden my view of applied and computational mathematics.

I thank all of my excellent collaborators and mentors, especially Nicholas Nelsen and Samuel Lanthaler, both of whom have influenced me greatly. Thank you to Dr. Metcalfe and Dr. Raulston, who introduced me to applied mathematics nine years ago.

Research is all the more enjoyable when surrounded by great friends, and I have had wonderful friend/colleagues in CMS. Pau, Eitan, Lauren, Chris, Laure, and Matthieu; thanks for being great colleagues, conference companions, and friends. The support of current and former CMS staff, especially Jolene Brink, Diana Bohler, and Bonnie Leung has been vital.

To my family for supporting me in my interests from the beginning, especially my parents and grandparents. To my husband John; I'm grateful every day that Caltech brought us together, and the stress tensor that you used as a conversation starter back in 2021 may be found in Chapter 2. Finally, to my daughter Laura; as a teething infant, you weren't exactly helpful while writing this thesis, but I wouldn't have it any other way. It's all for you now.

ABSTRACT

This thesis develops operator learning theory and methods for use in scientific computing. Operator learning uses data to approximate maps between infinite dimensional function spaces. As such, operator learning provides a natural framework for using machine learning in applications with partial differential equations (PDEs). While operator learning architectures have successfully modeled a variety of physical phenomena in practice, the theoretical foundations underpinning these successes remain in early stages of development.

The present work takes a step towards a complete understanding of operator learning and its potential use in scientific applications. The thesis begins by studying multiscale constitutive modeling, where operator learning models can serve as surrogates to accelerate simulation and aid in model discovery of physical laws. The work proposes, and theoretically and numerically analyzes, an operator learning architecture for modeling history dependence in homogenized constitutive equations. The thesis then addresses learning solutions to an elliptic PDE in the presence of discontinuities and corner interfaces in two-dimensional materials. By proving a key continuity result for the underlying PDE, a universal approximation result is obtained. In its second half, the thesis moves on from the setting of homogenized constitutive laws and gives insight to operator learning from a broader perspective. First, error analysis bounds a form of discretization error that arises in implementations of the Fourier Neural Operator (FNO). Next, a modified form of the FNO, the Fourier Neural Mapping, accommodates finite-dimensional data while retaining the underlying function space structure. This modification allows applications where the map of interest is governed by an infinite-dimensional operator with data, such as parameters or summary statistics, in the form of finite vectors. Finally, the thesis extends a theory-to-practice gap result in finite dimensions to the infinite-dimensional operator learning setting, asserting that even for classes of architectures whose model expressivity scales well with model size, their error convergence with respect to data size scales poorly. In summary, this thesis builds understanding of operator learning from several perspectives and contributes both theoretical advancements and practical methodologies that improve the applicability of operator learning models to scientific problems.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Kaushik Bhattacharya, Burigede Liu, Andrew M. Stuart, and Margaret Trautner. "Learning Markovian homogenized models in viscoelasticity". In: *Multiscale Modeling & Simulation* 21.2 (2023), pp. 641–679. DOI: 10.1137/22M149920.

M.T. was the lead author on this work. M.T. proved the results in this work except for the limit obtained in Appendix B.1 of the publication, which was proved by A.S. M.T. wrote the code and performed all numerical experiments and wrote the manuscript except for the mechanics portion of the literature review, which was written by K.B. The work in this paper makes up the bulk of Chapter 2.

[2] Burigede Liu, Eric Ocegueda, Margaret Trautner, Andrew M. Stuart, and Kaushik Bhattacharya. "Learning macroscopic internal variables and history dependence from microscopic models". In: *Journal of the Mechanics and Physics of Solids* 178 (2023), p. 105329. DOI: 10.1016/j.jmps.2023.105329.

This work was led by B.L., and E.O. performed the internal variable experiments of Figure 3. This publication and the prior one on viscoelasticity originated as one collaboration, which split into the prior work as the theoretical justification and this work as the numerical experiments in more complex applications. M.T. contributed to the computational methodology underpinning this work. A result from the one-dimensional elasto-viscoplastic experiment in this publication is included in Chapter 2 to demonstrate applicability of the theory developed there in more complex settings.

[3] Kaushik Bhattacharya, Nikola B. Kovachki, Aakila Rajan, Andrew M. Stuart, and Margaret Trautner. "Learning homogenization for elliptic operators". In: *SIAM Journal on Numerical Analysis* 62.4 (2024), pp. 1844–1873. DOI: 10.1137/23M1585015.

M.T. was the lead author on this work. M.T. proved the theoretical results apart from Proposition 1.2 and its contributing lemmas, which were proved by N.B.K. M.T. designed and performed all numerical experiments and wrote the manuscript aside from the appendix content supporting Proposition 1.2, which N.B.K. wrote. This publication forms Chapter 3.

[4] Samuel Lanthaler, Andrew M. Stuart, and Margaret Trautner. *Discretization error of Fourier neural operators*. 2024. arXiv: 2405.02221 [math.NA].

M.T. was the lead author on this work. A.S. conceptualized this project. M.T. proved all results except Theorem 3.3. and Appendix A, which were proved and written by S.L. M.T. wrote the code and performed all numerical experiments except the adaptive subampling algorithm, which was written and done by S.L. M.T. wrote the remainder of the manuscript. This work is adapted for Chapter 4.

[5] Kaushik Bhattacharya, Lianghao Cao, George Stepaniants, Andrew Stuart, and Margaret Trautner. *Learning Memory and Material Dependent Constitutive Laws*. 2025. arXiv: 2502.05463 [math.NA].

This work was led by G.S. and L.C. M.T. participated in the conceptualization of this work and discussions of both theory and numerical experiments. The theory component was led by G.S., and the numerical experiments and code were done and written by L.C. M.T. contributed to the analysis and design of the RNO-FNM architecture. Some results from this paper are included in Chapter 2 as they are natural extentions of the prior work on operator learning constitutive modeling.

[6] Philipp Grohs, Samuel Lanthaler, and Margaret Trautner. *Theory to Practice Gap for Neural Networks and Neural Operators*. 2025. arXiv: 2503.18219 [cs.LG].

S.L. conceptualized this project and wrote and proved results in Section 2. S.L and M.T. jointly wrote and proved results in Section 3 and the associated results in the appendix except for Lemma 3.14, proved by P.G. S.L. wrote the introduction and conclusion. This work is adapted for Chapter 6.

[7] Daniel Zhengyu Huang, Nicholas H. Nelsen, and Margaret Trautner. "An operator learning perspective on parameter-to-observable maps". In: *Foundations of Data Science* 7.1 (2025), pp. 163–225. DOI: 10.3934/fods.2024037.

N.H.N. conceptualized the project. M.T. wrote Sections 1 and 2 and performed the homogenization experiment. M.T. and N.H.N both contributed to Section 3 and the universal approximation theory proofs. N.H.N. wrote Section 4 and proved the results therein. N.H.N. also implemented the Fourier Neural Mapping architecture. D.Z.H. performed the airfoil experiment. All authors edited the manuscript. This work is adapted for Chapter 5, excluding the theoretical results on linear functional regression in Section 4 of the publication.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Number *Page*

Chapter 1

INTRODUCTION

Before the advent of computers, scientific data were useful for building models only to the extent that they could inform scientific theory or contribute to statistics. While limited by human computational ability, these use cases had the advantage of complete interpretability. As computers emerged and became increasingly more efficient at data processing, additional types of data-driven scientific models were made realistic, including ensemble methods [1], various classification algorithms [2], and additional statistical methods like the expectation-maximization algorithm [3]. These methods retained significant interpretability while allowing large amounts of data to inform models. However, they also tended to be problem specific and required some knowledge of the underlying problem to form an accurate model. Neural networks changed this paradigm; superpositions of nonlinear activations and affine maps could approximate *any* continuous map [4] between finite spaces at the cost of interpretability of the model. Although finding the optimal parameterization of such a network for a particular map is NP-hard, gradient descent methods turned out to be effective at finding network parameters that achieve good approximations of the optimal map despite dramatic non-convexity of the optimization landscape. Machine learning methods proved to be extremely successful at completing tasks that no prior models could, including image classification [5], speech recognition [6], and superhuman performance in games [7].

Scientific computing developed independently from machine learning as a body of computational tools to model phenomena in physics, chemistry, biology, and engineering. A large share of these phenomena are described by partial differential equations (PDEs) that specify the time and space evolution of functions in infinite-dimensional function space. The first attempts to use machine learning to approximate the behavior of PDEs did so by first discretizing functions to bring them down to the native finite-dimensional space of neural networks [8, 9]. However, this approach leads to significant drawbacks. In addition to making it more difficult to apply existing knowledge of PDE theory to machine learned models of PDEs, discretizing before learning fixes a single

discretization into the model itself, and new data of a different discretization becomes incompatible with the model. Furthermore, the model can overfit to a particular discretization. Operator learning addresses this problem by building architectures that map between infinite dimensional function spaces. Operator learning models are discretization-independent in the sense that no discretization size is built into the model, and any discretization of the underlying functions may be used with the model. Thus, operator learning models are natural for approximating maps that arise from partial differential equations.

Although machine learning methods are proving successful at modeling scientific phenomena, theory lags behind effective application. Efforts to interpret the effectiveness of machine learning models include universal approximation results [10], error bounds in terms of the model size or number of data points [11], and characterizations of the optimization landscape [12]. Other efforts have built in interpretable features to models such as additional constraints in the objective function [9], extraction of latent variables via autoencoder compression [13], or symmetry-enforcing components [14]. In scientific computing applications, there is opportunity both for existing scientific knowledge to help understand machine learning models and for the models to inform scientific knowledge. One way the latter occurs is through model discovery, where training a model can help identify interpretable theory for the underlying phenomena. Another application is surrogate modeling, where a learned model can perform one aspect of simulation very quickly, thus accelerating large-scale computations. As these approaches continue to evolve, the interplay between machine learning and science promises to accelerate discovery while simultaneously improving model interpretability and rigorous theoretical foundations.

This thesis makes a contribution towards understanding machine learning for scientific phenomena described by PDEs. The work exhibited here falls into two categories. The first category spotlights the application area of surrogate modeling for constitutive laws in multiscale materials. The second category investigates operator learning theory more generally and includes error bounds in terms of discretization, data size, and model size in various settings. The remainder of this introduction details each of these categories separately before giving an outline of the thesis.

1.1 Learning Homogenized Constitutive Models

Many materials in solid mechanics have dynamics governed by complex interactions across multiple scales. For instance, a material may have rapidly varying material properties on a small scale, but the dynamics of interest take place on a much larger scale. Multiscale modeling is a framework that has emerged to understand this complexity by assuming a hierarchy of scales with sufficient separation between adjacent pairs of scales. Dynamics may be computed iteratively by resolving force balance laws on each scale separately and exchanging the result between scales. Homogenization theory provides one method to exchange such information. Homogenization assumes a periodic or statistically regular microstructure and first analyzes the behavior within a characteristic piece of the material called a representative volume element. Homogenization then yields a map from the average material strain, or displacement gradient, to the average stress over a representative volume element. The map from averaged strain to averaged stress is called the *homogenized constitutive law*. This approach avoids having to resolve physical laws on the microscale by averaging out the dependence on the fine scale material properties. The drawback of this approach is that the homogenized map is often difficult to obtain in practice. In the case that both the material microstructure and the multiscale constitutive law are known, the homogenized map may not have a known explicit form and may require numerically solving a *cell problem* PDE each time microscale dynamics need to be resolved. In the case that the material microstructure of the multiscale constitutive law are not known, the homogenized behavior may only be approximated from experimental data. Both of these settings are ripe for the application of machine learning. In the first setting, machine learning can deliver surrogate models for the homogenized map. By training on data from a number of numerically solved cell problems on the microstructure, a surrogate model can approximate the homogenized map in a computationally efficient manner and facilitate efficient macroscale simulations. In the second setting, a surrogate model can also be obtained from experimental data, and this model can contribute to model discovery for the underlying physics. With these use cases in mind, this thesis explores novel operator learning architectures for learning homogenized constitutive laws, establishes rigorous theory underpinning their use, and shows their effectiveness in a multitude of numerical experiments.

Chapter 2 focuses on learning homogenized constitutive models that have his-

tory dependence. In some settings, such as in plastic materials, history dependence is present in the multiscale constitutive laws even before the equations are homogenized. In order to predict the stress this material will experience, one needs to know the entire material strain history. Other materials acquire history dependence through homogenization: Kelvin-Voigt (KV) viscoelastic materials are an example. The bulk of Chapter 2 is based on [15], published in *Multiscale Modeling & Simulation*, Vol. 21, Iss. 2 (2023), that explores learning homogenized models in this KV viscoelastic setting. This work and the companion paper [16], published in *Journal of the Mechanics and Physics of Solids*, Vol. 178 (2023), present a recurrent neural operator (RNO) architecture as a proposed surrogate model to capture history dependence in a Markovian manner, thereby avoiding the computational expense of accounting for the entire strain history at every time step in a simulation. The RNO model incorporates history dependence through a fixed number of *internal variables* that are updated via a numerical time stepping method. The linear setting of one-dimensional KV viscoelasticity allows for a complete analysis. In particular, it is proven that the RNO architecture can exactly capture the constitutive law in the case of a piecewise-constant material, and the architecture can approximate the law for piecewise-continuous materials to an arbitrary degree of accuracy. Numerical experiments demonstrate the empirical ability to find such an approximating model in practice. Although the theory only applies in the setting of one-dimensional KV viscoelasticity, the companion paper [16] shows that it is effective at modeling more complex materials including elasto-viscoplastic composites in two dimensions and elasto-viscoplastic polycrystals in three dimensions. The chapter includes an elasto-viscoplastic experiment in one dimension to support the use of the model outside the setting where the theory applies directly. Indeed, the motivation for developing the method is to deploy it in complex constitutive settings that are difficult to solve numerically and analyze. In these experiments, the RNO also facilitates model discovery by giving insight into how many internal variables are needed for a good approximation. Internal variable theory is well established in computational mechanics independent from data-driven methods [17]. For each experiment, the discretization-invariance properties of the model are tested, and it is found that by giving the RNO inputs consistent with the true equations, the model is more robust to changes in discretization; thus, this analysis also contributes to model discovery. Finally, Chapter 2 includes an extension of the method

to include material dependence as a model input, which is developed fully in the paper [18], available as a preprint at *arXiv: 2502.05463 [math.NA]*. The architecture for this extension combines the RNO with an architecture developed in Chapter 5 called Fourier Neural Mappings (FNMs). Similar theory justifies the use of this extended architecture, and a numerical example from this extension is included in Chapter 2 to show that the material-dependent model is accurate in practice. Altogether, Chapter 2 synthesizes knowledge from three papers to present a thorough investigation of operator learning for constitutive models with history dependence.

Originally published in *SIAM Journal on Numerical Analysis*, Vol. 62, Iss. 4 (2024), Chapter 3 addresses learning homogenized constitutive laws for elliptic operators in the presence of discontinuous materials. This research was prompted in part by an attempt to extend the theory developed in Chapter 2 for one-dimensional viscoelasticity to higher dimensions. The analysis immediately runs into a challenge: the one-dimensional theory relies on an intermediate approximation of a piecewise-constant material, and the two-dimensional analog of piecewise-constant materials are checkerboards which contain discontinuities and corner interfaces. In addition to the theoretical motivation to address this problem, practical motivation is present as well. Discontinuous microstructures are a common application setting in constitutive modeling since material microstructures often take the form of grains that are modeled by Voronoi tessellations. These microstructure complexities affect the smoothness of the underlying equations, and many applications of scientific machine learning confine themselves to smooth coefficients and materials to avoid addressing this issue. In particular, universal approximation, often the starting point for theory, is threatened by a lack of regularity. Universal approximation results for operator learning require the map of interest to be continuous between separable input and output spaces. In the case of the cell problem for linear elasticity with discontinuous materials, only continuity results with an input space of L^∞ are obvious, and L^∞ is not separable. Nevertheless, Chapter 3 proves two such continuity results: one continuity result with an input space of L^2 and one Lipschitz continuity result with an input space of L^p for some p such that $2 < p < \infty$. These results are used to establish universal approximation in this setting. Furthermore, numerous experiments are done to compare learning with discontinuous microstructures to smooth microstructures, and it is found that, although the error for dis-

continuous microstructures is an order of magnitude higher in practice that for smooth microstructures, the operator learning model is still an accurate approximator. In summary, Chapter 3 addresses the challenge of learning in the presence of discontinuities for the setting of a linear elliptic PDE.

1.2 Error Bounds in Operator Learning

The second half of the thesis moves away from constitutive modeling to answer broader theoretical questions that go beyond specific applications. Chapters 4, 5, and 6 each address different sources of error in operator learning and are introduced separately in this subsection.

Available as a preprint at *arXiv: 2405.02221 [math.NA]*, Chapter 4 analyzes discretization error of a common operator learning architecture, namely, the Fourier Neural Operator (FNO). The FNO maps between function spaces by combining traditional neural network components of affine transformations and nonlinear activations with a kernel integral operator parameterized in the Fourier domain that allows for nonlocality in the model. The FNO as defined [19] includes an inner product taken over a continuum that is used to compute the Fourier transform. In both the definition and in theory developed for the model [20], it is assumed that this inner product is computed exactly. However, the FNO used in practice must approximate this inner product since functions on a continuum are discretized in numerical computations. Thus, the implemented FNO is a *different operator* than the one defined and analyzed in prior work. The aliasing error that originates from approximations of the Fourier transform then propagates through the nonlinear layers of the model. This work bounds this error in terms of the size N of the discretization used and finds that despite the nonlinear error propagation, the output error behaves like N^{-s} where s governs the Sobolev regularity of the input. This error behavior is also observed experimentally in both random and trained models. Some implications of this result are that smooth activations like GeLU should be used in the FNO instead of ReLU, and if positional information is encoded in the input, as is standard in FNO usage, the positional encoding should be periodic to maintain regularity. Knowledge of this error inspires an adaptive subsampling algorithm that refines the data discretization during training to speed up computation time. This algorithm is also explored in the chapter. The discretization error that arises in FNO implementation is bounded and analyzed, filling a gap in existing theory for the FNO.

Chapter 5 contains portions of work originally published in *Foundations of Data Science*, Vol. 7, Iss. 1, (2025) that propose and analyze an operator learning architecture called Fourier Neural Mappings (FNMs) that can accommodate both finite and infinite dimensional inputs and outputs. In many cases the map of interest involves a finite-dimensional parameter input or observable output, but the underlying map is defined implicitly through an infinite-dimensional operator like a PDE. For example, the map from a material microstructure to the effective coefficient in linear elasticity is a map from a function input to a finite vector output, but the effective coefficient is obtained via the solution to a cell problem PDE. Thus, the underlying map involves an infinite-dimensional operator, but the object of interest is the finite-dimensional effective coefficient. The FNM modifies the FNO by appending linear functional and linear decoder layers to map functions to finite vectors and finite vectors to functions, respectively. The architecture preserves desirable properties of the FNO such as universal approximation and discretization invariance. In the setting where both a full-field solution, such as the cell problem solution function, and the finite vector observable, such as the effective coefficient, are obtainable, a natural question is whether it is more data efficient to learn the observable directly or to learn the full-field solution and then compute the observable using known equations. Although the original publication [21] includes an analysis of this question from a statistical learning perspective in a linear Bayesian setting, in this thesis the presentation is confined to numerical exploration of this question in nonlinear settings. The accuracy of the learned approximation versus the number of training data is shown for three nonlinear application problems of advection-diffusion, aerodynamics, and constitutive modeling. In these experiments, the FNM architecture outperforms finite dimensional neural networks that do not take advantage of the continuum perspective. The model error versus the number of training data is explored and resulting error rates are computed.

Available as a preprint at *arXiv:2503.18219 [cs.LG]*, Chapter 6 examines sampling complexity of ReLU neural networks and neural operators. Error of a neural network model may be decomposed into an approximation error component and a generalization error component. The approximation error describes the error of the best possible parameterization of a fixed architecture when compared to the true map. The approximation error is closely related to model expressivity and parametric complexity. The generalization

error quantifies the difference between this best possible error and the error achieved in practice by optimizing some objective function over a finite number of data samples. Thus, the generalization error is closely related to the sampling complexity of the model. The *theory-to-practice gap* refers to the fact that despite having "good" parametric convergence rates for the approximation error, neural networks have "bad" sampling convergence rates for the generalization error. In order words, neural network models may be highly expressive for their size, but their generalization error converges slowly with the number of data samples, independent of the reconstruction algorithm used. To make this more precise, let $U^\alpha([0,1]^d)$ be the set of functions on $[0,1]^d$ which, for any $n \in \mathbb{N}$, can be approximated by a neural network ψ_n with at most n nonzero weights and with approximation error $\|f - \psi_n\|_{L^\infty} \leq n^{-\alpha}$. In this expression, α is the *parametric convergence rate.* Reconstruction algorithms attempt to attain a neural network approximation to $f \in U^\alpha([0,1]^d)$ from point samples $f(x_1), \ldots, f(x_N)$. The optimal *sampling convergence rate* β_* is the largest β such that there exists a reconstruction algorithm $A : f \mapsto Q(f(x_1), \ldots, f(x_N))$ for some mapping $Q : \mathbb{R}^N \to L^p([0,1]^d)$ with a guarantee of the form $\sup_{f \in U^\alpha} \|f - A(f)\|_{L^p} \leq CN^{-\beta}$. The parametric convergence rate α describes the rate in terms of the number of model parameters n, but the sampling convergence rate β_* describes the rate in terms of the number of samples N. In several classical reconstruction methods, such as polynomial reconstruction and some kernel methods, $\beta_* = \alpha$. Prior work proved in that for finite-dimensional neural networks, β_* remains uniformly bounded even in the limit $\alpha \to \infty$ [22]. This discrepancy between α and β_* is the theory-to-practice gap. Chapter 6 first improves the bound on β_* obtained in the prior work to show that in an L^p setting, $\beta_* \leq \frac{1}{p} + \frac{1}{d}$. As a second contribution, the chapter extends the theory-to-practice gap result for operator learning, showing that in the infinite-dimensional setting, the optimal convergence rate in a Bochner L^p norm is controlled by $\beta_* \leq \frac{1}{p}$, for $p \in [1, \infty)$. These results are shown to apply both to kernel integral neural operators, including the FNO, and to Deep Operator Networks (DeepONets) [23]. In light of the empirical ability of modern neural network optimization to find good parameterizations with limited data, these hardness results are somewhat surprising and invite further investigation.

1.3 FNO Definition

The FNO is referred to by several different chapters, and its definition is stated here to avoid repetition. This definition is referred to in Chapters 2, 3, 4, and 6. Note that Chapter 5 retains its own definition with slightly adjusted notation as a substantial contribution of that chapter is modifying the architecture in detail.

***Definition* 1.3.1** (Fourier Neural Operator)**.** Let $\mathcal{A}$ and $\mathcal{U}$ be two Banach spaces of real vector-valued functions over domain $\mathbb{T}^d$. Assume input functions $a \in \mathcal{A}$ are $\mathbb{R}^{d_a}$-valued while the output functions $u \in \mathcal{U}$ are $\mathbb{R}^{d_u}$-valued. The neural operator architecture $\Psi_\theta : \mathcal{A} \to \mathcal{U}$ is

$$\begin{aligned} \Psi_\theta &= \mathcal{Q} \circ \mathsf{L}_{T-1} \circ \cdots \circ \mathsf{L}_0 \circ \mathcal{P}, \\ v_{t+1} &= \mathsf{L}_t v_t = \sigma_t(W_t v_t + \mathcal{K}_t v_t + b_t), \quad t = 0, 1, \ldots, T-1, \end{aligned}$$

with $v_0 = \mathcal{P}(a)$. Here, $\mathcal{P} : \mathbb{R}^{d_a} \to \mathbb{R}^{d_0}$ and $\mathcal{Q} : \mathbb{R}^{d_T} \to \mathbb{R}^{d_u}$ are shallow neural networks with globally Lipschitz and C^∞ activations σ_p and σ_q, and the σ_t are fixed nonlinear activation functions acting locally as maps $\mathbb{R}^{d_{t+1}} \to \mathbb{R}^{d_{t+1}}$ in each layer. $\mathcal{P}$, $\mathcal{Q}$, and the σ_t are viewed as operators acting pointwise, or pointwise almost everywhere, over the domain $\mathbb{T}^d$), $W_t \in \mathbb{R}^{d_{t+1} \times d_t}$ are matrices, $\mathcal{K}_t : \{v_t : \mathbb{T}^d \to \mathbb{R}^{d_t}\} \to \{v_{t+1} : \mathbb{T}^d \to \mathbb{R}^{d_{t+1}}\}$ are integral kernel operators and $b_t : \mathbb{T}^d \to \mathbb{R}^{d_{t+1}}$ are constant bias functions. The activation functions σ_t are restricted to the set of globally Lipschitz, non-polynomial, C^∞ functions. The integral kernel operators $\mathcal{K}_t$ are parameterized in the Fourier domain in the following manner. Let $i = \sqrt{-1}$ denote the imaginary unit. Then, for each t, the kernel operator $\mathcal{K}_t$ is parameterized by

$$(\mathcal{K}_t v_t)(x) = \sum_{k \in [[K]]^d} \left(\sum_{j=1}^{d_t} (P_t^{(k)})_j \langle e^{2\pi i \langle k, \cdot \rangle}, (v_t)_j \rangle_{L^2(\mathbb{T}^d; \mathbb{C})} \right) \mathrm{e}^{2\pi i \langle k, x \rangle} \in \mathbb{R}^{d_{t+1}} . \tag{1.3.1}$$

Here, each $P_t^{(k)} \in \mathbb{C}^{d_{t+1} \times d_t}$ constitutes the learnable parameters of the integral operator, with $(P_t)_j^{(k)}$ the jth column, and $K \in \mathbb{Z}^+$ is a mode truncation parameter. $\mathcal{K}_t$ is well-defined for $v_t \in L^2(\mathbb{T}^d)$. We denote by θ the collection of parameters that specify Ψ_θ, which include the weights W_t, biases b_t, kernel weights P_t, and the parameters describing $\mathcal{P}$ and $\mathcal{Q}$. $\lozenge$

1.4 Thesis Outline

The remainder of the thesis is structured as follows. Chapter 2 introduces the RNO as a method to model history-dependence homogenized constitutive laws

and provides underpinning theory for the model. Chapter 3 addresses learning the homogenized cell problem solution in the technical setting of discontinuous materials with corner interfaces in two spatial dimensions. Chapter 4 departs from constitutive modeling and bounds the discretization error produced by implementations of the FNO. Chapter 5 proposes a modification of the FNO that can account for finite-dimensional inputs and outputs while taking advantage of the operator learning perspective and investigates the error as a function of the number of data using this model. Chapter 6 establishes hardness results in the form of a theory-to-practice gap for operator learning that points out a fundamental limit in the ability of neural operators to converge quickly with respect to the number of data samples.

As each chapter is adapted from different publications geared towards different audiences, the chapters are self-contained and establish separate notation.

Chapter 2

LEARNING HOMOGENIZED CONSTITUTIVE MODELS WITH MEMORY

This chapter synthesizes several papers on the topic of data-driven learning of multiscale constitutive laws in solid mechanics:

[1] Kaushik Bhattacharya, Burigede Liu, Andrew M. Stuart, and Margaret Trautner. "Learning Markovian homogenized models in viscoelasticity". In: *Multiscale Modeling & Simulation* 21.2 (2023), pp. 641–679. DOI: 10.1137/22M149920.

[2] Burigede Liu, Eric Ocegueda, Margaret Trautner, Andrew M. Stuart, and Kaushik Bhattacharya. "Learning macroscopic internal variables and history dependence from microscopic models". In: *Journal of the Mechanics and Physics of Solids* 178 (2023), p. 105329. DOI: 10.1016/j.jmps.2023.105329.

[3] Kaushik Bhattacharya, Lianghao Cao, George Stepaniants, Andrew Stuart, and Margaret Trautner. *Learning Memory and Material Dependent Constitutive Laws*. 2025. arXiv: 2502.05463 [math.NA].

Fully resolving dynamics of materials with rapidly varying features involves expensive fine-scale computations which need to be conducted on macroscopic scales. The theory of homogenization provides an approach to derive effective macroscopic equations that eliminate the small scales by exploiting scale separation. An accurate homogenized model avoids the computationally expensive task of numerically solving the underlying balance laws at a fine scale, thereby rendering a numerical solution of the balance laws more computationally tractable.

In complex settings, homogenization only defines the constitutive model implicitly, and machine learning can be used to learn the constitutive model explicitly from localized fine-scale simulations. This chapter presents work on this topic, combining results from three different papers, and a fourth paper under the same umbrella forms Chapter 3. The first paper addressed by this chapter [15] covers the case of one-dimensional viscoelasticity, where the linearity of the model allows for a complete analysis. This paper forms the

bulk of the chapter, although the exposition is altered significantly. In this work, it is established that a homogenized constitutive model may be approximated by a recurrent neural operator (RNO) architecture. The memory is encapsulated in the evolution of an appropriate finite set of hidden variables, which are discovered through the learning process and dependent on the history of the strain. The architecture is developed in a *discretization-invariant* manner, where the same model may be used on strain histories with differing time discretizations. Theory is developed for this model, and simulations for one-dimensional viscoelasticity and one-dimensional elasto-viscoplasticity are presented. A companion paper to [15], namely, [16], tests the method empirically and shows that it is an accurate and computationally efficient model for more complex constitutive laws as well, including two-dimensional elasto-viscoplastic laminates and three-dimensional elasto-viscoplastic polycrystals. Both these works seek to approximate the map from homogenized strain to homogenized stress for a fixed material using the RNO architecture, and the theory developed is confined to one dimension. In the final paper addressed by this chapter, [18], we unite knowledge from prior work to build data driven models of multiscale materials that incorporate both memory and material dependence. This paper combines the RNO architecture presented in [15] and [16] with the Fourier Neural Mapping (FNM) architecture developed and analyzed in Chapter 5 to allow the neural networks in the RNO to take both function-valued and finite vector-valued inputs. In this chapter, we include the merged RNO-FNM architecture used in [18] to add material dependence as well as a numerical experiment demonstrating the ability of the model architecture to learn both memory and material dependence in one-dimensional viscoelasticity. Altogether, this chapter gives a thorough narrative of our work on data-driven operator learning for multiscale constitutive models.

2.1 Introduction

The dynamics of materials are governed by complex interactions between different time and length scales. Multiscale modeling addresses this challenge by assuming a hierarchy of scales with sufficient scale separation, identifying behavior within each scale, and resolving the dynamics by pairwise interaction between adjacent scales. One method of this type is homogenization, which averages the smaller scale to achieve the relevant behavior on the larger scale. This chapter leverages homogenization to model multiscale materials in

continuum mechanics. Here, a two scale separation is of interest, where the physical constitutive laws are known only on the smaller scale, but the goal is to model macroscale behavior. Homogenization theory assumes a periodic microstructure so that the nature of the averaged dynamics found via homogenization apply to the entire macroscale material. By averaging the smaller scales in this way, macroscale dynamics may be computed more efficiently.

As an additional challenge, many materials in continuum mechanics lead to constitutive laws which are history dependent. This property may be inherent to physics beneath the continuum scale (for example in plasticity [24, 25]) or may arise from homogenization of rapidly varying continua [26, 27] (for example in the Kelvin-Voigt (KV) model of viscoelasticity [28]). In the latter, case, history dependence is not present in the underlying microscale physics but emerges as a consequence of homogenization. History dependence introduces computational barriers because the state of the material at every past time step may impact the current dynamics, growing in complexity with time. Thus, Markovian homogenized models are desirable for both interpretability and computability. In some cases theory may be used to justify Markovian models which capture this history dependence, but in many cases data plays a central role in finding such models. In this chapter, we assume a data-driven Markovian model for history dependence. We justify this assumption with theoretical underpinnings in the case of one-dimensional KV viscoelasticity and apply the method to more complex materials to show its effectiveness outside the setting where theory applies directly.

The paper [29] preceding the work of this chapter adopted a data-driven learning approach to uncovering history-dependent homogenized models arising in crystal plasticity. However, the resulting constitutive model is not causal and instead learns causality approximately from computations performed at the level of the cell problem. Instead, we introduce a different approach, learning *causal* constitutive models with a discretization-invariant model, which we call the RNO. In order to give rigorous underpinnings to our approach, we first study the methodology in the setting of linear one-dimensional viscoelasticity. Here we can use theoretical understanding to justify and validate the methodology; we show that machine-learned homogenized models can accurately approximate the dynamics of multiscale models at much cheaper evaluation cost. We obtain insight into desirable choice of training data to learn

the homogenized constitutive model, and we study the effect of the multiple local minimizers which appear in the underlying optimization problem. Furthermore, the rigorous underpinnings enable us to gain insight into how to test model hypotheses. We demonstrate that hypothesizing the correct model leads to robustness with respect to changes in time discretization in the causal model: the model can be trained at one time step and used at others, and the model can be trained with one time-integration method and used with others. In contrast, hypothesizing an incorrect model leads to intolerable sensitivity with respect to the time step. Thus training at one time step and testing at other levels of resolution provides a method for testing model form hypotheses. For viscoelasticity, we work primarily with the one-dimensional KV model for which the constitutive model depends only on strain and strain rate. We also touch on the standard linear solid (SLS) model for which the constitutive relation depends only on the strain and the strain history and perform numerical experiments in a one-dimensional elasto-viscoplastic material; in so doing we show that the ideas presented extend beyond the specifics of the one-dimensional KV setting. Although not contained in this chapter, the experiments of [16] demonstrate that the methodology is effective in modeling a variety of more complex materials, including a two-dimensional elasto-viscoplastic laminated composite with and without exponential strain hardening and two-dimensional and three-dimensional elasto-viscoplastic polycrystals. There, the method is also used to accelerate macroscale experiments in three dimensions with the elastic-viscoplastic polycrystalline material.

In addition to modeling memory effectively, the RNO architecture may be extended to include dependence on the material itself; we refer to this extended architecture as the Recurrent Neural Operator-Fourier Neural Mapping (RNO-FNM), where the FNM is a variant of the Fourier Neural Operator and is developed in Chapter 5. This modification of the architecture allows the neural networks that model the time derivatives of the state to take both function-valued and finite vector-valued inputs. We present this architecture and a material-dependent one-dimensional viscoelasticity experiment; additional material-dependent experiments are performed in [18].

We first describe the overarching mathematical framework adopted and present a literature review. This is followed by a statement of our contributions and an overview of the chapter. Finally, we summarize notation used throughout

the remainder of the chapter.

Literature Review

The continuum assumption for physical materials approximates the inherently particulate nature of matter by a continuous medium and thus allows the use of partial differential equations to describe response dynamics. We refer the reader to [30, 31, 32] for a general background. In continuum mechanics, the governing equations are derived by combining universal balance laws of physics (balance of mass, momenta, and energy) with a constitutive relation that describes the properties of the material being studied. This is typically specified as the relation between a dynamic quantity like stress or energy and kinematic quantities like strain and its history. The constitutive relation of many materials are history dependent, i.e., the state of stress at an instant depends on the history of deformation. It is common in continuum mechanics to incorporate this history dependence through the introduction of internal variables, which are referred to as hidden variables in computer science. We refer the reader to [17] for a systematic formulation of internal variable theories.

Of particular interest in this chapter are viscoelastic materials, as we develop theory for our method in this setting. We refer the reader to [33, 34] for a general background. In viscoelastic materials, the state of stress at any instant depends on the strain and its history. There are various models where the stress depends only on strain and strain rate (Kelvin-Voigt), internal variables (standard linear solids), convolution kernels, and fractional time derivatives.

While constitutive laws were traditionally determined empirically, more recently there has been a systematic attempt to understand them from more fundamental solids, and this has given rise to a rich activity in multiscale modeling of materials [35, 36, 37]. Materials are heterogeneous on various length (and time) scales, and it is common to use different theories to describe the behavior at different scales [38]. The goal of multiscale modeling of materials is to use this hierarchy of scales to understand the overall constitutive behavior at the scale of applications. The hierarchy of scales includes a number of continuum scales. For example, a composite material is made of various distinct materials arranged at a scale that is small compared to the scale of application but large enough compared to an atomistic/discrete scale, so the behavior is adequately described by continuum mechanics. Or, for example, a polycrystal

is made of a large collection of grains (regions of identical anisotropic material but with differing orientation) that are small compared to the scale of application but large enough for a continuum theory. Homogenization theory leverages the assumption of the separation of scales to average out the effects of fine-scale material variations. To estimate macroscopic response of heterogeneous materials, asymptotic expansion of the displacement field yields a set of boundary value problems whose solution produces an approximation that does not depend on the microscale [26, 39]. The fundamentals of asymptotic homogenization theory are well-established [27, 40, 41]. Milton [42] provides a comprehensive survey of the effective or homogenized properties.

Homogenization in the context of viscoelasticity was initiated by Sanchez-Palencia ([39] Chapter 6), who pointed out that the homogenization of a Kelvin-Voigt model leads to a model with fading memory. Further discussion of homogenization theory in (thermo-)viscoelasticity can be found in Francfort and Suquet [28], and a detailed discussion of the overall behavior including memory in Brenner and Suquet [43]. A broader discussion of homogenization and memory effects can be found in Tartar [44]. It is now understood that homogenization of various constitutive models gives rise to memory.

As noted above, according to homogenization theory, the macroscopic behavior depends on the solution of a boundary value problem at the microscale. Evaluating the macroscopic behavior by the solution of a boundary value problem computationally leads to what has been called computational micromechanics [45]. These often involve periodic boundary conditions, and fast Fourier transform-based methods are widely used since Moulinec and Suquet [46] (see [47, 48] for recent summaries). While these enable us to compute the macroscopic response for a particular deformation history, one needs to repeat the calculation for all possible deformation histories.

Therefore, recent work in the mechanics literature addresses the issue of learning homogenized constitutive models from computational data [49, 50, 29, 16] or experimental data [51]. This learning problem requires determination of maps that take as inputs functions describing microstructural properties and leads in to the topic of operator learning. Operator learning is a branch of machine learning designed to approximate maps between infinite dimensional function spaces [52]. In the case of constitutive models, the data come in the form of pairs of input functions: the time trajectories of average strain and the

time trajectories of average stress. Causal architectures have been developed for finite-dimensional maps, namely, recurrent neural networks (RNNs) with LSTM units and GRU networks, both of which have been used for constitutive models [53, 49]. Here we present a causal neural operator architecture, the RNO. The RNO architecture is modeled after internal variable theories of history-dependent materials [17, 30]. These theories maintain that the effect of history may be summarized by a fixed number of state variables that are updated at each point in time. Accordingly, the RNO assumes a Markovian model by maintaining a fixed number of hidden variables that also change over time. Furthermore, unlike the finite-dimensional architectures, the RNO is time-discretization invariant; when the correct model hypothesis is assumed, the RNO maintains accuracy when the input time discretization is altered. This idea is explored in numerical experiments in this work.

Use of the data-driven RNO model has two aims. First, we may gain insight into the internal variables and facilitate model discovery. Indeed, we show empirically that hypothesizing the correct model form in the RNO inputs leads to greater time-discretization invariance. Second, the learned model may be used to accelerate computations of macroscale behavior as a surrogate model. Recalling the separation of scales, the RNO serves to resolve the microscale dynamics efficiently so that the resulting averaged dynamics may be used in computations on the macroscale. In this chapter we use an RNO as a surrogate model for the constitutive relation on the microscale. Our RNO architecture takes the form of two feed-forward neural networks: one which computes the time derivative of hidden variables, and one which outputs the stress pointwise in space and time. In this manner, the history dependence is contained entirely in the hidden variables rather than directly in the neural network. This leads to an interpretable model. The RNO can then be used to evaluate the forward dynamic response on the microscale cells, whose results are combined with traditional numerical approximation methods to yield the macroscale response. Furthermore, the RNO that we train at a particular time discretization is also accurate when used at other time discretizations if the correct model form is proposed.

Without additional modification, the RNO may only be used on a particular material microstructure, and data-driven constitutive models must be retrained for new microstructures. In an extension paper [18], we develop a

method to consider material microstructure as a model input. Some alternate architectures have been proposed that take in summary statistics of the material to sidestep this issue [54, 49, 55]. Our extended method recognizes that the material microstructure takes the form of a function input on a domain. This approach has been taken in [56] using the Fourier Neural Operator (FNO) architecture on an elastic material to map the material microstructure to the solution of the cell problem PDE. However, in the case of elasticity, there is no history dependence. Here, we encode the material as a function as input to the neural networks in our RNO architecture, thereby addressing both material and history dependence simultaneously. This extension is summarized in this chapter and a numerical experiment with material dependence is presented as well.

Our Contributions and Chapter Overview

Our contributions are as follows:

1. We propose a data-driven Markovian model to learn homogenized constitutive laws in viscoelasticity and plasticity in the form of the RNO. We provide theoretical underpinnings for this model in the case of one-dimensional viscoelasticity.

2. We prove that in the one-dimensional Kelvin-Voigt (KV) setting, any solution of the multiscale problem can be approximated by the solution of a homogenized problem with Markovian structure and that the constitutive model for this Markovian homogenized system can be approximated by the RNO learned from data generated by solving the appropriate cell problem.

3. We provide simulations which numerically demonstrate the accuracy of the learned Markovian model in several application materials, including elasto-viscoplasticity.

4. We extend the methodology to the case of material dependence as well, introducing the RNO-FNM architecture which models history dependence and material dependence simultaneously.

In Section 2.2, we formulate the KV viscoelastic problem and its homogenized solution. In Section 2.3, we present our main theoretical results, addressing

contributions 1 and 2; these are in the setting of one-dimensional KV viscoelasticity. We prove that the solution of the multiscale problem can be approximated by solution of a homogenized Markovian memory-dependent model that does not depend on small scales, and we prove that an RNO can approximate the constitutive law for this homogenized problem. Section 2.4 summarizes the extension to material dependence, addressing contribution 3. Finally, Sections 2.5 and 2.6 contains numerical experiments which make contribution 4.

Notation

Let $\mathcal{D} \subset \mathbb{R}^d$ be a bounded open set and $\mathcal{T} = (0, T)$ to be the bounded time domain of interest. We denote by $\mathbb{T}^d$ the d-dimensional torus $[0, 1]^d$. Let $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ denote the standard inner product and induced norm operations on the Hilbert space $L^2(\mathcal{D}; \mathbb{R})$. Additionally, let $\| \cdot \|_\infty$ denote the $L^\infty(\mathcal{D}; \mathbb{R})$ norm. The space $W^{k,p}(\mathcal{T}; \mathbb{R}^d)$ denotes the Sobolev space of functions defined on $\mathcal{T}$ with weak derivatives up to order k which are all in $L^p(\mathcal{T}; \mathbb{R}^d)$, $1 \leq p \leq \infty$.

It is convenient to define the $\xi-$dependent quadratic form

$$q_\xi(u, w) := \int_{\mathcal{D}} \xi(x) \frac{\partial u(x)}{\partial x} \frac{\partial w(x)}{\partial x} \, dx \tag{2.1.1}$$

for arbitrary $\xi \in L^\infty\big(\mathcal{D}; (0, \infty)\big)$; furthermore we define

$$\xi^+ := \operatorname*{ess\,sup}_{x \in \mathcal{D}} \xi(x) < \infty \tag{2.1.2}$$

and

$$\xi^- := \operatorname*{ess\,inf}_{x \in \mathcal{D}} \xi(x) \geq 0. \tag{2.1.3}$$

In this chapter we always work with ξ such that $\xi^- > 0$. Under these assumptions $q_\xi(\cdot, \cdot)$ defines an inner product, and we can define the following norm

$$\|u\|^2_{H^1_0, \xi} := q_\xi(u, u)$$

from it; note also that we may define a norm on $H^1_0(\mathcal{D}; \mathbb{R})$ by

$$\|u\|^2_{H^1_0} := q_{\mathbf{1}}(u, u),$$

where $\mathbf{1}(\cdot)$ is the function in $L^\infty\big(\mathcal{D}; (0, \infty)\big)$ taking value 1 in $\mathcal{D}$ a.e.. The resulting norms are all equivalent on the space $H^1_0(\mathcal{D}; \mathbb{R})$; this is a consequence of the following lemma:

***Lemma* 2.1.1.** For any $\xi_1, \xi_2 \in L^\infty\big(\mathcal{D}; (0,\infty)\big)$ satisfying properties (2.1.2) and (2.1.3), the norms $\|u\|_{H_0^1,\xi_1}$ and $\|u\|_{H_0^1,\xi_2}$ are equivalent in the sense that

$$\frac{\xi_2^-}{\xi_1^+}\|u\|^2_{H_0^1,\xi_1} \le \|u\|^2_{H_0^1,\xi_2} \le \frac{\xi_2^+}{\xi_1^-}\|u\|^2_{H_0^1,\xi_1}.$$

◊

Proof. For $i = 1, 2$:

$$\xi_i^- \int_0^1 \left|\frac{\partial u}{\partial x}\right|^2 \mathsf{d}x \le \int_0^1 \xi_i(x)\left|\frac{\partial u}{\partial x}\right|^2 \mathsf{d}x \le \xi_i^+ \int_0^1 \left|\frac{\partial u}{\partial x}\right|^2 \mathsf{d}x.$$

The result follows. □

As a consequence of the preceding lemma, we may use $q_\xi(u, w)$ as the inner product on the space $H_0^1(\mathcal{D}; \mathbb{R})$ for any ξ satisfying (2.1.2) and (2.1.3). We also define

$$\mathcal{Z} = L^\infty(\mathcal{T}; L^2(\mathcal{D}; \mathbb{R}^m)), \quad \mathcal{Z}_2 = L^2(\mathcal{T}; L^2(\mathcal{D}; \mathbb{R}^m))$$

with norms

$$\|r\|_{\mathcal{Z}} = \operatorname*{ess\,sup}_{t\in\mathcal{T}}(\|r(\cdot,t)\|), \quad \|r\|_{\mathcal{Z}_2} = \Big(\int_0^{\mathcal{T}} \|r(\cdot,t)\|^2 \,\mathsf{d}t\Big)^{\frac{1}{2}}.$$

We note that $\mathcal{Z}$ is continuously embedded into $\mathcal{Z}_2$.

For any time-dependent function g we denote by $\{g(t)\}_{t\in\mathcal{T}}$ the set that includes pointwise evaluation of $g(t)$ and its time-derivative for all $t \in \mathcal{T}$. When it is clear in the appropriate context, we write g as shorthand for $\{g(t)\}_{t\in\mathcal{T}}$. We use $\dot{g}$ to indicate a time derivative of the trajectory g. In particular $\dot{\bar{\epsilon}}$ is a time derivative of $\bar{\epsilon}$. Note however that, in the context of elasto-viscoplasticity, we use the commonly adopted convention that $\dot{\epsilon}_{p0}$ denotes the rate constant; in particular it is not the derivative of a time-dependent function.

We denote the *variation* of a function $u \in L^1_{\mathrm{loc}}(\mathcal{D})$ by

$$V(u,\mathcal{D}) = \sup\Big\{\sum_{i=1}^d \int_{\mathcal{D}} \frac{\partial \Phi_i}{\partial x_i} u \,\mathsf{d}x : \ \Phi \in C_0^\infty(\mathcal{D}; \mathbb{R}^d),\ \|\Phi\|_{L^\infty(\mathcal{D};\mathbb{R}^d)} \le 1\Big\}$$

and the set of functions of bounded variation on $\mathbb{T}^d$ as

$$\mathrm{BV} = \{u \in L^1(\mathbb{T}^d) : \ V(u, \mathbb{T}^d) < \infty\}.$$

For further information on BV, we refer to [57].

Denote by $\mathcal{M}^B_{f_{\min},f_{\max}}$ the set of functions $f \in \mathrm{BV}(\mathcal{D})$ satisfying

$$|f|_{\mathrm{BV}} \leq B, \quad \operatorname*{ess\,sup}_{y\in\mathcal{D}} f(y) \leq f_{\max}, \quad \operatorname*{ess\,inf}_{y\in\mathcal{D}} f(y) \geq f_{\min} \tag{2.1.4}$$

for some $0 < f_{\min} \leq f_{\max} < \infty$ and $B > 0$.

Setting

Consider the problem of material response on an arbitrary spatial domain $\mathcal{D} \subset \mathbb{R}^d$ where the material properties vary rapidly within the domain. We denote by $u_\varepsilon \in \mathbb{R}^d$ the displacement, where $\varepsilon : 0 < \varepsilon \ll 1$ denotes the scale of the material fluctuations. We consider multiscale continuum models which satisfy dynamical equations of the form

$$\rho\partial_t^2 u_\varepsilon(x,t) = \nabla_x \cdot \sigma_\varepsilon(x,t) + f(x,t), \qquad x \in \mathcal{D}, t \in \mathcal{T}, \tag{2.1.5a}$$

$$\sigma_\varepsilon(x,t) = \Psi^\dagger_\varepsilon(\{\nabla_x u_\varepsilon(x,s)\}_{s\in\mathcal{T}}; M, x)(t), \qquad x \in \mathcal{D}, t \in \mathcal{T}, \tag{2.1.5b}$$

$$u_\varepsilon(x,0) = \partial_t u_\varepsilon(x,0) = 0, \qquad x \in \mathcal{D}, \tag{2.1.5c}$$

$$u_\varepsilon(x,t) = 0, \qquad x \in \partial\mathcal{D}, t \in \mathcal{T}. \tag{2.1.5d}$$

From these equations we seek the displacement $u_\varepsilon : \mathcal{D} \times \mathcal{T} \mapsto \mathbb{R}^d$. Equation (2.1.5a) is the balance equation with inertia term $\rho\partial_t^2 u_\varepsilon$ for known parameter $\rho \in \mathbb{R}^+$, resultant stress term $\nabla \cdot (\sigma_\varepsilon)$ where $\sigma_\varepsilon \in \mathbb{R}^{d\times d}$ is the internal stress tensor, and known external forcing $f \in \mathbb{R}^d$; equations (2.1.5c,2.1.5d) specify the initial and boundary data for the displacement. M represents the dependence on the parameters describing the material. Equation (2.1.5b) is the constitutive law relating properties of the the strain $\nabla_x u_\varepsilon$ to the stress σ_ε via map $\Psi^\dagger_\varepsilon$. The dependence on $\{\nabla_x u_\varepsilon(x,s)\}_{s\in\mathcal{T}}$ indicates that $\Psi^\dagger_\varepsilon$ may depend on the entire strain history up to time t as well as the time derivatives of the strain up to that point. Additionally $\Psi^\dagger_\varepsilon$ depends on x to allow for material properties which depend on the rapidly varying $\frac{x}{\varepsilon}$; it is also possible to allow for material properties which exhibit additional dependence on the slowly varying x, but this case is beyond the scope of our work. In this chapter we will consider this model with inertia ($\rho > 0$) and without inertia ($\rho \equiv 0$). The form of the multiscale constitutive model in 2.1.5 includes a variety of plastic, viscoelastic, and viscoplastic materials.

The goal of homogenization is to find constitutive models which eliminate the small-scale dependence on ε and obtain a homogenized constitutive law Ψ_0

and homogenized displacement u_0 that satisfy equations

$$\rho\partial_t^2 u_0(x,t) = \nabla_x \cdot \sigma_0(x,t) + f(x,t), \qquad x \in \mathcal{D}, t \in \mathcal{T}, \tag{2.1.6a}$$

$$\sigma_0(x,t) = \Psi_0^\dagger(\{\nabla_x u_0(x,s)\}_{s\in\mathcal{T}}; M)(t), \qquad x \in \mathcal{D}, t \in \mathcal{T}, \tag{2.1.6b}$$

$$u_0(x,0) = \partial_t u_0(x,0) = 0, \qquad x \in \mathcal{D}, \tag{2.1.6c}$$

$$u_0(x,t) = 0, \qquad x \in \partial\mathcal{D}, t \in \mathcal{T}. \tag{2.1.6d}$$

Throughout this work, we denote by $\bar{\epsilon}$ the homogenized strain that appears in equations (2.1.6); $\bar{\epsilon}(x,t) \coloneqq \nabla_x u_0(x,t)$ for all $x \in \mathcal{D}$ and $t \in \mathcal{T}$. The key property of this homogenized model is that parameter ε no longer appears. Furthermore, since we assumed that the multiscale model material properties depend only on the rapidly varying scale x/ε and not on x, we have that $\Psi_0^\dagger$ does not depend explicitly on x; it does, however, still have spatial dependence through the local values of strain, strain rate, and strain history. Furthermore, for appropriate $\Psi_0^\dagger$, for small enough ε, this model has the property that $(u_0, \sigma_0) \approx (u_\varepsilon, \sigma_\varepsilon)$. If the homogenized model is identified correctly, then dynamics under the multiscale model $\Psi_\varepsilon^\dagger$, i.e. u_ε, can be approximated by dynamics under the homogenized model $\Psi_0^\dagger$, i.e. u_0. This potentially facilitates cheaper computations since length-scales of size ε need not be resolved. Often, an explicit expression for $\Psi_0^\dagger$ is unattainable, and $\Psi_0^\dagger$ is instead approximated numerically. In this chapter we first examine a case where we *can* express $\Psi_0^\dagger$ exactly, namely, the case of one-dimensional KV viscoelasticity where the underlying material is piecewise constant. In this case, we may do a complete analysis.

We observe, however, that for KV viscoelasticity, the homogenized model contains non-Markovian history dependence (memory) even though the multiscale model does not. Markovian history dependence is desirable for two reasons: first, Markovian models encode conceptual understanding, representing the history dependence in a compact, interpretable form; second, Markovian expression reduces computational cost from $\mathcal{O}(|\mathcal{T}|^2)$ in the general memory case to $\mathcal{O}(|\mathcal{T}|)$ in the Markovian case. In the general media setting, for a multitude of models in viscoelasticity, viscoplasticity, and plasticity, the homogenized model will depend on the memory in a non-Markovian manner. However, it is interesting to determine situations in which accurate Markovian approximations can be found. In fact, the one-dimensional piecewise-constant KV case delivers a Markovian form for the history dependence. The analysis of this

case supports the use of a data-driven Markovian model for approximating $\Psi_0^\dagger$ and forms the basis of the theory developed in this chapter.

We now define the RNO model used to create the Markovian approximation to $\Psi_0^\dagger$.

***Definition* 2.1.2** (RNO Architecture)**.** Define $\Psi^{\mathrm{RNO}} : C^1(\mathcal{T};\mathbb{R}^{d\times d}) \to C(\mathcal{T};\mathbb{R}^{d\times d})$, $\{\overline{\epsilon}(t)\}_{t\in\mathcal{T}} \mapsto \{\overline{\sigma}(t)\}_{t\in\mathcal{T}}$ through

$$\overline{\sigma}(t) = \mathcal{F}(\overline{\epsilon}(t), \dot{\overline{\epsilon}}(t), \xi(t)), \qquad t \in \mathcal{T}, \tag{2.1.7a}$$

$$\dot{\xi}(t) = \mathcal{G}(\overline{\epsilon}(t), \xi(t)), \qquad t \in \mathcal{T}, \tag{2.1.7b}$$

$$\xi(0) = 0, \tag{2.1.7c}$$

where $\mathcal{F}$ and $\mathcal{G}$ are feed-forward neural networks, and $\xi \in \mathbb{R}^L$ for some $L \in \mathbb{Z}_+$. ◊

In the above definition, ξ is a vector of hidden variables, closely related to the *internal variables* used in the mechanics literature. The hidden variables vector ξ carries the history dependence on $\overline{\epsilon}$ through its Markovian evolution. In dimension $d > 1$ there are further symmetries that should be built into the model, but as the concrete analysis in this chapter is in dimension $d = 1$ we do not detail these symmetries here [58]. Since the RNO only applies to a single material microstructure, we have dropped the explicit parametric dependence on the material M, as any dependence on the material itself will be encoded in the trainable parameters of $\mathcal{F}$ and $\mathcal{G}$. Dependence on the material is reintroduced in Section 2.4.

In general such a Markovian model can only *approximate* the true model, and the nature of the physics leading to a good approximation will depend on the specific continuum mechanics problem. To determine $\mathcal{F}$ and $\mathcal{G}$ in practice we parameterize them as neural networks, which enables us to use general purpose optimization software to determine suitable values of the parameters. Within computational implementations of the learned homogenized models, the neural networks $\mathcal{F}$ and $\mathcal{G}$ act pointwise in time to generate the stress and time derivatives of the hidden variables at each time step. In doing so we identify an operator class $\Psi_0(\cdot\,;\theta)$ and parameter space Θ such that, for some judiciously chosen $\theta^* \in \Theta$, $\Psi_0(\cdot\,;\theta^*) \approx \Psi_0^\dagger$.

In this chapter we concentrate on justifying a Markovian homogenized approximation in the context of one-dimensional KV viscoelasticity. Our justi-

fication uses theory that is specific to one-dimensional linear viscoelasticity, and we demonstrate that the approach also works for the general SLS, which includes the KV model as a particular limit. We also include results from one-dimensional elasto-viscoplasticity. Furthermore, the paper [16] contains evidence that the ideas we develop apply beyond one dimension and into nonlinear plasticity in higher spatial dimensions.

The specific property of one-dimensional viscoelasticity that we exploit to underpin our analysis is that, for piecewise-constant media, the homogenized model has a memory term which can be represented in a Markovian way. Therefore, to justify our strategy of approximating by Markovian models we: first, approximate the rapidly varying medium by a piecewise-constant rapidly varying medium; second, homogenize this model to find a Markovian description; and finally, demonstrate how the Markovian description can be learned from data at the level of the unit cell problem. For more general problems we anticipate a similar justification holding, but with different specifics leading to the existence of good approximate Markovian homogenized models. The benefit of the one-dimensional viscoelastic setting is that, through theory, we obtain underpinning insight into the conceptual approach more generally. This theory underpins the numerical experiments which follow.

2.2 One-Dimensional Kelvin-Voigt Viscoelasticity

The theory of this chapter is focused on one-dimensional KV viscoelasticity because the model is amenable to rigorous analysis. The resulting analysis sheds light on the learning of constitutive models more generally.

Governing Equations and Weak Form

The one-dimensional KV model for viscoelasticity postulates that stress is affine in the strain and strain rate, with affine transformation dependent on the spatially varying material properties. For a multiscale material varying with respect to x/ε we thus have the following definition of $\Psi_\varepsilon^\dagger$ from (2.1.5), in the one-dimensional KV model:

$$\sigma_\varepsilon = E_\varepsilon \partial_x u_\varepsilon + \nu_\varepsilon \partial_{xt}^2 u_\varepsilon,$$

where $E_\varepsilon(x) = E\left(\frac{x}{\varepsilon}\right)$ and $\nu_\varepsilon(x) = \nu\left(\frac{x}{\varepsilon}\right)$ are rapidly varying material elasticity and viscosity, respectively. Both E and ν are assumed to be 1-periodic. Then equations (2.1.5) without inertia ($\rho \equiv 0$) on spatial domain $\mathcal{D} = [0, D]$ become

$$-\partial_x\big(E_\varepsilon(x)\partial_x u_\varepsilon(x,t)+\nu_\varepsilon(x)\partial^2_{xt}u_\varepsilon(x,t)\big)=f(x,t), \qquad x\in\mathcal{D}, t\in\mathcal{T}, \tag{2.2.1a}$$

$$u_\varepsilon(x,0)=\partial_t u_\varepsilon(x,0)=0, \qquad x\in\mathcal{D}, \tag{2.2.1b}$$

$$u_\varepsilon(0,t)=u_\varepsilon(D,t)=0, \qquad t\in\mathcal{T}. \tag{2.2.1c}$$

Any classical solution to equations (2.2.1) will also solve the corresponding *weak form*: find $u_\varepsilon \in C^1(\mathcal{T}; H^1_0(\mathcal{D};\mathbb{R}))$ such that

$$q_{\nu_\varepsilon}(\partial_t u_\varepsilon,\varphi)+q_{E_\varepsilon}(u_\varepsilon,\varphi)=\langle f,\varphi\rangle \tag{2.2.2}$$

for all test functions $\varphi \in H^1_0(\mathcal{T}; H^1_0(\mathcal{D};\mathbb{R}))$.

Homogenization

In the inertia-free setting $\rho = 0$ we perform homogenization to eliminate the dependence on the small scale ε in (2.2.1). First, we take the Laplace transform of (2.2.1), which gives, for Laplace parameter s and with the hat symbol denoting Laplace transform,

$$-\partial_x((E_\varepsilon(x)+\nu_\varepsilon(x)s)\partial_x\widehat{u}_\varepsilon(x,s))=\widehat{f}(x,s), \qquad x\in\mathcal{D},$$
$$\widehat{u}_\varepsilon(0,s)=\widehat{u}_\varepsilon(D,s)=0.$$

The initial condition is applied upon Laplace inversion. Since $\epsilon \ll 1$, we may apply standard techniques from multiscale analysis [26, 27] and seek a solution in the form

$$\widehat{u}_\varepsilon=\widehat{u}_0+\varepsilon\widehat{u}_1+\varepsilon^2\widehat{u}_2+\dots\,.$$

Let $E_\varepsilon(x)=E(\frac{x}{\varepsilon})$ and $\nu_\varepsilon(x)=\nu(\frac{x}{\varepsilon})$ for $E,\nu:\mathbb{T}\to\mathbb{R}$. For convenience, define $\widehat{a}(y,s)=E(y)+\nu(y)s$. Note that $\widehat{a}(\cdot,s)$ is 1-periodic. The leading order term in our approximation, $\widehat{u}_0$, solves the following uniformly elliptic PDE with Dirichlet boundary conditions:

$$-\partial_x(\widehat{a}_0(s)\partial_x\widehat{u}_0(x,s))=\widehat{f}(x,s) \qquad x\in\mathcal{D}, \tag{2.2.3a}$$

$$\widehat{u}_0(0,s)=\widehat{u}_0(D,s)=0. \tag{2.2.3b}$$

Here the coefficient $\widehat{a}_0$ is given by

$$\widehat{a}_0(s)=\int_0^1(\widehat{a}(y,s)+\widehat{a}(y,s)\partial_y\chi(y))\ \mathsf{d}y$$

and $\chi(y) : \Omega \to \mathbb{R}$ for $\Omega = [0,1]$ satisfies the *cell problem*

$$-\partial_y(\widehat{a}(y,s)\partial_y\chi(y)) = \partial_y\widehat{a}(y,s), \qquad y \in \Omega, \tag{2.2.4a}$$

$$\int_0^1 \chi(y)\,\mathsf{d}y = 0, \quad \chi \text{ is 1-periodic.} \tag{2.2.4b}$$

Using this, the coefficient $\widehat{a}_0$ can be computed explicitly as the harmonic average of the original coefficient $\widehat{a}$ [27]:

$$\widehat{a}_0(s) = \left\langle \widehat{a}(y,s)^{-1} \right\rangle^{-1} = \left(\int_0^1 \frac{\mathsf{d}y}{s\nu(y) + E(y)} \right)^{-1}, \tag{2.2.5}$$

where $\langle \cdot \rangle$ denotes spatial averaging over the unit cell Ω.

Equations (2.2.3a) indicate that the homogenized map $\Psi_0^\dagger$ appearing in (2.1.6) is, for one-dimensional linear viscoelasticity, defined from

$$\Psi_0^\dagger\big(\{\partial_x u_0(x,\tau)\}_{\tau\in\mathcal{T}}\big)(t) = \mathcal{L}^{-1}\Big(\widehat{a}_0(s)\partial_x\widehat{u}_0\Big)(t); \tag{2.2.6}$$

here $\mathcal{L}^{-1}$ denotes the inverse Laplace transform. Note that (2.2.5) shows that $\widehat{a}_0$ grows linearly in $s \to \infty$ and computing the constant term in a regular power series expansion at $s = \infty$ shows that we may write

$$\widehat{a}_0(s) = \nu' s + E' + \widehat{\kappa}(s),$$

where $\widehat{\kappa}(s)$ decays to 0 as $s \to \infty$. Here

$$\nu' = \left(\int_0^1 \frac{1}{\nu(y)}\,\mathsf{d}y \right)^{-1}, \quad E' = \left(\int_0^1 \frac{E(y)}{\nu(y)^2}\,\mathsf{d}y \right) \Big/ \left(\int_0^1 \frac{1}{\nu(y)}\,\mathsf{d}y \right)^2.$$

Details are presented in Appendix A.2. Laplace inversion of $\widehat{a}_0(s)\partial_x\widehat{u}_0$ then yields the conclusion that

$$\Psi_0^\dagger\big(\{\overline{\epsilon}(x,\tau)\}_{\tau\in\mathcal{T}}\big)(t) = E'\overline{\epsilon}(x,t) + \nu'\dot{\overline{\epsilon}}(x,t) - \int_0^t \kappa(t-\tau)\overline{\epsilon}(x,\tau)\;\mathsf{d}\tau, \tag{2.2.7}$$

where we define $\overline{\epsilon} = \partial_x u_0$ to indicate the homogenized strain.

***Remark* 2.2.1.** When $\rho = 0$, the homogenized solution provably approximates u_ε in the $\varepsilon \to 0$ limit; see Theorem 2.3.7. However, although we derived it with inertia set to zero, the homogenized solution given by equation (2.2.8) is also valid when the inertia term $\rho\partial_t^2 u_\varepsilon$ generates contributions which are $\mathcal{O}(1)$ with respect to ε. ◊

The homogenized PDE for one-dimensional viscoelasticity follows by combining equations (2.1.6) with equation (2.2.7) to give

$$\rho\partial_t^2 u_0(x,t) = \nabla_x \cdot \sigma_0(x,t) + f(x,t), \quad \partial_x u_0(x,t) = \bar{\epsilon}(x,t), \quad x \in \mathcal{D}, t \in \mathcal{T}, \tag{2.2.8a}$$

$$\sigma_0(x,t) = E'\bar{\epsilon}(x,t) + \nu'\dot{\bar{\epsilon}}(x,t) - \int_0^t \kappa(t-\tau)\bar{\epsilon}(x,\tau)\, d\tau, \quad x \in \mathcal{D}, t \in \mathcal{T}, \tag{2.2.8b}$$

$$u_0(x,0) = \partial_t u_0(x,0) = 0, \qquad x \in \mathcal{D}, \tag{2.2.8c}$$

$$u_0(0,t) = u_0(D,t) = 0, \qquad t \in \mathcal{T}. \tag{2.2.8d}$$

The price paid for homogenization is dependence on the strain history. We show in the next section, however, that we can approximate the general homogenized map with one in which the history dependence is expressed in a Markovian manner.

2.3 Main Theorems: Statement and Interpretation

In this section we present results of three types. First, we show that the solution u_ε to equation (2.2.1) is Lipschitz when viewed as a map from the unit cell material properties $E(\cdot), \nu(\cdot)$ in L^∞ into $\mathcal{Z}$; hence, an $\mathcal{O}(\delta)$ approximation of E, ν by piecewise-constant functions leads to an $\mathcal{O}(\delta)$ approximation of u_ε. Second, we demonstrate that the homogenized model based on piecewise-constant material properties can be represented in a Markovian fashion by introducing *hidden variables*; hence, combining with the first point, we have a mechanism to approximate u_ε by solving a Markovian homogenized model. Third, we show the existence of neural networks which provide arbitrarily good approximation of the constitutive law arising in the Markovian homogenized model; this suggests a model class within which to learn homogenized, Markovian constitutive models from data. We then establish our framework for the optimization methods used to learn such constitutive models; this framework is employed in the subsequent Section 2.6.

***Assumption* 2.3.1.** We make the following assumptions on E, ν, and f throughout:

1. $f \in L^2(\mathcal{D}; \mathbb{R})$ for all $t \in \overline{\mathcal{T}}$; thus $\|f\|_{\mathcal{Z}} < \infty$;

2. $E^+, \nu^+ < \infty$, and $E^-, \nu^- > 0$.

$\Diamond$

Note that $E^+ = E^+_\varepsilon$ and $\nu^+ = \nu^+_\varepsilon$, so we drop the ε subscript in this notation.

Approximation by Piecewise Constant Material

Consider (2.2.1) with continuous material properties E and ν. We show in Theorem 2.3.4 that we can approximate the solution u_ε to this system by a solution $u^{\mathrm{PC}}_\varepsilon$ which solves (2.2.1) with suitable piecewise-constant material properties E^{PC} and ν^{PC}, in such a way that u_ε and $u^{\mathrm{PC}}_\varepsilon$ are close. To this end we make precise the definition of piecewise-constant material properties.

***Definition* 2.3.2** (Piecewise Constant)**.** A material is *piecewise constant* on the unit cell with L pieces if the elasticity function $E(y)$ and the viscosity function $\nu(y)$ both take constant values on L intervals $[0, a_1), [a_1, a_2), \ldots, [a_{L-1}, 1]$. In particular, $E(y)$ and $\nu(y)$ have discontinuities only at the same $L-1$ points in the unit cell. We use the terminology $L-$piecewise constant to specify the number of pieces. $\Diamond$

***Remark* 2.3.3.** The situation in which $E(y)$ and $\nu(y)$ have discontinuities at different values of $y \in (0, 1)$ can be reduced to the case in Definition 2.3.2 by increasing the value of L. $\Diamond$

***Theorem* 2.3.4** (Piecewise-Constant Approximation)**.** Let E and ν be piecewise-continuous functions, with a finite number of discontinuities, satisfying Assumptions 2.3.1; let u_ε be the corresponding solution to (2.2.1). Then, for any $\delta > 0$, there exist piecewise-constant E^{PC} and ν^{PC} such that solution $u^{\mathrm{PC}}_\varepsilon$ of equations (2.2.1) with these material properties satisfies

$$\|u^{\mathrm{PC}}_\varepsilon - u_\varepsilon\|_{\mathcal{Z}} < \delta.$$

$\Diamond$

Note that Theorem 2.3.4 is stated in the setting of no inertia. The proof depends on the following lemma; proof of both the theorem and the lemma may be found in Appendix A.1. We observe that, since the Lipschitz result is in the $L^\infty-$norm with respect to the material properties, it holds with constant

C independent of ε, in the case of interest where the material properties vary rapidly on scale ε.

***Lemma* 2.3.5** (Lipschitz Solution)**.** Let u_i be the solution to

$$-\partial_x\big(E_i(x)\partial_x u_i(x,t) + \nu_i(x)\partial^2_{xt}u_i(x,t)\big) = f(x,t), \qquad x \in \mathcal{D}, t \in \mathcal{T}, \quad (2.3.1)$$

$$u_i(x,0) = \partial_t u_i(x,0) = 0, \qquad x \in \mathcal{D}, \quad (2.3.2)$$

$$u_i(0,t) = u_i(D,t) = 0, \qquad t \in \mathcal{T}, \quad (2.3.3)$$

associated with material properties E_i, ν_i, for $i \in \{1,2\}$, and forcing f, all satisfying the Assumptions 2.3.1. Then

$$\|u_1 - u_2\|_{\mathcal{Z}} \le C(\|\nu_1 - \nu_2\|_\infty + \|E_1 - E_2\|_\infty)$$

for some constant $C \in \mathbb{R}^+$ dependent on $f, E_i^+, E_i^-, \nu_i^+, \nu_i^-$, and L and independent of ε. ◊

Homogenization for Piecewise Constant Material

We show in Theorem 2.3.6 that for piecewise-constant material properties $E(\cdot)$ and $\nu(\cdot)$, the homogenized map $\Psi_0^\dagger$ given in (2.2.7) can be written explicitly with a finite number of parameters, and in particular the memory is expressible in a Markovian form. This Markovian form implicitly defines a finite number of hidden variables.

***Theorem* 2.3.6** (Existence of Exact Parametrization)**.** Let $\Psi_0^\dagger$ be the map from strain history to stress in the homogenized model, as defined by equation (2.2.7), in a piecewise-constant material with $L+1$ pieces. Define $\Psi_0^{\mathrm{PC}} : \mathbb{R}^2 \times C^1(\mathcal{T};\mathbb{R}) \times \mathcal{T} \times \Theta \to \mathbb{R}$ by

$$\Psi_0^{\mathrm{PC}}(\bar\epsilon(t), \dot{\bar\epsilon}(t), \{\bar\epsilon(\tau)\}_{\tau\in\mathcal{T}}, t; \theta) = E'\bar\epsilon(t) + \nu'\dot{\bar\epsilon}(t) - \sum_{\ell=1}^{L}\xi_\ell(t), \quad (2.3.4a)$$

$$\partial_t\xi_\ell(t) = \beta_\ell\bar\epsilon(t) - \alpha_\ell\xi_\ell(t),\ \xi_\ell(0) = 0, \quad \ell \in \{1,\dots,L\}. \quad (2.3.4b)$$

Then, under Assumptions 2.3.1, there exists a choice of parameters $\theta^* = (E', \nu', \alpha, \beta, L)$ such that

$$\Psi_0^\dagger(\bar\epsilon(t), \dot{\bar\epsilon}(t), \{\bar\epsilon(\tau)\}_{\tau\in\mathcal{T}}, t) = \Psi_0^{\mathrm{PC}}(\bar\epsilon(t), \dot{\bar\epsilon}(t), \{\bar\epsilon(\tau)\}_{\tau\in\mathcal{T}}, t; \theta^*)$$

for all $u_0 \in \mathcal{C}^2(\mathcal{D}\times\mathcal{T};\mathbb{R})$ and $t \in \mathcal{T}$. ◊

The proof of the above theorem may be found in Appendix A.1. The parameters E', ν', α, and β are determined via an appropriate decomposition of $\widehat{a}_0$ in equation (2.2.5); details are in the proof. In particular, E' and ν' are homogenized elasticity and viscosity coefficients, respectively, α are decay rates for the hidden variables ξ, and β are coefficients for each decay term. Note that the model in equations (2.3.4) is Markovian. Furthermore, although the model in (2.3.4) requires an input of t for evaluation, the spatial variable x only enters implicitly through the local values of $\overline{\epsilon}$ and $\dot{\overline{\epsilon}}$; the model acts pointwise in space. In what follows it is useful to define u_0^{PC} to be the solution to the following system defined with constitutive model Ψ_0^{PC} from Theorem 2.3.6.

$$\rho\partial_t^2 u_0^{\mathrm{PC}}(x,t) - \partial_x \sigma_0(x,t) = f(x,t), \quad \partial_x u_0^{\mathrm{PC}}(x,t) = \overline{\epsilon}^{\mathrm{PC}}(x,t), \qquad x \in \mathcal{D}, t \in \mathcal{T}, \tag{2.3.5a}$$

$$\sigma_0(x,t) = \Psi_0^{\mathrm{PC}}\Big(\overline{\epsilon}^{\mathrm{PC}}(x,t), \dot{\overline{\epsilon}}^{\mathrm{PC}}(x,t), \big\{\overline{\epsilon}^{\mathrm{PC}}(x,\tau)\big\}_{\tau\in\mathcal{T}}, t\Big), \qquad x \in \mathcal{D}, t \in \mathcal{T}, \tag{2.3.5b}$$

$$u_0^{\mathrm{PC}}(x,0) = \partial_t u_0^{\mathrm{PC}}(x,0) = 0, \qquad x \in \mathcal{D}, \tag{2.3.5c}$$

$$u_0^{\mathrm{PC}}(0,t) = u_0^{\mathrm{PC}}(D,t) = 0, \qquad t \in \mathcal{T}. \tag{2.3.5d}$$

Using a homogenization theorem, together with approximation by piecewise-constant material properties, we now show that u_ε can be approximated by u_0^{PC}; this will follow from the inequality

$$\|u_\varepsilon - u_0^{\mathrm{PC}}\|_{\mathcal{Z}_2} \le \|u_\varepsilon - u_\varepsilon^{\mathrm{PC}}\|_{\mathcal{Z}_2} + \|u_\varepsilon^{\mathrm{PC}} - u_0^{\mathrm{PC}}\|_{\mathcal{Z}_2}.$$

The first term on the right-hand side may be controlled using Theorem 2.3.4. The fact that dynamics under constitutive law $\Psi_\varepsilon^\dagger$ converge to those under $\Psi_0^\dagger$ as $\epsilon \to 0$ may be used to control the second term; this fact is a consequence of the following theorem:

***Theorem* 2.3.7.** Under Assumptions 2.3.1, the solution u_ε to equations (2.2.1) converges weakly to u_0, the solution to equations (2.2.8) with $\rho = 0$, in $W^{1,2}(\mathcal{T}; H_0^1(\mathcal{D};\mathbb{R}))$. Thus, for any $\eta > 0$ there exists $\varepsilon_{\mathrm{crit}} > 0$ such that for all $\varepsilon \in (0, \varepsilon_{\mathrm{crit}})$,

$$\|u_\varepsilon - u_0\|_{\mathcal{Z}_2} < \eta. \tag{2.3.6}$$

$\Diamond$

Proof. Since $f \in \mathcal{Z}$, continuous embedding gives $f \in \mathcal{Z}_2$. Applying Theorem 3.1[28] (noting that the work in that paper is set in dimension $d = 3$, but is readily extended to dimension $d = 1$) establishes weak convergence of u_ε to u_0 in $W^{1,2}(\mathcal{T}, H_0^1(\mathcal{D};\mathbb{R}))$. Hence strong convergence in $\mathcal{Z}_2$ follows, by compact embedding of $W^{1,2}(\mathcal{T}; H_0^1(\mathcal{D};\mathbb{R}))$ into $\mathcal{Z}_2$. □

The following corollary is a consequence of Theorem 2.3.7.

***Corollary* 2.3.8.** Under Assumptions 2.3.1 and assuming E, ν are piecewise constant, the solution $u_\varepsilon^{\mathrm{PC}}$ to equations (2.2.1) converges weakly to u_0^{PC}, the solution to equations (2.3.5) with $\rho = 0$, in $W^{1,2}(\mathcal{T}; V)$. Thus, for any $\eta > 0$ there exists $\varepsilon_{\mathrm{crit}} > 0$ such that for all $\varepsilon \in (0, \varepsilon_{\mathrm{crit}})$,

$$\|u_\varepsilon^{\mathrm{PC}} - u_0^{\mathrm{PC}}\|_{\mathcal{Z}_2} < \eta. \tag{2.3.7}$$

◊

Combining this result with that of Theorem 2.3.4, noting continuous embedding of $\mathcal{Z}$ into $\mathcal{Z}_2$, allows us to approximate u_ε by u_0^{PC}:

***Corollary* 2.3.9.** Let E and ν be piecewise-continuous functions, with a finite number of discontinuities satisfying, along with f, Assumptions 2.3.1; let u_ε be the corresponding solution to (2.2.1). Then for any $\eta > 0$, there exists L_{crit} and $\varepsilon_{\mathrm{crit}}$ with the property that for all $L \geq L_{\mathrm{crit}}$ there are $L-$piecewise-constant E^{PC} and ν^{PC} such that for all $\varepsilon \in (0, \varepsilon_{\mathrm{crit}})$, the solution to u_0^{PC} to (2.3.5) with $\rho = 0$ satisfies

$$\|u_\varepsilon - u_0^{\mathrm{PC}}\|_{\mathcal{Z}_2} < \eta. \tag{2.3.8}$$

◊

Neural Network Approximation of the Constitutive Model

For the specific KV model in one dimension we know the postulated form of Ψ_0^{PC} and can in principle use this directly as a constitutive model. However, in more complex problems we do not know the constitutive model analytically, and it is then desirable to learn it from data from within an expressive model class. To this end we demonstrate that Ψ_0^{PC} can be approximated by an operator Ψ_0^{RNO} which has a similar form to that defined by equations (2.3.4) but in which the right-hand sides of those equations are represented

by neural networks, leading to a recurrent neural network structure. With this structure, the neural network outputs the stress at a single point in space and time; in practice, repeated evaluation generates output stress trajectories from the spatio-temporal dynamics. As such, the architecture is not the same as standard LSTM RNN models. Instead, the feed-forward neural network $\mathcal{G}$ produces time derivatives of the hidden variables ξ, which are used in a forward Euler step to generate the updated hidden variable value. In this manner, the model acts pointwise but incorporates memory through a hidden variable. The feed-forward networks $\mathcal{F}$ and $\mathcal{G}$ are the same at every time step, justifying the "recurrent" terminology.

Let $\theta^* = (E', \nu', \alpha, \beta, L)$, be those chosen in Theorem 2.3.6 to achieve the equivalence

$$\Psi_0^\dagger(\bar{\epsilon}(t), \dot{\bar{\epsilon}}(t), \{\bar{\epsilon}(\tau)\}_{\tau\in\mathcal{T}}, t) = \Psi_0^{\mathrm{PC}}(\bar{\epsilon}(t), \dot{\bar{\epsilon}}(t), \{\bar{\epsilon}(\tau)\}_{\tau\in\mathcal{T}}, t; \theta^*).$$

We first define the linear functions $\mathcal{F}^{\mathrm{PC}} : \mathbb{R}\times\mathbb{R}\times\mathbb{R}^L \to \mathbb{R}$ and $\mathcal{G}^{\mathrm{PC}} : \mathbb{R}^L\times\mathbb{R} \to \mathbb{R}$ by

$$\mathcal{F}^{\mathrm{PC}}(b, c, r) = E'b + \nu'c - \langle \mathbf{1}, r\rangle \tag{2.3.9a}$$
$$\mathcal{G}^{\mathrm{PC}}(r, b) = -Ar + \beta b, \tag{2.3.9b}$$

where $A = \mathrm{diag}(\alpha) \in \mathbb{R}^{L\times L}$, $\beta \in \mathbb{R}^L$, and $\mathbf{1}$ is the all-ones vector of length L. We then have

$$\Psi_0^{\mathrm{PC}}(\bar{\epsilon}(t), \dot{\bar{\epsilon}}(t), \{\bar{\epsilon}(\tau)\}_{\tau\in\mathcal{T}}, t; \theta^*) = \mathcal{F}^{\mathrm{PC}}\big(\bar{\epsilon}(t), \dot{\bar{\epsilon}}(t), \xi(t)\big), \tag{2.3.10a}$$
$$\dot{\xi}(t) = \mathcal{G}^{\mathrm{PC}}(\xi(t), \bar{\epsilon}(t)), \quad \xi(0) = 0, \tag{2.3.10b}$$

as in Theorem 2.3.6.

We seek to approximate this map by Ψ_0^{RNO} defined by replacing the linear functions $\mathcal{F}^{\mathrm{PC}}$ and $\mathcal{G}^{\mathrm{PC}}$ by neural networks $\mathcal{F}^{\mathrm{RNO}} : \mathbb{R}\times\mathbb{R}\times\mathbb{R}^L \to \mathbb{R}$ and $\mathcal{G}^{\mathrm{RNO}} : \mathbb{R}^L\times\mathbb{R} \to \mathbb{R}$ to obtain

$$\Psi_0^{\mathrm{RNO}}(\bar{\epsilon}(t), \dot{\bar{\epsilon}}(t), \{\bar{\epsilon}(\tau)\}_{\tau\in\mathcal{T}}, t) = \mathcal{F}^{\mathrm{RNO}}\big(\bar{\epsilon}(t), \dot{\bar{\epsilon}}(t), \xi(t)\big), \tag{2.3.11a}$$
$$\dot{\xi}(t) = \mathcal{G}^{\mathrm{RNO}}(\xi(t), \bar{\epsilon}(t)), \quad \xi(0) = 0. \tag{2.3.11b}$$

Let $R > 0$ and define the bounded set $\mathsf{Z}_R = \{w : \mathbb{R}^+ \to \mathbb{R} \mid \sup_{t\in\mathcal{T}} |w(t)| \le R\}$.

***Theorem* 2.3.10** (RNO Approximation)**.** Consider Ψ_0^{PC} as defined by equations (2.3.9), (2.3.10). Assume that there exist $a_0 > 0$ and $0 \leq B < \infty$ such that $a_0 < \min_\ell |\alpha_\ell|$ and $\max_\ell |\beta_\ell| \leq B$. Then, under Assumptions 2.3.1, for every $\eta > 0$ there exists Ψ_0^{RNO} of the form (2.3.11) such that

$$\sup_{t\in\mathcal{T},b,c\in\mathsf{Z}_R} \left|\Psi_0^{\mathrm{PC}}\big(b(t),c(t),\{b(\tau)\}_{\tau\in\mathcal{T}},t;\theta^*\big) - \Psi_0^{\mathrm{RNO}}\big(b(t),c(t),\{b(\tau)\}_{\tau\in\mathcal{T}},t\big)\right| < \eta.$$

$\Diamond$

The proof of Theorem 2.3.10 can be found in Appendix A.1.

Note that Ψ_0^{RNO} both avoids dependence on the fine-scale ε and is Markovian. The non-homogenized map $\Psi_\varepsilon^\dagger$ is local in time while the homogenized map Ψ_0^{RNO} is nonlocal in time and depends on the strain history. Let u_0^{RNO} be the solution to the following system with constitutive model Ψ_0^{RNO}:

$$\rho\partial_t^2 u_0^{\mathrm{RNO}}(x,t) - \partial_x\sigma_0(x,t) = f(x,t), \quad \partial_x u_0^{\mathrm{RNO}}(x,t) = \bar{\epsilon}^{\mathrm{RNO}}(x,t), \qquad x\in\mathcal{D}, t\in\mathcal{T}, \tag{2.3.12a}$$

$$\sigma_0(x,t) = \Psi_0^{\mathrm{RNO}}\Big(\bar{\epsilon}^{\mathrm{RNO}}(x,t), \dot{\bar{\epsilon}}^{\mathrm{RNO}}(x,t), \big\{\bar{\epsilon}^{\mathrm{RNO}}(x,\tau)\big\}_{\tau\in\mathcal{T}}, t\Big), \qquad x\in\mathcal{D}, t\in\mathcal{T}, \tag{2.3.12b}$$

$$u_0^{\mathrm{RNO}}(x,0) = \partial_t u_0^{\mathrm{RNO}}(x,0) = 0, \qquad x\in\mathcal{D}, \tag{2.3.12c}$$

$$u_0^{\mathrm{RNO}}(0,t) = u_0^{\mathrm{RNO}}(D,t) = 0, \qquad t\in\mathcal{T}. \tag{2.3.12d}$$

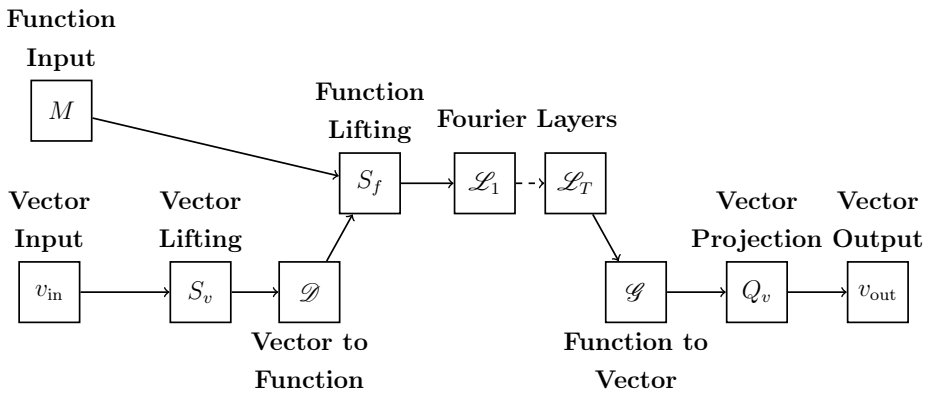
Ideally we would like an approximation result bounding $\|u_\varepsilon - u_0^{\mathrm{RNO}}\|_{\mathcal{Z}_2}$, the difference between solution of the multiscale problem (2.2.1) and the Markovian RNO model (2.3.12), in the case $\rho = 0$. Using Corollary 2.3.9 shows that this would follow from a bound on $\|u_0^{\mathrm{PC}} - u_0^{\mathrm{RNO}}\|_{\mathcal{Z}}$, where u_0^{PC} solves (2.3.5), in the case $\rho = 0$. We note, however, that although Theorem 2.3.10 gives us an approximation result between Ψ_0^{PC} and Ψ_0^{RNO}, proving that u_0^{PC} and u_0^{RNO} are close requires developing new theory for the fully nonlinear PDE for u_0^{RNO}; developing such a theory is beyond the scope of this work. Developing such a theory is difficult for two primary reasons: (i) the monotonicity property of Ψ_0^{RNO} with respect to strain rate is hard to establish globally, for a trained model; (ii) the functions $\mathcal{F}^{\mathrm{RNO}}, \mathcal{G}^{\mathrm{RNO}}$ may not be differentiable. As a result, existence and uniqueness of u_0^{RNO} remain unproven; however, numerical ex-

periments in Section 2.6 indicate that in practice, u_0^{RNO} does approximate u_ε well.

***Remark* 2.3.11.** Monotonicity of Ψ_0^{RNO} with respect to strain rate is a particular issue when $\rho = 0$ (no inertia) as in this case it is needed to define an (implicit) equation for $\partial_t u_0$ to determine the dynamics. It is for this reason that our experiments will all be conducted with $\rho > 0$, obviating the need for the determination of an (implicit) equation for $\partial_t u_0$. However this leads to the issue that the homogenized equation is only valid for a subset of initial conditions, in the inertial setting $\rho > 0$; see Remark 2.2.1. ◊

2.4 Learning material dependence

Up to this point, our model architecture and supporting theory have assumed a single material microstructure. In this subsection, we describe an architecture that allows the model to take varying material microstructures as input. Material dependence is the main contribution of our work in [18], and more details and analysis may be found in that work. To include the material microstructure as an input to the model, we modify $\mathcal{F}^{\text{RNO}}$ and $\mathcal{G}^{\text{RNO}}$ of equations 2.3.11 to be neural operators themselves rather than finite-dimensional neural networks. In particular, $\mathcal{F}^{\text{RNO}}$ and $\mathcal{G}^{\text{RNO}}$ take a form similar to the FNO in definition 1.3.1 where the function inputs are the material properties E and ν defined on $\mathbb{T}^d$. However, the architecture is modified to also allow the finite vector inputs of $\overline{\epsilon}(x,t), \dot{\overline{\epsilon}}(x,t)$, and $\xi(t)$ and finite vector outputs of $\dot{\xi}(t)$ and $\overline{\sigma}(t)$, which are not functions over $\mathbb{T}^d$. The modified architecture is called a Fourier Neural Mapping and is developed in Chapter 5. We present a diagram below and include the detailed definition in Appendix A.5. In combination, the architecture used to model material and history dependence simultaneously is referred to as an RNO-FNM.



This modified architecture inherits the universal approximation property of Theorem 2.3.10 with varying material inputs over compact sets in $\mathcal{M}^B_{E_{\min},E_{\max}}$ and $\mathcal{M}^B_{\nu_{\min},\nu_{\max}}$, respectively. We do not include the formal statement or proof in this thesis, but it may be found in [18]. A material-dependent experiment is included in Section 2.6 to demonstrate success of this extension.

2.5 Numerical experiments: data and optimization

In this section, we present numerical results using a trained RNO as a *surrogate model*: an efficient approximation of the complex microscale dynamics. First, we discuss the problem of finding such an RNO. To learn the RNO operator approximation, we are given data

$$\{\bar{\epsilon}_n, \dot{\bar{\epsilon}}_n, (\sigma_0)_n\}_{n=1}^N, \tag{2.5.1}$$

where the suffix n denotes the n^{th} strain, strain rate, and stress trajectories over the entire time interval $\mathcal{T}$. Each strain trajectory $\bar{\epsilon}_n$ is drawn i.i.d. from a measure μ on $\mathcal{C}(\mathcal{T};\mathbb{R})$.

The data for the homogenized constitutive model is given by

$$\sigma_0(t) = \Psi_0^\dagger\big(\bar{\epsilon}(t), \dot{\bar{\epsilon}}(t), \{\bar{\epsilon}(\tau)\}_{\tau\in\mathcal{T}}, t\big),$$

defined via solution of the cell-problem (2.2.4); but it may also be obtained as the solution to a forced boundary problem on the microscale, as stated in the following lemma.

***Lemma* 2.5.1.** Let $\Omega = (0,1)$, and let σ be determined by the following equations, where E, ν, and b are given:

$$\begin{aligned}
\partial_y \sigma(y,t) &= 0, && y\in\Omega, t\in\mathcal{T}, && \text{(2.5.2a)}\\
\sigma(y,t) &= E(y)\partial_y u(y,t) + \nu(y)\partial^2_{yt}u(y,t), && y\in\Omega, t\in\mathcal{T}, && \text{(2.5.2b)}\\
u(0,t) &= 0, \quad u(1,t) = b(t), && t\in\mathcal{T}, && \text{(2.5.2c)}\\
u(y,0) &= 0, && y\in\Omega. && \text{(2.5.2d)}
\end{aligned}$$

Then

$$\{\sigma(t)\}_{t\in\mathcal{T}} = \Psi_0^\dagger(b(t), \partial_t b(t), \{b(t)\}_{t\in\mathcal{T}}, t),$$

where $\Psi_0^\dagger$ is the map defined in (2.2.6). ◊

The proof can be found in Appendix A.2 and justifies the application of data resulting from this problem to the homogenized model. In the following, we

denote by $(\widehat{\sigma}_0)_n$ and $\widehat{\dot{\xi}}_n$ the output of $\mathcal{F}^{\text{RNO}}$ and $\mathcal{G}^{\text{RNO}}$ on data point n over $t \in \mathcal{T}$:

$$(\widehat{\sigma}_0)_n(t) = \mathcal{F}^{\text{RNO}}\Big(\bar{\epsilon}_n(t), \dot{\bar{\epsilon}}_n(t), \widehat{\xi}_n(t)\Big)$$
$$\widehat{\dot{\xi}}_n(t) = \mathcal{G}^{\text{RNO}}\Big(\bar{\epsilon}_n(t), \widehat{\xi}_n(t)\Big), \quad \widehat{\xi}_n(0) = 0.$$

To train the RNO, we use the following relative L^2 loss function, which should be viewed as a function of the parameters defining the neural networks $\mathcal{F}^{\text{RNO}}$ and $\mathcal{G}^{\text{RNO}}$.

Accessible Loss Function:

$$\mathsf{Loss}(\{\sigma_0\}_{n=1}^N, \{\widehat{\sigma}_0\}_{n=1}^N) = \frac{1}{N}\sum_{n=1}^{N} \frac{\|(\sigma_0)_n - (\widehat{\sigma}_0)_n\|_{L^2(\mathcal{T};\mathbb{R})}}{\|(\sigma_0)_n\|_{L^2(\mathcal{T};\mathbb{R})}}. \tag{2.5.3}$$

***Remark* 2.5.2.** To test robustness of our conclusions, we also employed relative and absolute L^2 squared loss functions. In doing so we did not observe significant differences in the predictive accuracy of the resulting models. ◊

In the case of a material that is 2-piecewise constant on the microscale, we can explicitly write down the analytic form of the solution, and thus can also know the values of the hidden variable $\{\xi_n\}_{n=1}^N$ and its derivative $\{\dot{\xi}_n\}_{n=1}^N$, for each data trajectory as expressed in equation (2.3.10). It is intuitive that training an RNO on an extended data set which includes the hidden variable should be easier than using the original data set (2.5.1). In order to deepen our understanding of the training process we will include training on a 2-piecewise-constant material which uses this hidden data, motivating the following loss function. Since, in general, the hidden variable is inaccessible in the data, we refer to the resulting loss as the inaccessible relative loss function.

Inaccessible Loss Function:

$$\mathsf{Loss}(\{(\sigma_0)_n\}_{n=1}^N, \{(\widehat{\sigma}_0)_n\}_{n=1}^N, \{\dot{\xi}_n\}_{n=1}^N, \{\widehat{\dot{\xi}}_n\}_{n=1}^N)$$
$$= \frac{1}{N}\sum_{n=1}^{N}\left(\frac{\|(\sigma_0)_n - (\widehat{\sigma}_0)_n\|_{L^2(\mathcal{T};\mathbb{R})}}{\|(\sigma_0)_n\|_{L^2(\mathcal{T};\mathbb{R})}} + \frac{\|\dot{\xi}_n - \widehat{\dot{\xi}}_n\|_{L^2(\mathcal{T};\mathbb{R})}}{\|\dot{\xi}_n\|_{L^2(\mathcal{T};\mathbb{R})}}\right).$$

2.6 Numerical Results

The numerical results make the following contributions:

I. **Machine-Learned Constitutive Models.** We can find RNOs that yield low-error simulations when used as a surrogate model in the macroscopic system (2.1.6) to approximate the multiscale system (2.1.5), in the one-dimensional KV setting with inertia. We also discuss how in some material parameter settings, inertial effects lead to higher error in the homogenized approximation.

II. **Effect of Non-Convex Optimization.** When the inaccessible loss function is used for training, the trained RNO exhibits desirable properties of (approximate) linearity in its arguments in the domain of interest, as is proved for the homogenized constitutive model (2.3.9) for piecewise-constant materials. When using the accessible loss function, the trained RNO may perform well as a surrogate model without exhibiting linearity in the equation for evolution of the hidden variables. This is attributable to the existence of local minimizers of the loss function and highlights the need for caution in training constitutive models.

III. **Elasto-viscoplasticity.** We demonstrate success of the RNO model in approximating the homogenized constitutive law for a one-dimensional elasto-viscoplastic material as well. The discretization invariance figure for this example is taken from our work in [16], where the method is tested in two-dimensional elasto-viscoplastic materials and two- and three-dimensional elasto-viscoplastic polycrystalline materials.

IV. **Model Choice.** The correct choice of architecture for the RNO leads to discretization robustness in time: a model learned with one choice of time discretization dt performs well when tested on another dt; this is not true for poor model choices. Discretization robustness can thus be used as a guide to model choice. This observation holds for both the one-dimensional viscoelasticity example and the one-dimensional elasto-viscoplasticity example.

V. **Material dependence**. We include an experiment from [18] that shows accuracy of the RNO-FNM method in modeling the effects of both history dependence and material dependence simultaneously. We also include the result of a macroscale simulation with material dependence.

In the remainder of the section, we demonstrate that in appropriate settings the solution u_0^{RNO} obtained under the dynamics of a trained RNO approximates the true solution u_ε well when used in the macroscopic setting; furthermore, this RNO is shown to exhibit linearity in its arguments in the domain of interest. We also discuss the error arising from inertial effects. We then discuss the performance of an RNO learned using the accessible loss function, the discretization-robustness property of the RNO, and the choice of data sampling distribution μ. We also perform experiments on media with more piecewise-constant pieces and analyze the number of hidden variables required to capture the behavior. We then apply the methodology to the case of elasto-viscoplasticity. Finally, we include a material-dependent experiment using the modified RNO-FNM model architecture.

RNO as a Surrogate Model

In this subsection we discuss two RNOs: RNO A trained using only the inaccessible loss function in equation (2.5) and RNO B trained with the standard loss function in equation (2.5.3), but initialized at parameters obtained via training with the inaccessible loss function. Descriptions of these RNOs, and others we introduce in subsequent subsections, may be found in Table 2.1. A visualization of typical input and output trajectories from the data used for training and testing may be found in Figure 2.1. For details on RNO training, see Appendix A.3.

In the first surrogate model experiment, we subject the material to sinusoidal boundary forcing of $b(t) = 0.1\sin(2\pi t)$ starting from 0 initial displacement and velocity. As a ground-truth comparison, we use a traditional finite element solver with periodic domain of width 0.04, spatial resolution of $h = 0.005$,

Table 2.1: RNO Descriptions

RNO	Description
A	Trained on 2-piecewise-constant media only with inaccessible loss function (2.5)
B	Trained on 2-piecewise-constant media; initialized at solution found with inaccessible loss function then trained with accessible loss function (2.5.3)
C	Trained on 2-piecewise-constant media only with accessible loss function
D	Trained on continuous media with accessible loss function

and time discretization $dt = 0.1h^2$; we refer to this solution as u_ε and name it FEM. In contrast, the RNO-based macroscale computation employs spatial resolution of $h_{cell} = 0.04$, with a time discretization of $dt = 0.4h_{cell}^2$; for economy of notation; this solution is denoted u_0^{RNO} and named RNO. We also compare the results to the displacement obtained using as macroscale constitutive model the analytic solution to the cell problem. To make comparisons we use the relative error given by

$$e(u_0^{\text{RNO}}, u_\varepsilon)(t) = \frac{\|u_\varepsilon(t) - u_0^{\text{RNO}}(t)\|_{L^2(\Omega;\mathbb{R})}}{\|u_\varepsilon(t)\|_{L^2(\Omega;\mathbb{R})} + 0.01}. \tag{2.6.1}$$

The relative error plots for RNOs A and B are shown in Figures 2.2a and 2.2b. In a second experiment, we subject the material to integrated Brownian motion forcing starting from null initial conditions. The FEM solver uses the same discretizations as in the sinusoidal forcing experiment, and the RNO spatial discretization was $h_{cell} = 0.05$ with time discretization of $dt = 0.4h_{cell}^2$. The results for RNOs A and B are shown in the Appendix in Figures A.3a and A.3b.

Both sets of experiments show that the RNO-based macroscopic models accurately reproduce the microscale FEM simulation at far lower computational cost. The RNO-based results have some errors in comparison with the microscale simulation, but the errors are of the same order of magnitude as the errors arising when the exact homogenized constitutive model is used. The initial error between the analytic solution and the FEM solution is due to inertial effects discussed in Remark 2.2.1. The inertial errors become more significant with the ratio between E and ν varies more across the interval.

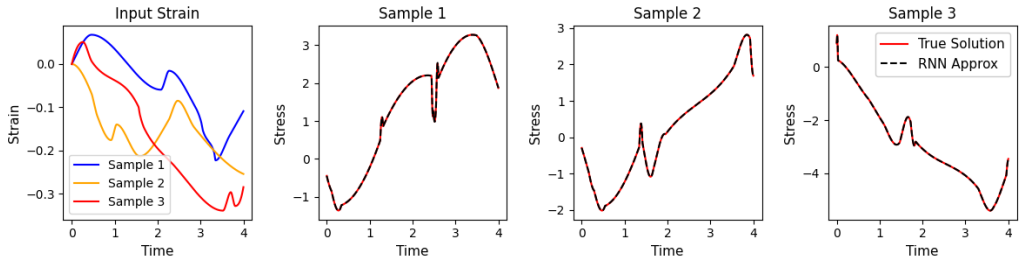
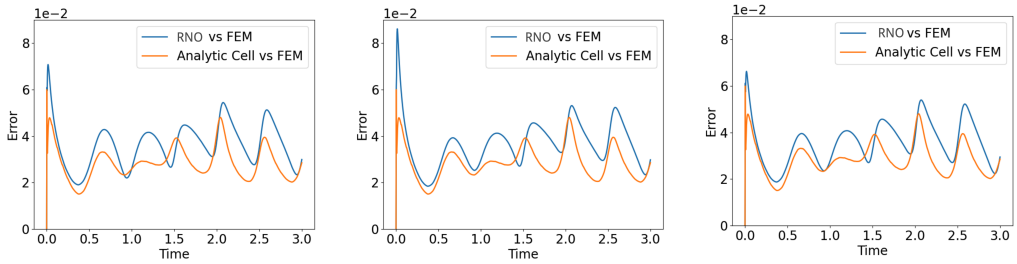


Figure 2.1: Representative data: input strain trajectories and output stress trajectories for three randomly chosen test data samples. The RNO approximation shown was generated with RNO C.



(a) RNO A (b) RNO B (c) RNO C

Figure 2.2: Analytic cell and RNO relative error versus FEM solution using sinusoidal forcing; this supports Numerical Experiments, conclusion I.

RNO Trained with Standard Loss

We also train a third RNO, denoted RNO C, using only the accessible loss function. Details of training may be found in Appendix A.3. While this RNO performs well as a surrogate model, indeed is comparable in errors to those of RNOs A and B, it does not exhibit a close linear match to the known analytic expression for $\mathcal{G}$. A figure illustrating this behavior may be found in the appendix as Figure A.2. All three RNOs approximate the linear structure of $\mathcal{F}$ well; the difficulty is in obtaining the correct linear dependence in the hidden variable rate, $\dot{\xi}$. Interestingly, by changing the material parameter ν_2 from 0.2 to 2, training via the method of RNO C with only the accessible loss function yields an RNO that matches the true linear dependence in $\mathcal{G}$ very well. However, in this parameter regime, inertial effects perturb the simulations on the macroscale to an unacceptable degree, meaning that the homogenization theory that we use as benchmark is not valid, and so we avoid this regime. The inability of RNO C to capture the exact linear dependence in $\mathcal{G}$ is unsurprising; indeed, had we guaranteed convergence to the optimal function for any choice of material parameters, we would have entirely sidestepped the problem of high-dimensional optimization inherent to machine learning.

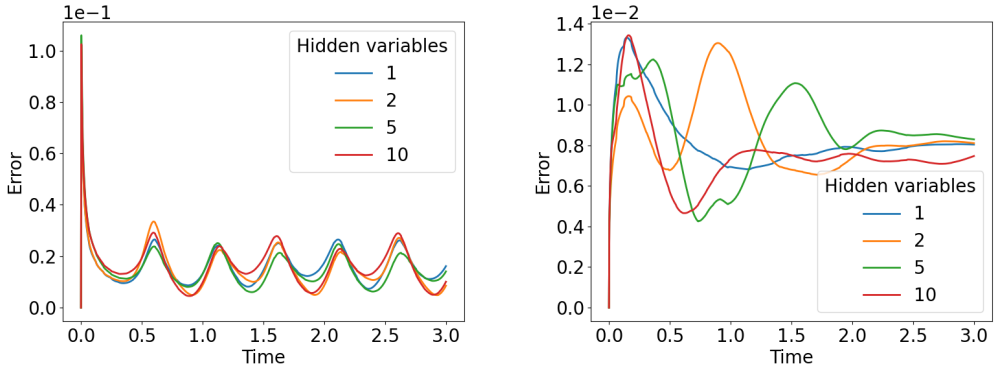
In the case of continuous material properties, we do not have a known analytic solution to the microscale problem and thus do not have access to the hidden variable ξ in the train and test data; in this case, we may only use the accessible loss function. We RNO type D on continuous media with different numbers of hidden variables and use the trained RNOs as surrogate models in the macroscale system subjected to boundary forcing. Training details may be found in Appendix A.3. The relative error of RNO D for the sinusoidal and Brownian motion forcing experiments described previously is shown in Figure

2.3. In this figure, the ground truth is obtained via an FEM simulation; details are in Appendix A.3. We note that similar error is found with all dimensions of the hidden variable, suggesting that 1 hidden variable suffices in this case; the fact that the error does not decrease suggests that the error we see is primarily from the effects of homogenization in the macroscale simulation rather than piecewise-constant approximation. This conclusion agrees with the result of Figure 2.2 that the RNO surrogate model error is the same magnitude as that of the exact homogenized solution.

Time Discretization and RNO Training

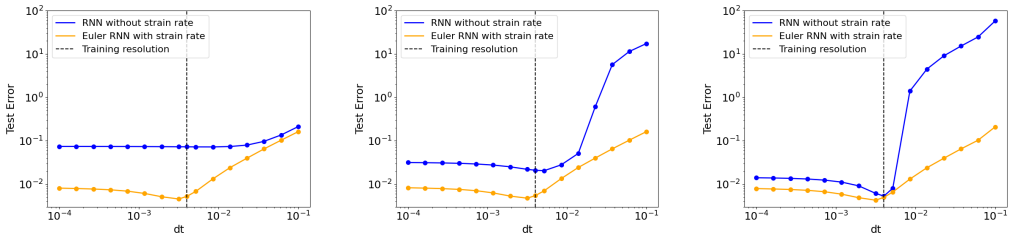
Discretization-robustness is a desirable feature of an RNO surrogate model. To test robustness to changes in time discretization we work in the piecewise-constant media setting. We evaluate the test error when using RNOs A, B, and C using values of time step dt different from those used in the training. Additionally, to demonstrate the value of postulating the correct model form, we train three additional RNOs via the same methods as described earlier in this section but without giving them strain rate as an input, leading to an *incorrect* model form. Figure 2.4 shows that all three RNOs trained with strain rate as an input parameter were more robust to changes in time discretization than their non-strain-rate counterparts, supporting the conclusion that this approach can aid model discovery.

To generate the training strain, we sampled trajectories as follows: first, we randomly partitioned the time interval $\mathcal{T}$ into 10 pieces; second, at each point



(a) Sinusoidal forcing (b) Integrated Brownian motion forcing

Figure 2.3: Relative error of continuous-material RNOs D with different numbers of hidden variables when used as a surrogate model in the macroscale system; this supports Numerical Experiments, conclusion I.



(a) RNO A (b) RNO B (c) RNO C

Figure 2.4: Time discretization error for RNOs A, B, and C.

between these time intervals, we generated a value of strain via a balanced random walk from the previous value scaled by the length of the time interval; third, we used a piecewise cubic Hermite interpolating polynomial (pchips) function to interpolate between these values of strain. This choice of distribution has the desirable property that it generates data with a variety of strain/strain-rate pairings evenly dispersed throughout the domain of interest rather than introducing large correlations between the two.

Additional Piecewise-Constant Experiments

We claim that to approximate an N-piecewise-constant material, the RNO ought to have at least $N-1$ hidden variables to achieve the best accuracy. Therefore, we train RNOs with different numbers of hidden variables on data from piecewise-constant materials with 3, 5, and 10 pieces. The results are shown in Figure 2.5.

For the 3 and 5-piecewise-constant cases, the error flattens out after 2 and 4 hidden variables, respectively, as expected. For the 10-piecewise constant case, the error plateaus first at 4 hidden variables and then again at 9 hidden variables, and this can be explained by examining the analytic solution. For the choices of E and ν used, the constitutive law takes the approximate form of

$$\begin{aligned}\sigma_0(t) = E'\partial_x u_0(t) + \nu'\partial_t\partial_x u_0(t) - \int_0^t \partial_x u_0(\tau)\Big(&0.09e^{-1.83(t-\tau)} + 0.16e^{-2.92(t-\tau)} \\ &+ 0.02e^{-4.44(t-\tau)} + 0.20e^{-5.13(t-\tau)} + 0.12e^{-8.18(t-\tau)} + 0.08e^{-9.29(t-\tau)} \\ &+ 0.39e^{-11.3(t-\tau)} + 0.67e^{-15.30(t-\tau)} + 0.06e^{-18.41(t-\tau)}\Big)\, d\tau.\end{aligned}$$

Note that the exponential decay terms each correspond to one of the nine hidden variables ξ_ℓ, and they are written in order of decreasing exponential

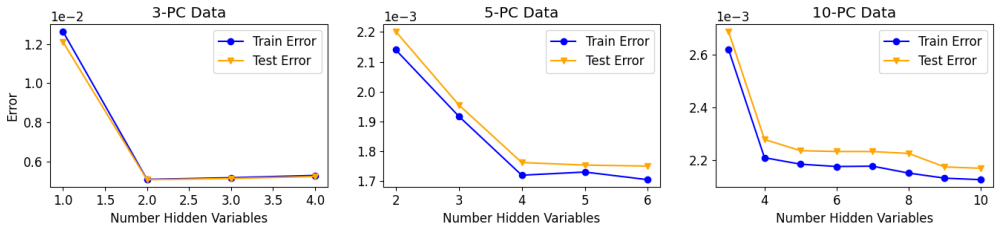


Figure 2.5: Absolute L^2 error of RNOs trained with different numbers of hidden variables on different piecewise-constant materials.

term $-\alpha_\ell$. From this we can see that terms with higher magnitudes of $|\alpha_\ell|$ will be negligible compared to the terms with smaller magnitude. The experimental results align with these values; there is a large jump from the fourth exponent (-5.13) to the fifth exponent (-8.18), so the behavior is well-captured with only four hidden variables. However, with nine hidden variables, the model can completely capture the decay terms. This result further justifies the practical use of the piecewise-constant approximation for smooth materials.

Elasto-viscoplasticity

The purpose of this example is to demonstrate that the ideas developed in this work have implications beyond linear viscoelasticity. The same RNO architecture is able to learn elastic-viscoplastic dynamics. We present the results of a simple experiment with isotropic rate hardening in one spatial dimension.

Consider the following equations:

$$\partial_x \sigma_\varepsilon(x,t) = 0, \quad \sigma_\varepsilon(x,t) = E_\varepsilon(x)(\partial_x u_\varepsilon(x,t) - \epsilon_p(x,t)), \quad (x,t) \in \mathcal{D} \times \mathcal{T}, \tag{2.6.2a}$$

$$\dot{\epsilon}_p(x,t) = \dot{\epsilon}_{p0}\,\mathrm{sign}(\sigma_\varepsilon(x,t))\left(\frac{|\sigma_\varepsilon(x,t)|^n}{\sigma_h}\right), \quad (x,t) \in \mathcal{D} \times \mathcal{T}, \tag{2.6.2b}$$

$$u_\varepsilon(0,t) = 0, u_\varepsilon(1,t) = b(t), \quad t \in \mathcal{T}, \tag{2.6.2c}$$

$$u_\varepsilon(x,0) = 0, \epsilon_p(x,0) = 0 \quad x \in \mathcal{D}, \tag{2.6.2d}$$

where u_ϵ is the displacement, ϵ_p is the plastic strain, σ_ϵ is the stress, and $\dot{\epsilon}_{p0}$, σ_h, and n are constants. We seek to learn the map from average strain to average stress $(\langle \partial_x u_\varepsilon(t) \rangle)_{t\in\mathcal{T}} \to (\langle \sigma_\varepsilon(t) \rangle)_{t\in\mathcal{T}}$, where we recall that $\langle \cdot \rangle$ indicates spatial averaging over the cell. In this setting, the homogenized constitutive

map does not depend on the strain rate. Development of the homogenization theory for these equations can be found in [16].

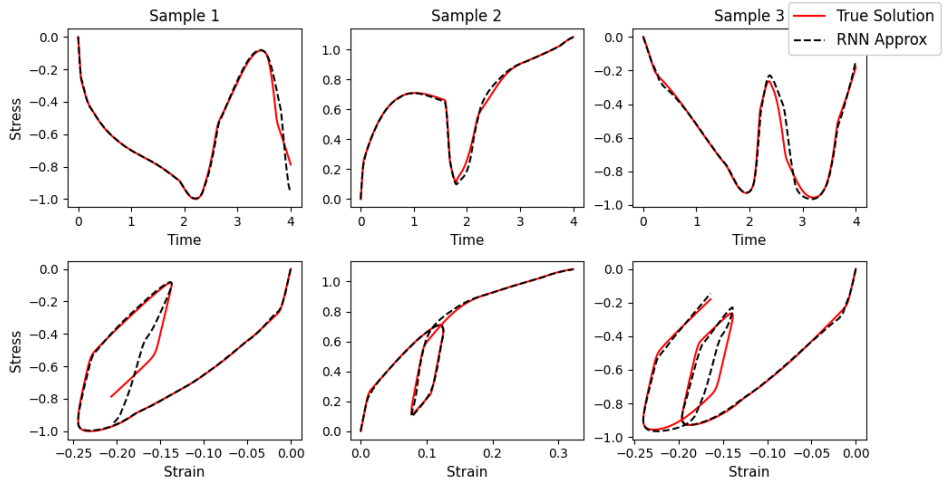


Figure 2.6: RNO trained on elasto-viscoplastic data; a comparison between the true solution and the RNO-predicted solution for three random test samples. Top row: stress trajectories in time. Bottom row: stress-strain trajectories for the same samples.

We train an RNO with one hidden variable and the same architecture as prescribed for the viscoelastic case but without strain rate dependence, and we use data generated via a numerical solution of equations (2.6.2). As shown in Figure 2.6, the RNO is able to learn plastic behavior. Specifically, the strain-stress trajectories exhibit plastic transition. The mean relative L^2 error is $\approx 7\%$, which is reasonable for plasticity experiments. When a discretization-robustness test is done on this example, greater discretization-invariance is again seen with the correct model hypothesis. In Figure 2.7, the "VE" model includes strain rate as an input variable, while the "E-VP" model does not. The "E-VP" is the correct model form in this setting, and it exhibits greater robustness to changes in test discretization.

Material dependence

As a final discussion, we include an experiment from [18] to show that the material-dependent extension described in Section 2.4 is effective in practice. First, a dataset of piecewise-constant material properties (E, ν) are randomly generated in the following manner. A number of constant pieces L are selected uniformly at random from $5-20$. Locations of the discontinuities are randomly selected from the set $\{0.02k\}_{k=0}^{50}$ with replacement. Finally, the values of E

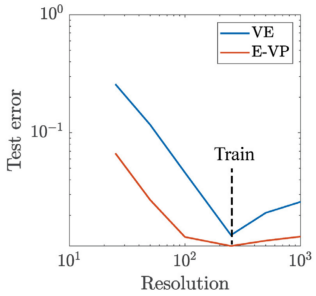


Figure 2.7: Discretiztion-invariance experiment for one-dimensional elasto-viscoplasticity. In the "VE" model, strain rate is given as an input, while it is not given in the "E-VP" model.

and ν on each segment are sampled from $\mathcal{U}([0.1, 1])$. Piecewise-cubic Hermite interpolating polynomials are again used to generate random averaged strain trajectories. Figure 2.8 shows the relative L^2 error of the RNO-FNM model for different choices of hidden variable count. The error varies between $0.7\% - 1.2\%$, indicating accuracy of the model despite the variation in material input. A discretization-robustness test is also shown in Figure 2.8, and the model exhibits discretization-invariance in both the space and the time discretization. We refer to [18] for additional material-dependent experiments.

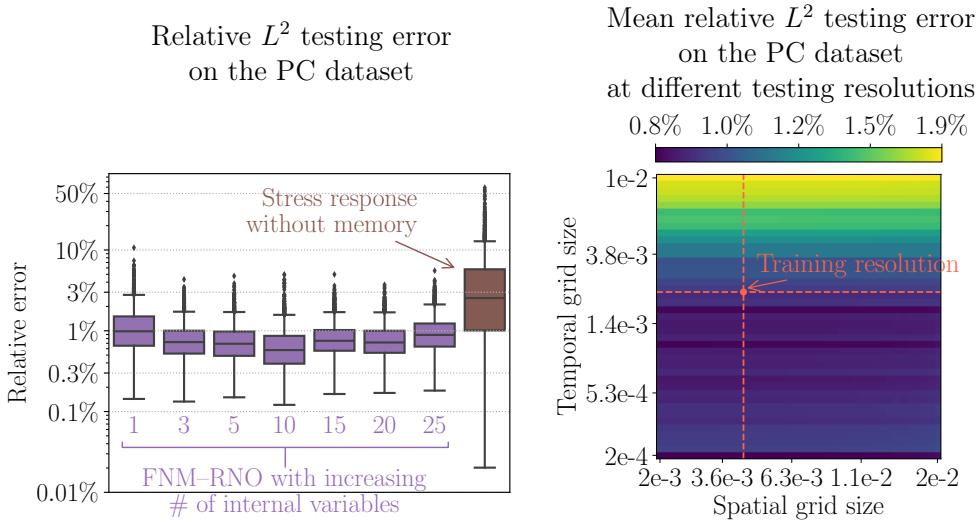


Figure 2.8: The distributions of the relative L^2 error on 2,500 testing samples from the PC dataset (*left*). We visualize the errors in FNM–RNOs predictions where the trained FNM–RNOs have a varying number of internal variables. We also visualize the distribution of error given by the linear stress response without memory effects, where the response function is obtained using (2.2.7) with $\kappa \equiv 0$. The mean relative L^2 error for the same dataset is on the right for different testing resolutions with five internal variables.

2.7 Conclusions

In this chapter, we develop theory to support learning Markovian models for history-dependent constitutive laws. The theory presented applies to the one-dimensional KV case, but the underlying ideas extend to more complex systems, as demonstrated with an experiment with elasto-viscoplasticity. Furthermore, an extension of the method to include material dependence is shown to be empirically accurate in the one-dimensional viscoelastic setting. In [16], numerical experiments suggest that the methodology can be useful in higher dimensions as well. Conclusions drawn from our numerical experiments, underpinned by the theory of this chapter, provide useful guidance for these more complex nonlinear models in higher spatial dimensions.

Chapter 3

LEARNING HOMOGENIZATION FOR ELLIPTIC OPERATORS

This chapter is adapted from the following publication:

[1] Kaushik Bhattacharya, Nikola B. Kovachki, Aakila Rajan, Andrew M. Stuart, and Margaret Trautner. "Learning homogenization for elliptic operators". In: *SIAM Journal on Numerical Analysis* 62.4 (2024), pp. 1844–1873. DOI: 10.1137/23M1585015.

Multiscale partial differential equations (PDEs) arise in various applications, and several schemes have been developed to solve them efficiently. Homogenization theory is a powerful methodology that eliminates the small-scale dependence, resulting in simplified equations that are computationally tractable while accurately predicting the macroscopic response. In the field of continuum mechanics, homogenization is crucial for deriving constitutive laws that incorporate microscale physics in order to formulate balance laws for the macroscopic quantities of interest. However, obtaining homogenized constitutive laws is often challenging as they do not in general have an analytic form and can exhibit phenomena not present on the microscale. In response, data-driven learning of the constitutive law has been proposed as appropriate for this task. However, a major challenge in data-driven learning approaches for this problem has remained unexplored: the impact of discontinuities and corner interfaces in the underlying material. These discontinuities in the coefficients affect the smoothness of the solutions of the underlying equations. Given the prevalence of discontinuous materials in continuum mechanics applications, it is important to address the challenge of learning in this context; in particular, to develop underpinning theory that establishes the reliability of data-driven methods in this scientific domain. The chapter addresses this unexplored challenge by investigating the learnability of homogenized constitutive laws for elliptic operators in the presence of such complexities. Approximation theory is presented, and numerical experiments are performed which validate the theory in the context of learning the solution operator defined by the cell problem arising in homogenization for elliptic PDEs.

3.1 Introduction

Homogenization theory is a well-established methodology that aims to eliminate fast-scale dependence in partial differential equations (PDEs) to obtain homogenized PDEs which produce a good approximate solution of the problem with small scales while being more computationally tractable. In continuum mechanics, this methodology is of great practical importance as the constitutive laws derived from physical principles are governed by material behavior at small scales, but the quantities of interest are often relevant on larger scales. These homogenized constitutive laws often do not have a closed analytic form and may have new features not present in the microscale laws. Consequently, there has been a recent surge of interest in employing data-driven methods to learn homogenized constitutive laws.

The goal of this chapter is to study the learnability of homogenized constitutive laws in the context of one of the canonical model problems of homogenization: the divergence form elliptic PDE. One significant challenge in applications of homogenization in material science arises from the presence of discontinuities and corner interfaces in the underlying material. This leads to a lack of smoothness in the coefficients and solutions of the associated equations, a phenomenon extensively studied in numerical methods for PDEs. Addressing this challenge in the context of learning remains largely unexplored and is the focus of our work. We develop underlying theory and provide accompanying numerical studies to address learnability in this context.

Problem Formulation

Consider the following linear multiscale elliptic equation on a bounded domain $\Omega \subset \mathbb{R}^d$:

$$-\nabla_x \cdot (A^\epsilon \nabla_x u^\epsilon) = f \quad x \in \Omega, \tag{3.1.1a}$$

$$u^\epsilon = 0 \quad x \in \partial\Omega. \tag{3.1.1b}$$

Here $A^\epsilon(x) = A\left(\frac{x}{\epsilon}\right)$ for $A(\cdot)$ which is 1-periodic and positive definite: $A : \mathbb{T}^d \to \mathbb{R}^{d\times d}_{\text{sym},\succ 0}$, a condition which holds throughout this work. Assume further that $f \in L^2(\Omega;\mathbb{R})$ and has no microscale variation with respect to x/ϵ.

Our focus is on linking this multiscale problem to the homogenized form of

equation (3.1.1), which is

$$-\nabla_x \cdot \left(\overline{A}\nabla_x u\right) = f \quad x \in \Omega, \tag{3.1.2a}$$

$$u = 0 \quad x \in \partial\Omega, \tag{3.1.2b}$$

where $\overline{A}$ is given by

$$\overline{A} = \int_{\mathbb{T}^d} \left(A(y) + A(y)\nabla\chi(y)^T\right) \, \mathsf{d}y, \tag{3.1.3}$$

and $\chi : \mathbb{T}^d \to \mathbb{R}^d$ solves the cell problem

$$-\nabla \cdot (\nabla\chi A) = \nabla \cdot A, \quad \chi \text{ is 1-periodic.} \tag{3.1.4}$$

All of the preceding PDEs are to be interpreted as holding in the weak sense. For $0 < \epsilon \ll 1$, the solution u^ϵ of (3.1.1) is approximated by the solution u of (3.1.2), and the error converges to zero as $\epsilon \to 0$ in various topologies [26, 59, 27].

We assume that

$$\|A\|_{L^\infty} := \sup_{y\in\mathbb{T}^d} |A(y)|_F < \infty,$$

where $|\cdot|_F$ is the Frobenius norm. Hence $A \in L^\infty\left(\mathbb{T}^d; \mathbb{R}^{d\times d}\right)$ and $A^\epsilon \in L^\infty\left(\Omega; \mathbb{R}^{d\times d}\right)$. Similarly, for $A \in L^2(\mathbb{T}^d; \mathbb{R}^{d\times d})$, we define

$$\|A\|_{L^2}^2 := \int_{\mathbb{T}^d} |A(y)|_F^2 \, \mathsf{d}y.$$

Also, for given $\beta \geq \alpha > 0$, we define the following subset of 1-periodic, positive-definite, symmetric matrix fields in $L^\infty\left(\mathbb{T}^d; \mathbb{R}^{d\times d}\right)$ by

$$\mathsf{PD}_{\alpha,\beta} = \{A \in L^\infty(\mathbb{T}^d; \mathbb{R}^{d\times d}) : \ \forall (y,\xi) \in \mathbb{T}^d \times \mathbb{R}^d, \ \alpha|\xi|^2 \leq \langle \xi, A(y)\xi\rangle \leq \beta|\xi|^2\}.$$

For open set $\Omega \subset \mathbb{R}^d$, we denote the *variation* of a function $u \in L^1_{\text{loc}}(\Omega)$ by

$$V(u,\Omega) = \sup\left\{ \sum_{i=1}^d \int_\Omega \frac{\partial \Phi_i}{\partial x_i} u \, \mathsf{d}x : \ \Phi \in C_0^\infty(\Omega; \mathbb{R}^d), \ \|\Phi\|_{L^\infty(\Omega;\mathbb{R}^d)} \leq 1 \right\}$$

and the set of functions of bounded variation on $\mathbb{T}^d$ as

$$\text{BV} = \{u \in L^1(\mathbb{T}^d) : \ V(u, \mathbb{T}^d) < \infty\}.$$

For further information on BV, we refer to [57]. Finally, we often work in the Sobolev space H^1 restricted to spatially mean-zero periodic functions, denoted

$$\dot{H}^1 := \left\{ v \in W^{1,2}(\mathbb{T}^d) \;\middle|\; v \text{ is 1-periodic}, \int_{\mathbb{T}^d} v \ \mathsf{d}y = 0 \right\};$$

the norm on this space is defined by

$$\|g\|_{\dot{H}^1} := \|\nabla g\|_{L^2}. \tag{3.1.5}$$

Numerically solving (3.1.1) is far more computationally expensive than solving the homogenized equation (3.1.2), motivating the wish to find the homogenized coefficient $\overline{A}$ defining equation (3.1.2). The difficult part of obtaining the equation (3.1.2) is solving the cell problem (3.1.4). Although explicit solutions exist in the one-dimensional setting for piecewise constant A [56] and in the two-dimensional setting where A is a layered material [27], in general a closed form solution is not available and the cell problem must be solved numerically. Note that in general the action of the divergence $\nabla\cdot$ on terms involving A in the cell problem necessitates the use of weak solutions for $A \notin C^1(\mathbb{T}^d, \mathbb{R}^{d\times d})$; this is a commonly occurring situation in applications such as those arising from porous medium flow, or to vector-valued generalizations of the setting here to elasticity, rendering the numerical solution non-trivial. For this reason, it is potentially valuable to approximate the solution map

$$G: \ A \mapsto \chi, \tag{3.1.6}$$

defined by the cell problem, using a map defined by a neural operator. More generally it is foundational to the broader program of learning homogenized constitutive models from data to thoroughly study this issue for the divergence form elliptic equation as the insights gained will be important for understanding the learning of more complex parameterized homogenized models, such as those arising in nonlinear elasticity, viscoelasticity, and plasticity.

The full map from A to the homogenized tensor $\overline{A}$ is expressed by $A \mapsto (\chi, A) \mapsto \overline{A}$, and one could instead learn the map

$$F : A \mapsto \overline{A}. \tag{3.1.7}$$

Since the map $(\chi, A) \mapsto \overline{A}$ is is defined by a quadrature, we focus on the approximation of $A \mapsto \chi$ and state equivalent results for the map $A \mapsto \overline{A}$ that emerge as consequences of the approximation of χ. Direct learning of $\overline{A}$ is addressed in Chapter 5. In this work we make the following contributions:

1. We state and prove universal approximation theorems for the map G defined by (3.1.4) and (3.1.6), and map F defined by (3.1.3), (3.1.4), and (3.1.7).

2. We provide explicit examples of microstructures which satisfy the hypotheses of our theorems; these include microstructures generated by probability measures which generate discontinuous functions in BV.

3. We provide numerical experiments to demonstrate the ability of neural operators to approximate the solution map on four different classes of material parameters A, all covered by our theoretical setting.

We provide an overview of the literature followed by a discussion of stability estimates for (3.1.4), with respect to variations in A; these are at the heart of the analysis of universal approximation. The main body of the text then commences with Section 3.2, which characterizes the microstructures of interest to us in the context of continuum mechanics. Section 3.3 states universal approximation theorems for $G(\cdot)$ and $F(\cdot)$, using the Fourier neural operator. In Section 3.4 we give numerical experiments illustrating the approximation of the map G defined by (3.1.6) on microstructures of interest in continuum mechanics. Details of the stability estimates, the proofs of universal approximation theorems, properties of the microstructures, and details of numerical experiments are given in Appendices B.1, B.2, B.3, and B.4 respectively.

Literature Review

Homogenization aims to derive macroscopic equations that describe the effective properties and behavior of solutions to problems at larger scales given a system that exhibits behavior at one or more smaller scales. Although it is developed for the various cases of random, statistically stationary, and periodic small-scale structures, we work here entirely in the periodic setting. The underlying assumption of periodic homogenization theory is that the coefficient is periodic in the small-scale variable, and that the scale separation is large compared to the macroscopic scales of interest. Convergence of the solution of the multiscale problem to the homogenized solution is well studied; see [41, 40]. We refer to the texts [26, 59, 27] for more comprehensive citations to the literature. Homogenization has found extensive application in the setting of continuum mechanics [60] where, for many multiscale materials, the scale-separation assumption is natural. In this work, we are motivated in part by learning constitutive models for solid materials, where crystalline microstructure renders the material parameters discontinuous and may include corner interfaces. This difficulty has been explored extensively in the context of nu-

merical methods for PDEs, particularly with adaptive finite element methods [61, 62, 63, 64].

There is a significant body of work on the approximation theory associated with parametrically dependent solutions of PDEs, including viewing these solution as a map between the function space of the parameter and the function space of the solution, especially for problems possessing holomorphic regularity [65, 66, 67]. This work could potentially be used to study the cell problem for homogenization that is our focus here. However, there has been recent interest in taking a data-driven approach to solving PDEs via machine learning because of its flexibility and ease of implementation. A particular approach to learning solutions to PDEs is operator learning, a machine learning methodology where the map to be learned is viewed as an operator acting between infinite-dimensional function spaces rather than between finite-dimensional spaces [68, 19, 23, 69, 52]. Determining whether, and then when, operator learning models have advantages over classical numerical methods in solving PDEs remains an active area of research [70]. The paper [71] makes a contribution to this area, in the context of the divergence form elliptic PDE and the map from coefficient to solution when the coefficient is analytic over its domain; the authors prove that ϵ error is achievable for a DeepONet [23] of size only polylogarithmic in ϵ, leveraging the exponential convergence of spectral collocation methods for boundary value problems with analytic solutions. However, in the setting of learning homogenized constitutive laws in material science, discontinuous coefficients form a natural focus and indeed form the focus of this work. A few characteristics make operator learning a promising option in this context. First, machine learning has been groundbreaking in application settings with no clear underlying equations, such as computer vision and language models [72, 73]. In constitutive modeling, though the microscale constitutive laws are known, the homogenized equations are generally unknown and can incorporate dependencies that are not present on the microscale, such as history dependence, anisotropy, and slip-stick behavior [38, 74]. Thus, constitutive models lie in a partially equation-free setting where data-driven methods could be useful. Second, machine learned models as surrogates for expensive computation can be valuable when the cost of producing data and training the model can be amortized over many forward uses of the trained model. Since the same materials are often used for fabrication over long time periods, this can be a setting where the upfront cost of data production and model training is

justified.

Other work has already begun to explore the use of data-driven methods for constitutive modeling; a general review of the problem and its challenges, in the context of constitutive modeling of composite materials, may be found in [75]. Several works use the popular framework of physics-informed machine learning to approach the problem [76, 77, 78, 79]. In [51], physical constraints are enforced on the network architecture while learning nonlinear elastic constitutive laws. In [80], the model is given access to additional problem-specific physical knowledge. Similarly, the work of [81] predicts the Cholesky factor of the tangent stiffness matrix from which the stress may be calculated; this method enforces certain physical criteria. The paper [82] studies approximation error and uncertainty quantification for this learning problem. In [83], a derivative-free approach is taken to learning homogenized solutions where regularity of the material coefficient is assumed. The work of [29] illustrates the potential of operator learning methodology to model constitutive laws with history dependence, such as those that arise in crystal plasticity. Finally, a number of further works demonstrate empirically the potential of learning constitutive models, including [49, 84, 85, 86].

However, the underlying theory behind operator learning for constitutive models lags behind its empirical application. In [56], approximation theories are developed to justify the use of a recurrent Markovian architecture that performs well in application settings with history dependence. This architecture is further explored in [16] with more complex microstructures. Universal approximation results are a first step in developing theory for learning because they guarantee that there exists an ϵ-approximate operator within the operator approximation class, which is consistent with an assumed true model underlying the data [4, 87, 52, 20]. In addition to universal approximation, further insight may be gained by seeking to quantify the data or model size required to obtain a given level of accuracy; the papers [87, 20, 88] also contain work in this direction, as do the papers [89, 90], which build on the analysis developed in [65, 66, 67] referred to above. In our work we leverage an existing universal approximation theorem for Fourier neural operators (FNOs), a particular practically useful architecture from within the neural operator (NO) class [20]. We take two different approaches to proving approximation theorems based on separate PDE solution stability results in pursuit of a more robust under-

standing of the learning problem. Since the state of the field is in its infancy, it is valuable to have different approaches to these analysis problems. Finally, we perform numerical experiments on various microstructures to understand the practical effects of non-smooth PDE coefficients in learning solutions. We highlight the fact that in this chapter we do not tackle issues related to the non-convex optimization problem at the heart of training neural networks; we simply use state of the art stochastic gradient descent for training, noting that theory explaining its excellent empirical behaviour is lacking.

Throughout this chapter we focus on equation (3.1.1), which describes a conductivity equation in a heterogeneous medium; a natural generalization of interest is to the constitutive law of linear elasticity, in which the solution is vector-valued and the coefficient is a fourth order tensor. Though it is a linear elliptic equation, we echo the sentiment of Blanc and Le Bris [59] with their warning "do not underestimate the difficulty of equation (3.1.1)." There are many effects to be understood in this setting, and resolving learning challenges is a key step towards understanding similar questions for the learning of parametric dependence in more complex homogenized constitutive laws where machine learning may prove particularly useful.

Stability Estimates

At the heart of universal approximation theorems is stability of the solution map (3.1.6); in particular continuity of the map for certain classes of A. In this subsection, we present three key stability results that are used to prove the approximation theorems in Section 3.3. The proofs of the following stability estimates may all be found in Appendix B.1.

A first strike at the stability of the solution map (3.1.6) is a modification of the classic L^∞/H^1 Lipschitz continuity result for dependence of the solution of elliptic PDEs on the coefficient; here generalization is necessary because the coefficient also appears on the right-hand side of the equation defining $G(\cdot)$.

***Proposition* 3.1.1.** Consider the cell problem defined by equation (3.1.4). The following hold:

1. If $A \in \mathsf{PD}_{\alpha,\beta}$, then (3.1.4) has a unique solution $\chi \in \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)$ and

$$\|\chi\|_{\dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)} \leq \frac{\sqrt{d}\beta}{\alpha}.$$

2. For $\chi^{(1)}$ and $\chi^{(2)}$ solutions to the cell problem in equation (3.1.4) associated with coefficients $A^{(1)}, A^{(2)} \in \mathsf{PD}_{\alpha,\beta}$, respectively, it follows that

$$\|\chi^{(2)} - \chi^{(1)}\|_{\dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)} \leq \frac{\sqrt{d}}{\alpha}\left(1 + \frac{\beta}{\alpha}\right)\|A^{(1)} - A^{(2)}\|_{L^\infty(\mathbb{T}^d;\mathbb{R}^{d\times d})}. \quad (3.1.8)$$

◊

However, this perturbation result is insufficient for approximation theory because the space L^∞ is not separable and it is not natural to develop approximation theory in such spaces [91, Chapter 9]. While it is possible to define the problem on a separable subspace of L^∞ (see Lemma B.1.1) such spaces are not particularly useful in applications to micromechanics. Many natural models for realistic microstructures work with classes of discontinuous functions in which the boundary of material discontinuity can occur anywhere in the domain. Such functions cannot be contained in any separable subspace of L^∞; see Lemma B.1.2. To deal with this issue it is desirable to establish continuity from L^q to $\dot{H}^1$ for some $q \in [2, \infty)$. To this end, we provide two additional stability results. The first stability result gives continuity, but not Lipschitz continuity, from L^2 to $\dot{H}^1$. The second stability result gives Lipschitz continuity from L^q to $\dot{H}^1$, some $q \in (2, \infty)$.

***Proposition* 3.1.2.** Endow $\mathsf{PD}_{\alpha,\beta}$ with the $L^2(\mathbb{T}^d;\mathbb{R}^{d\times d})$ induced topology and let $K \subset \mathsf{PD}_{\alpha,\beta}$ be a closed set. Define the mapping $G : K \to \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)$ by $A \mapsto \chi$ as given by (3.1.4). Then there exists a bounded continuous mapping

$$\mathcal{G} \in C(L^2(\mathbb{T}^d;\ \mathbb{R}^{d\times d}); \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d))$$

such that $\mathcal{G}(A) = G(A)$ for any $A \in K$. ◊

The preceding L^2 continuity proposition is used to prove the approximation results for the FNO in Theorems 3.3.2 and 3.3.3. While not necessary for the approximation theory proofs, the following proposition on Lipschitz continuity from L^q to $\dot{H}^1$ establishes a more concrete bound on the approximation error, which allows for additional analysis such as providing rough bounds on grid error as discussed in Section 3.4.

***Proposition* 3.1.3.** There exists $q_0 \in (2, \infty)$ such that, for all q satisfying $q \in (q_0, \infty]$, the following holds. Endow $\mathsf{PD}_{\alpha,\beta}$ with the $L^q(\mathbb{T}^d;\mathbb{R}^{d\times d})$ topology

and let $K \subset \mathsf{PD}_{\alpha,\beta}$ be a closed set. Define the mapping $G : K \to \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)$ by $A \mapsto \chi$ as given by (3.1.4). Then there exists a bounded Lipschitz-continuous mapping

$$\mathcal{G} : \ L^q(\mathbb{T}^d;\mathbb{R}^{d\times d}) \to \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)$$

such that $\mathcal{G}(A) = G(A)$ for any $A \in K$. ◊

***Remark* 3.1.4.** Explicit upper bounds for q_0 in Proposition 3.1.3 exist and are discussed in Remark B.1.14. ◊

3.2 Microstructures

The main application area of this work is constitutive modeling. In this section we describe various classes of microstructures that our theory covers. In particular, we describe four classes of microstructures in two dimensions:

1. Smooth microstructures generated via truncated, rescaled log-normal random fields.
2. Discontinuous microstructures with smooth interfaces generated by Lipschitz star-shaped inclusions.
3. Discontinuous microstructures with square inclusions.
4. Voronoi crystal microstructures.

Visualizations of examples of these microstructures may be found in Figure 3.1. We emphasize that all four examples lead to functions in BV, a fact that we exploit in Section 3.4 when showing that our abstract analysis from Section 3.3 applies to them all.

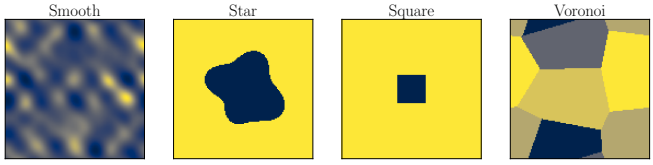


Figure 3.1: Microstructure Examples

Smooth Microstructures The smooth microstructures are generated by exponentiating a rescaled Gaussian random field. A is symmetric and coercive everywhere in the domain with a bounded eigenvalue ratio. Furthermore, the

smooth function A and its derivatives are Lipschitz. Our theory is developed specifically to analyze non-smooth microstructures, so this example is used mainly as a point of comparison.

Star Inclusions For the star inclusion microstructure, A is taken to be constant inside and outside the star-shaped boundary. The boundary function is smooth and Lipschitz in each of its derivatives. A is positive and coercive in both regions with a bounded eigenvalue ratio. This microstructure introduces discontinuities, but the boundary remains smooth.

Square Inclusions For the square inclusion microstructure, A is taken to be constant inside and outside the square boundary. Since we assume periodicity, without loss of generality the square inclusion is centered. The size of the square inclusion within the cell is varied between samples as are the constant values of A. This microstructure builds on the complexity of the star inclusion microstructure by adding corners to the inclusion boundary.

Voronoi Interfaces The Voronoi crystal microstructures are generated by assuming a random Voronoi tessellation and letting A be piecewise constant taking a single value on each Voronoi cell. The values of A on the cells and locations of the cell centers may be varied. This is the most complex microstructure among our examples and is a primary motivation for this work as Voronoi tessellations are a common model for crystal structure in materials.

3.3 Universal Approximation Results

In this section we state the two approximation theorems for learning solution operators to the cell problem. Theorem 3.3.2 concerns learning the map $A \to \chi$ in equation (3.1.4), and Theorem 3.3.3 concerns learning the map $A \to \overline{A}$ described by the combination of equations (3.1.4) and (3.1.3). Theorems 3.3.2 and 3.3.3 are specific to learning a Fourier neural operator (FNO), which is a subclass of the general neural operator. The proofs of the theorems in this section may be found in Appendix B.2.

Definitions of Neural Operators

First, we define a general neural operator (NO). The definition of the NO and the FNO are largely taken from [52], and we refer to this work for a more in-

depth understanding of these operators. In this work, we restrict the domain to the torus.

***Definition* 3.3.1** (General Neural Operator)**.** Let $\mathcal{A}$ and $\mathcal{U}$ be two Banach spaces of real vector-valued functions over domain $\mathbb{T}^d$. Assume input functions $a \in \mathcal{A}$ are $\mathbb{R}^{d_a}$-valued while the output functions $u \in \mathcal{U}$ are $\mathbb{R}^{d_u}$-valued. The neural operator architecture $\mathcal{G}_\theta : \mathcal{A} \to \mathcal{U}$ is

$$\begin{aligned} \mathcal{G}_\theta &= \mathcal{Q} \circ \mathsf{L}_{T-1} \circ \cdots \circ \mathsf{L}_0 \circ \mathcal{P}, \\ v_{t+1} &= \mathsf{L}_t v_t = \sigma_t(W_t v_t + \mathcal{K}_t v_t + b_t), \quad t = 0, 1, \ldots, T-1 \end{aligned}$$

with $v_0 = \mathcal{P}(a)$, $u = \mathcal{Q}(v_T)$, and $\mathcal{G}_\theta(a) = u$. Here, $\mathcal{P} : \mathbb{R}^{d_a} \to \mathbb{R}^{d_{v_0}}$ is a local lifting map, $\mathcal{Q} : \mathbb{R}^{d_{v_T}} \to \mathbb{R}^{d_u}$ is a local projection map and the σ_t are fixed nonlinear activation functions acting locally as maps $\mathbb{R}^{d_{v_{t+1}}} \to \mathbb{R}^{d_{v_{t+1}}}$ in each layer (with all of $\mathcal{P}$, $\mathcal{Q}$, and the σ_t viewed as operators acting pointwise, or pointwise almost everywhere, over the domain $\mathbb{T}^d$), $W_t \in \mathbb{R}^{d_{v_{t+1}} \times d_{v_t}}$ are matrices, $\mathcal{K}_t : \{v_t : \mathbb{T}^d \to \mathbb{R}^{d_{v_t}}\} \to \{v_{t+1} : \mathbb{T}^d \to \mathbb{R}^{d_{v_{t+1}}}\}$ are integral kernel operators and $b_t : \mathbb{T}^d \to \mathbb{R}^{d_{v_{t+1}}}$ are bias functions. For any $m \in \mathbb{N}_0$, the activation functions σ_t are restricted to the set of continuous $\mathbb{R} \to \mathbb{R}$ maps which make real-valued, feed-forward neural networks dense in $C^m(\mathbb{R}^d)$ on compact sets for any fixed network depth.[1] The integral kernel operators $\mathcal{K}_t$ are defined as

$$(\mathcal{K}_t v_t)(x) = \int_{\mathbb{T}^d} \kappa_t(x, y) v_t(y)\, dy$$

with standard multi-layered perceptrons (MLP) $\kappa_t : \mathbb{T}^d \times \mathbb{T}^d \to \mathbb{R}^{d_{v_{t+1}} \times d_{v_t}}$. We denote by θ the collection of parameters that specify $\mathcal{G}_\theta$, which include the weights W_t, biases b_t, parameters of the kernels κ_t, and the parameters describing the lifting and projection maps $\mathcal{P}$ and $\mathcal{Q}$ (usually also MLPs). $\Diamond$

The FNO is a subclass of the NO. Recall its definition in the introduction of this thesis in 1.3.1.

From the definition of the FNO, we note that parameterizing the kernels in the Fourier domain allows for efficient computation using the FFT. We refer to [52, 19] for additional details.

Finally we observe that in numerous applications, an example being learning of the map $A \mapsto \overline{A}$ (3.1.3), (3.1.4), it is desirable to modify the FNO so

[1] We note that all globally Lipschitz, non-polynomial, $C^m(\mathbb{R})$ functions belong to this class.

that the output space is simply a Euclidean space, and not a function space; this generalization is explored in [21]. An alternative approach, exemplified by Theorem 3.3.3 in the next subsection, is to allow the FNO output to be a function that may be evaluated at any point in the domain to yield an approximation of the point in Euclidean space.

Main Theorems

These two theorems guarantee the existence of an FNO approximating the maps $A \mapsto \chi$ and $A \mapsto \overline{A}$ and are based on the stability estimate for continuity from $L^2 \to \dot{H}^1$ obtained in Proposition 3.1.2. Both theorems are proved in Appendix B.2.

***Theorem* 3.3.2.** Let $K \subset \mathsf{PD}_{\alpha,\beta}$ and define the mapping $G : K \to \dot{H}^1(\mathbb{T}^d; \mathbb{R}^d)$ by $A \mapsto \chi$ as given by (3.1.4). Assume in addition that K is compact in $L^2(\mathbb{T}^d; \mathbb{R}^{d\times d})$. Then, for any $\epsilon > 0$, there exists an FNO $\Psi : K \to \dot{H}^1(\mathbb{T}^d; \mathbb{R}^d)$ such that

$$\sup_{A\in K} \|G(A) - \Psi(A)\|_{\dot{H}^1} < \epsilon.$$

$\Diamond$

***Theorem* 3.3.3.** Let $K \subset \mathsf{PD}_{\alpha,\beta}$ and define the mapping $F : K \to \mathbb{R}^{d\times d}$ by $A \mapsto \overline{A}$ as given by (3.1.3), (3.1.4). Assume in addition that K is compact in $L^2(\mathbb{T}^d; \mathbb{R}^{d\times d})$. Then, for any $\epsilon > 0$, there exists an FNO $\Phi : K \to L^\infty(\mathbb{T}^d; \mathbb{R}^{d\times d})$ such that

$$\sup_{A\in K} \sup_{x\in\mathbb{T}^d} |F(A) - \Phi(A)(x)|_F < \epsilon.$$

$\Diamond$

The above approximation results can also be formulated to hold, on average, over any probability measure with a finite second moment that is supported on $\mathsf{PD}_{\alpha,\beta}$. In particular, if we let μ be such a probability measure then there exists an FNO or a neural operator Ψ such that

$$\mathbb{E}_{A\sim\mu}\|G(A) - \Psi(A)\|_{\dot{H}^1} < \epsilon. \tag{3.3.1}$$

This follows by applying Theorem 18 from [20] in the respective proofs instead of Theorem 5 from the same work. We do not carry out the full details here. While this allows approximation over the non-compact set $\mathsf{PD}_{\alpha,\beta}$, the error can only be controlled on average instead of uniformly. In Section 3.4, inputs are

generated via probability measures supported on compact subsets of L^2; thus both the approximation Theorem 3.3.2, and its analog in the form (3.3.1), are relevant.

3.4 Numerical Experiments

In this section, we show that it is possible to find good operator approximations of the homogenization map (3.1.6), defined by (3.1.4), in practice. We focus on use of the FNO and note that, while Theorems 3.3.2 and 3.3.3 assert the existence of desirable operator approximations, they are not constructive and do not come equipped with error estimates. We find approximations using standard empirical loss minimization techniques and, by means of numerical experiments, quantify the complexity with respect to volume of data and with respect to size of parametric approximation.

We work with the microstructures from Section 3.2. In this context we note that Theorems 3.3.2 and 3.3.3 apply. To demonstrate this it is necessary to establish that the subsets of coefficient functions employed are compact in L^2. We achieve this by noting that all our sets of coefficient functions are contained in $\mathsf{PD}_{\alpha,\beta} \cap \mathrm{BV}$. Then we use Lemma B.3.1 to establish compactness of these subsets of coefficient functions in L^2. The smooth microstructure example serves as a comparison case for examining the impact of discontinuous coefficients on the learning accuracy. The remaining three examples present different approximation theoretic challenges including curved boundaries (star inclusions), corners (square inclusions), and junctions of several domains (Voronoi).

The experiments are all conducted using an FNO with a fixed number $T = 4$ of hidden layers. The two remaining parameters to vary are the channel width d_v and the number of Fourier modes k_{max}. For implementation details, see Appendix B.4. We make the following observations based on the numerical experiments.

1. The effective $\overline{A}$ tensors computed from the model predicted solutions exhibit relative error under 1% for all examples; the effective $\overline{A}$ is computed from the learned cell problem solution χ using equation (3.1.3).

2. The error in the learned χ is significantly higher along discontinuous material boundaries and corner interfaces, as expected. However, the

FNO operator approximation is able to approximate the solution with reasonable relative error even for the most complex case; this most complex case concerns the set of input functions with varying Voronoi geometry and varying microstructural properties within the domain.

3. In comparison with the smooth microstructure case, learning the map for the Voronoi microstructure requires substantially more data to avoid training a model which plateaus at a poor level of accuracy.

4. When compared with the smooth microstructure case, the error for the Voronoi microstructure decreases more slowly with respect to increasing model width, but shows more favourable response with respect to increasing the number of Fourier modes.

5. Models trained at one discretization may be evaluated at different discretizations for both the smooth and Voronoi microstructures as is characteristic of the FNO. The Voronoi microstructure exhibits, empirically, greater robustness to changes in discretization.

We first describe implementation details of each of the microstructures, and then we show and discuss results of numerical experiments.

Microstructure Implementation

baFor each microstructure, two positive eigenvalues and three components of the two eigenvectors are randomly generated, and the final eigenvector component is chosen to enforce symmetry. All eigenvalue ratios are at most e^2 by construction. In this manner, A is symmetric and coercive and has a bounded eigenvalue ratio.

Smooth Microstructures The smooth microstructures are generated by exponentiating a rescaled approximation of a Gaussian random field. The random field used to generate the eigenvalues and three eigenvector components of $A(x)$ is as follows:

$$\widehat{\lambda}_i(x) = \sum_{k_1,k_2=1}^{4} \xi^{(1)}_{k_1,k_2} \sin(2\pi k_1 x_1)\cos(2\pi k_2 x_2) + \xi^{(2)}_{k_1,k_2} \cos(2\pi k_1 x_1)\sin(2\pi k_2 x_2),$$

$$\lambda_i(x) = \exp\left(\frac{\widehat{\lambda}_i(x)}{\max_{x'\in[0,1]^2} |\widehat{\lambda}_i(x')|}\right),$$

where $\xi_{k_1,k_2}^{(j)}$ are i.i.d. normal Gaussian random variables.

Star-Shaped Inclusions The star-shaped inclusions are generated by defining a random Lipschitz polar boundary function as

$$r(\theta) = \mathsf{a} + \mathsf{b} \sum_{k=1}^{5} \xi_k \sin(k\theta),$$

where ξ_k are i.i.d. uniform random variables $U[-1, 1]$, and a and b are constants that guarantee $0 < \epsilon < r < 0.5 - \epsilon$ for some fixed $\epsilon > 0$. Then $A(x)$ is constant inside and outside the boundary. We randomly sample eigenvalues for A on each domain via $\lambda_i \sim U[e^{-1}, e]$. The three components of the eigenvectors are i.i.d. normal random variables.

Square Inclusions The radius of the square is randomly generated via

$$r = \mathsf{a} + \mathsf{b}\zeta,$$

where ζ is a uniform random variable on $[0, 1]$ and a and b are positive constants that guarantee $0 < \epsilon < r < 0.5 - \epsilon$ for some fixed $\epsilon > 0$. The values of A on each of the constant domains are chosen in the same manner as in the star-shaped inclusion case.

Voronoi Interfaces The Voronoi crystal microstructure has constant A on each Voronoi cell and is chosen uniformly at random in the same manner as for the star inclusions. Voronoi tessellations are a common model for crystal structure in materials. In one Voronoi example, we fix the geometry for all data, and in a second Voronoi example we vary the geometry by randomly sampling five cell centers from a uniform distribution on the unit square.

Results

Each FNO model is trained using the empirical estimate of the mean squared H^1 norm:

$$\text{Loss}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \Big(\|\chi^{(n)} - \widehat{\chi}^{(n)}\|_{L^2}^2 + \|\nabla\chi^{(n)} - \nabla\widehat{\chi}^{(n)}\|_{L^2}^2 \Big), \tag{3.4.1}$$

where n is the sample index, χ is the true solution, and $\widehat{\chi}$ is the FNO approximation of the solution parameterized by θ. In the analysis, we examine several

different measures of error, including the following relative H^1 and relative $W^{1,10}$ errors:

$$\text{Relative } H^1 \text{ Error (RHE)} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{\|\chi^{(n)} - \widehat{\chi}^{(n)}\|_{L^2}^2 + \|\nabla\chi^{(n)} - \nabla\widehat{\chi}^{(n)}\|_{L^2}^2}{\|\chi^{(n)}\|_{L^2}^2 + \|\nabla\chi^{(n)}\|_{L^2}^2}\right)^{\frac{1}{2}} \tag{3.4.2a}$$

$$\text{Relative } W^{1,10} \text{ Error (RWE)} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{\|\chi^{(n)} - \widehat{\chi}^{(n)}\|_{L^{10}}^{10} + \|\nabla\chi^{(n)} - \nabla\widehat{\chi}^{(n)}\|_{L^{10}}^{10}}{\|\chi^{(n)}\|_{L^{10}}^{10} + \|\nabla\chi^{(n)}\|_{L^{10}}^{10}}\right)^{\frac{1}{10}}. \tag{3.4.2b}$$

The $W^{1,10}$ norm gives a sense of the higher errors that occur at interfaces, corners, and functions. We could have used $W^{1,p}$ for any p large enough.

Finally, we also look at error in $\overline{A}$, which we scale by the difference between the arithmetic and harmonic mean of A. Any effective $\overline{A}$ should have a norm in this range; these are known in mechanics as Voigt-Reuss bounds and have a physical interpretation as bounds obtained via energy principles by ignoring equilibrium for the upper bound (arithmetic mean) and ignoring compatibility for the lower bound (harmonic mean) [92]. The resulting error measure is given by

$$\text{Relative } \overline{A} \text{ Error (RAE)} = \frac{\|\overline{A} - \widehat{\overline{A}}\|_F}{a_m - a_h}, \tag{3.4.3}$$

where the arithmetic mean a_m and harmonic mean a_h are given by

$$a_m = \left\| \int_{\mathbb{T}^2} A(x)\, \mathsf{d}x \right\|_F$$
$$a_h = \left\| \left(\int_{\mathbb{T}^2} A^{-1}(x)\, \mathsf{d}x \right)^{-1} \right\|_F.$$

We note that using $a_m - a_h$ rather than $\|\overline{A}\|_F$ as a scaling factor in equation (3.4.3) leads to a larger error value, so achieving low error in this measure of distance is harder.

We train models on five different datasets. Visualizations of the median-error test samples for each example may be viewed in Figure 3.2, and the numerical errors are shown in Figure 3.3. Each of these models is trained on 9500 data samples generated using an FE solver on a triangular mesh with the solution interpolated to a 128×128 grid. Additional model details may be found in Appendix B.4.

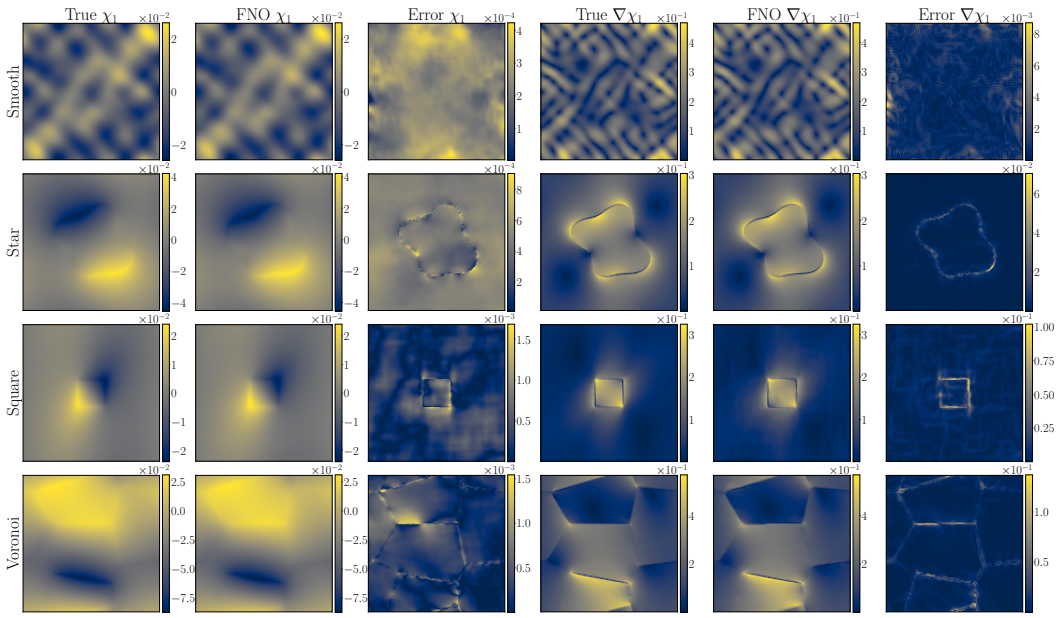


Figure 3.2: Visualization of the trained models evaluated on test samples that gave median relative H^1 error for each microstructure. The microstructure inputs of each row correspond to those of Figure 3.1. The first shows the true χ_1, the second shows the FNO predicted χ_1, and the third shows the absolute value of the error between the true and predicted χ_1. The fourth column shows the 2-norm of the gradient of the true χ_1, and the fifth shows the 2-norm of the gradient of the predicted χ_1. The last column shows the 2-norm of the difference between the two gradients.

We perform an experiment to test the discretization robustness of the FNO model, results of which are shown in Figure 3.4. The models are trained with data from the resolution 128×128 and evaluated on test data with different resolution. We emphasize that evaluating the FNO on different resolutions is trivial in implementation by design.

We also investigate the effects of the number of training data and the model size on the error for the smooth and Voronoi microstructures; similar experiments, for different operator learning problems, are presented in [93]. A plot of error versus training data may be found in Figure 3.5, and plots of error versus the number of Fourier modes for fixed total model size, as measured by (model width) $\times$ (number Fourier modes), may be found in Figure 3.6. Figure 3.6 addresses the question of how to optimally distribute computational budget through different parameterizations to achieve minimum error at given cost as measured by number of parameters; it should be compared to similar experiments in [94].

Microstructure	Mean RHE	Mean RWE	Median RAE
Smooth	$0.0062 \pm 1 \cdot 10^{-4}$	$0.0091 \pm 1 \cdot 10^{-4}$	$0.0007 \pm 1 \cdot 10^{-5}$
Star	$0.0313 \pm 1 \cdot 10^{-4}$	$0.1318 \pm 5 \cdot 10^{-4}$	$0.0014 \pm 3 \cdot 10^{-5}$
Square	$0.1012 \pm 5 \cdot 10^{-4}$	$0.2741 \pm 2 \cdot 10^{-3}$	$0.0047 \pm 1 \cdot 10^{-4}$
Voronoi	$0.0565 \pm 4 \cdot 10^{-4}$	$0.2129 \pm 3 \cdot 10^{-3}$	$0.0027 \pm 8 \cdot 10^{-5}$
Voronoi (Fixed Geometry)	$0.0073 \pm 3 \cdot 10^{-5}$	$0.0140 \pm 3 \cdot 10^{-4}$	$0.0007 \pm 2 \cdot 10^{-5}$

Figure 3.3: Errors for each each numerical experiment; five sample models are trained for each microstructure. The expressions for the RHE (Relative H^1 Error), RWE (Relative $W^{1,10}$ Error) and RAE (Relative $\overline{A}$ Error) may be found in equations (3.4.2) and (3.4.3). The errors are evaluated over a test set of size 500. All examples have varying geometry except the second Voronoi example.

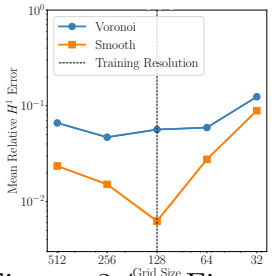


Figure 3.4: Five sample models trained on Smooth and Voronoi data at 128×128 grid resolution evaluated at different resolutions.

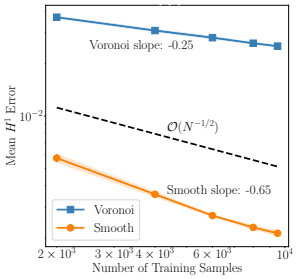


Figure 3.5: A comparison of test error for different amounts of training data for models trained on Voronoi and Smooth data. Five sample models are used for each data point.

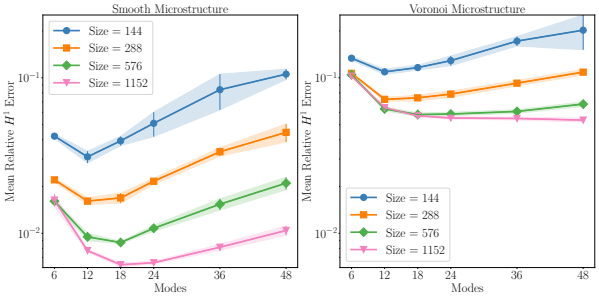


Figure 3.6: Relative H^1 error versus model size for the smooth and Voronoi examples with varying geometry. The number of Fourier modes in each direction and the model width were varied. Each line indicates a constant product of modes×width. Training data size was fixed at 9500 samples, and five samples were used for each data point.

Discussion

As can be seen from the data in Figure 3.3, the microstructures exhibiting discontinuities lead to higher model error than the smooth microstructure, and the introduction of corner interfaces leads to further increase in error. The visualizations of the median-error test samples in Figure 3.2 give some intuition; error is an order of magnitude higher along discontinuous boundaries; this is most apparent in the gradient. The true solution gradient often takes its most extreme values along the discontinuities, and the RWE gives an indication of how well the model captures the most extreme values in the solution. Unsurprisingly, this error is much higher than the RHE, but we note that it

is confined to a small area of the domain along discontinuous boundaries and corner interfaces.

In the discretization-robustness experiment described in Figure 3.4, we observe that the Voronoi model exhibits greater robustness to changes in discretization. We hypothesize that, in the direction of decreasing resolution, the smaller error increase for the Voronoi model, in comparison with the smooth model, could be due to the piecewise-constant nature of the Voronoi microstructure on faces; improved resolution here does not help. On the other hand, for larger grid sizes, increased resolution on corners and discontinuities can help, which could explain the decrease in error from grid edge size of 128 to 256 for the Voronoi model while the smooth model increases in error. One could fine-tune the trained models with small amounts of data from different resolutions, but we leave this transfer learning exploration to future work.

We also examine the effect of the number of training data samples and the FNO size on model accuracy for the smooth and Voronoi microstructures. For data size dependence, we observe in Figure 3.5 that for these two microstructures, the test error scales $\approx N^{-0.65}$ and $\approx N^{-0.25}$, respectively, where N is the number of training data. In theory, we do not expect to beat the Monte Carlo error decay of $\frac{1}{\sqrt{N}}$ [95]. We note that this is comparable to the behavior during training over 400 epochs; the test error for the smooth microstructure continues to decrease over the entire training periodic, but the test error for the Voronoi microstructure plateaus by around 100 epochs. The model size also presents a qualitatively different effect on error for the smooth and Voronoi microstructures. In Figure 3.6, we see the tradeoff between the number of Fourier modes and the model width for approximately constant model size, measured as the product of the width and number of modes. The Voronoi example benefits from additional Fourier modes, whereas the smooth example worsens. On the other hand, the smooth model benefits more from an increase in model width. We refer to [93, 94] for in-depth numerical studies of errors, choice of hyperparameters, and parameter distributions for FNO; here we highlight only the qualitative differences between the model behavior for different microstructures.

We also note that a significant portion of the model error may be attributed to grid ambiguity; with a 128×128 grid, the FNO does not know where between gridpoints a discontinuity may fall. This may be quantified empirically in the

case of the square microstructure. We perform an experiment in which we create data of square microstructure inclusions whose boundary falls exactly on the gridpoints. One dataset treats the boundary as open, and the other treats the boundary as closed; the input grid points that fall on the boundary differ between the two datasets. We quantify grid ambiguity error by the difference in the outputs of a model given both the open square data and the closed square data. We find that the absolute H^1 norm of the difference between these two outputs is 0.041, which is slightly under twice the absolute H^1 norm of the output compared to the true solution, which has a value of 0.025. We hypothesize that the model learns to assume the boundary falls near the middle of the grid square, which explains why the output difference between the two datasets is roughly twice the true error. From a theory standpoint, one could bound the Lipschitz constant of the FNO and compare it to the Lipschitz constant of the true map described by Proposition 3.1.3. However, we leave the theoretical estimates of error rates to future work.

Finally, we compare the error in the effective $\overline{A}$ defined in (3.1.3). This error is scaled by a difference between the Frobenius norms of the arithmetic and harmonic means of the true A because the Frobenius norm of the true $\overline{A}$ should fall within that range. For this reason, in the case where the arithmetic and harmonic means are very close, as is frequently the case for the square and star inclusions, it is not valuable to learn the true $\overline{A}$. On the other hand, the varying-geometry Voronoi microstructure example on average has about 100 times greater difference between the arithmetic and harmonic means, in comparison with the star and square microstructure examples. This characteristic of the Voronoi microstructure further underscores the value of learning in this setting.

3.5 Conclusions

In this work, we establish approximation theory for learning the solution operator arising from the elliptic homogenization cell problem (3.1.4), viewed as a mapping from the coefficient to the solution; the theory allows for discontinuous coefficients. We also perform numerical experiments that validate the theory, explore qualitative differences between various microstructures, and quantify error/cost trade-offs in the approximation. We provide two different stability results for the underlying solutions that build understanding of the underlying map. These stability results, when combined with existing

universal approximation results for neural operators, result in rigorous approximation theory for learning in this problem setting. On the empirical side we provide, and then study numerically, examples of various microstructures that satisfy the conditions of the approximation theory. We observe that model error is dominated by error along discontinuous and corner interfaces, and that discontinuous microstructures give rise to qualitatively different learning behavior. Finally, we remark that the learned effective properties are highly accurate, especially in the case of the Voronoi microstructure that we regard as the most complex. Since discontinuous microstructures arise naturally in solid mechanics, understanding learning behavior in this context is an important prerequisite for using machine learning for applications. In this area and others, numerous questions remain which address the rigor necessary for use of machine learning in scientific applications.

We have confined our studies to one of the canonical model problems of homogenization theory, the divergence form elliptic setting with periodic microstructure, to obtain deeper understanding of the learning constitutive laws. One interesting potential extension of this work is the setting in which the material coefficient A is not periodic but random with respect to the microstructure. Another is where it is only locally periodic and has dependence on the macroscale variable as well; thus $A^\epsilon = A(x, \frac{x}{\epsilon})$. In this case, the form of the cell problem (3.1.4) and homogenized coefficient (3.1.3) remain the same, but A and χ both have parametric dependence on x. The approximation theory and the empirical learning problem would grow in complexity in comparison to what is developed here, but the resulting methodology could be useful and foundational for understanding more complex constitutive models in which the force balance equation couples to other variables. Indeed, the need for efficient learning of constitutive models is particularly pressing in complex settings such as crystal plasticity. We anticipate that the potential use of machine learning to determine parametric dependence of constitutive models defined by homogenization will be for these more complex problems. The work described in this chapter provides an underpinning conceptual approach, foundational analysis, and set of numerical experiments that serve to underpin more applied work in this field.

Chapter 4

DISCRETIZATION ERROR OF FOURIER NEURAL OPERATORS

This chapter is adapted from the following preprint:

[1] Samuel Lanthaler, Andrew M. Stuart, and Margaret Trautner. *Discretization error of Fourier neural operators*. 2024. arXiv: 2405.02221 [math.NA].

Operator learning is a variant of machine learning that is designed to approximate maps between function spaces from data. The Fourier Neural Operator (FNO) is one of the main model architectures used for operator learning. The FNO combines linear and nonlinear operations in physical space with linear operations in Fourier space, leading to a parameterized map acting between function spaces. Although in definition, FNOs are objects in continuous space and perform convolutions on a continuum, their implementation is a discretized object performing computations on a grid, allowing efficient implementation via the FFT. Thus, there is a discretization error between the continuum FNO definition and the discretized object used in practice that is separate from other previously analyzed sources of model error. We examine this discretization error here and obtain algebraic rates of convergence in terms of the grid resolution as a function of the input regularity. Numerical experiments that validate the theory and describe model stability are performed. In addition, an algorithm is presented that leverages the discretization error and model error decomposition to optimize computational training time.

4.1 Introduction

Overview

While most learning architectures are designed to approximate maps between finite-dimensional spaces, operator learning is a method that approximates maps between infinite-dimensional function spaces. These maps appear commonly in scientific machine learning applications such as surrogate modeling of partial differential equations (PDEs) or model discovery from data. Fourier Neural Operators (FNOs) [19] are a type of operator learning architecture

that parameterize the model directly in function space, naturally generalizing deep neural networks (DNNs). In particular, each hidden layer of an FNO assigns a trainable integral kernel that acts on the hidden states by convolution in addition to the usual affine weights and biases of a DNN. Taking advantage of the duality between convolution and multiplication under Fourier transforms, these convolutional kernels are represented by Fourier multiplier matrices, whose components are optimized during training alongside the regular weights and biases acting in physical space. FNOs have proven to be an effective and popular operator learning method in several PDE application areas including weather forecasting [96], biomedical shape optimization [97], and constitutive modeling [56]. It is thus of interest to study their theoretical properties.

Although FNOs approximate maps between function spaces, in practice, these functions must be discretized. In particular, kernel integral operators, including the FNO, perform convolution via an integration that must be computed numerically. The error arising from this difference is called *aliasing error*, and during a forward pass of the FNO, the aliasing error propagates through the subsequent model layers and may be amplified by nonlinearities. Thus, the continuum FNO object differs from the implemented model due to *discretization error*. This may be summarized by the following decomposition:

$$\Psi^\dagger - \Psi_{FNO}^N = \underbrace{\left[\Psi^\dagger - \Psi_{FNO}\right]}_{\text{model discrepancy}} + \underbrace{\left[\Psi_{FNO} - \Psi_{FNO}^N\right]}_{\text{discretization error}}. \tag{4.1.1}$$

Here, $\Psi^\dagger$ is the true map to be approximated by a data-driven model, Ψ_{FNO} is the continuum FNO map, and Ψ_{FNO}^N is the discretized version of the FNO. In previous analyses of the universal approximation properties of the FNO [20, 52], the discretization error component is ignored completely; only the continuum definition of the FNO is used. While this approach to universal approximation is mathematically sound, it leaves the discretization components of the error unquantified in practice. Understanding and controlling this discretization error is as important for this model as bounding the model discrepancy error arising from sources such as limited data, optimization, and model capacity. In this chapter, we analyze the discretization error both in theory and experimentally.

Aliasing error depends on the regularity, or smoothness, of the input function in the Sobolev sense; this is well known in Fourier analysis. Thus, to bound the

error for an entire FNO implementation, regularity must be maintained as the state passes through the layers of the network, including the nonlinear activation function. In particular, regularity-preserving properties of compositions of nonlinear functions are required. Bounds of this type are given by Moser [98] and form a key component of the proofs in this work. Because the smooth GeLU (Gaussian Error Linear Unit) [99] activation preserve regularity, while the non-differentiable ReLU activations do not, the analysis in this chapter is confined to the former and extends to other smooth activation functions.

Contributions

In this chapter, we make the following contributions.

(C1) We bound the discretization error that results from implementing the continuum FNO on a grid.

(C2) We validate this theory concerning the discretization error of the FNO with numerical experiments.

(C3) We propose an adaptive subsampling algorithm for faster operator learning training.

In Section 4.2 we set up the framework for our theoretical results. Section 4.3 studies the discretization error of the FNO in theory, making contribution (C1). In Section 4.4 we present numerical experiments that illustrate the theory and propose an algorithm for adaptively refining the discretization during training, making contributions (C2, C3). We conclude in Section 4.5. The appendices include a self-contained background on aliasing error as well as additional proofs and technical details.

Related Work

Neural networks have been very successful in approximating solutions of partial differential equations using data. Several approaches are used for such models, including physics-informed neural networks (PINNs), constructive networks, and operator learning models. In the case of PINNs, a standard feed-forward machine learning architecture is trained with a loss function involving a constraint of satisfying the underlying PDE [9]. Another approach to applying machine learning to PDEs is to construct approximating networks from classical PDE-solver methods. For example, in [100, 101, 102, 103], ReLU neural

networks are shown to replicate polynomial approximations and continuous, piecewise-linear elements used in finite element methods exactly. Both of these two approaches to approximating PDE solution maps require a choice of discretization within the model to approximate an infinite-dimensional operator.

Operator learning is a branch of machine learning that aims to approximate maps between function spaces, which include solution maps defined by partial differential equations (PDEs) [52]. Several operator learning architectures exist, including DeepONet [23], Fourier Neural Operators (FNO) [19], PCA-Net [68], and random features models [69]. Our work focuses on FNOs, which directly parameterize the model in Fourier space through an integral kernel and allow for changes in discretization in both the input and the output functions, potentially allowing for non-uniform grids [104]. In addition, FNO takes advantage of the computational speedup of the FFT to gain additional model capacity with less evaluation time. A key advantage of the FNO is that it is a *discretization-invariant* operator in the sense that its definition involves no discretization and its implementation can be trivially used on various discretizations with no change of parameter values.

Error analysis for operator learning begins with establishing universal approximation: results which guarantee that, for a class of possible maps, a particular choice of model architecture, and a desired maximum error, there exists a parameterization of the model that gives at most that error. Universal approximation results are established for a variety of architectures including ReLU neural networks in [4], DeepONet in [87], FNO in [20], and a general class of neural operators in [52]. Following universal approximation, model size bounds give a worst-case bound on the model parameter sizes required to achieve a certain error threshold for particular classes of problems. These have been established for FNO [20, 105], but the analysis uses only the continuum definition of the FNO and ignores the fact that in practice the FNO implementation must work with a discretized version. In this work, we close the error gap by quantifying and bounding the error that results from discretizing the continuum FNO.

Perhaps the most conceptually similar work to ours is [106], which addresses the fact that discretizations of neural operators deviate from their continuum counterparts. The authors of [106] introduce an "alias-free" neural operator that bypasses inconsistencies resulting from discretization. In practice, this

research direction has led to operator learning frameworks such as Convolutional Neural Operators (CNO) [107], which are not strictly alias-free, but reduce aliasing errors via spatial upsampling. These prior works have empirically shown the benefits and importance of carefully controlling discretization errors in operator learning. Prior work has also examined the effects of changing the number of spectral modes in the implemented FNO and an algorithm to optimize training with variable modes [108]. In this work, we also propose an adaptive subsampling algorithm that varies the resolution of the data used in training in a manner designed to minimize training time.

FNOs remain a widespread neural operator architecture, and an analysis of errors resulting from numerical discretization have so far been missing from the literature. To fill this gap, in this work we bound the discretization error of FNOs and perform experiments that provide greater insight into the behavior of this error.

4.2 Notation

Fix integer m. Let $|\cdot|$ denote the Euclidean norm on $\mathbb{R}^m$, $\|\cdot\|$ the $L^2(\mathbb{T}^d, \mathbb{R}^m)$ norm, and $\|\cdot\|_\infty$ the $L^\infty(\mathbb{T}^d) = L^\infty(\mathbb{T}^d, \mathbb{R}^m)$ norm. Here, $\mathbb{T}^d$ denotes the d-dimensional torus, which we identify with $[0,1]^d$ with periodic boundary conditions; we simply write $L^2(\mathbb{T}^d)$ when no confusion will arise. Let $\|\cdot\|_2$ be the induced matrix 2-norm and $\|\cdot\|_F$ be the Frobenius norm. For a shallow neural network $\phi(u) = A_2\sigma(A_1u + b)$ with matrices A_1 and A_2 and vector b, we denote by $\|\phi\|_{\text{NN}}^2 \coloneqq \|A_1\|_F^2 + \|A_2\|_F^2 + \|b\|_F^2$. For nonnegative integer s, define the Sobolev space $H^s(\mathbb{T}^d) = H^s(\mathbb{T}^d, \mathbb{R}^m)$ as

$$H^s(\mathbb{T}^d) = \left\{ f : \mathbb{T}^d \to \mathbb{R}^m \;\middle|\; \sum_{k\in\mathbb{Z}^d} (1 + |k|^{2s})|\widehat{f}(k)|^2 < \infty \right\}, \tag{4.2.1}$$

where $\widehat{f}$ denotes the Fourier transform of f. Define the semi-norm

$$|v|_s^2 := \int_{\mathbb{T}^d} v(-\Delta)^s v \; \mathsf{d}x$$

for functions $v : \mathbb{T}^d \to \mathbb{R}^m$. It is useful to consider the following equivalent definition of the space $H^s(\mathbb{T}^d)$ for integer $s > d/2$ in terms of this seminorm:

$$\begin{aligned} H^s(\mathbb{T}^d) &= \{f : \mathbb{T}^d \to \mathbb{R}^m \mid \|f\|_{H^s} < \infty\} \\ \|f\|_{H^s} &= \left((2\pi)^{-2s}|f|_s^2 + \|f\|^2\right)^{1/2}. \end{aligned}$$

We say an element $f \in H^{s-}$ if $f \in H^{s-\epsilon}$ for any $\epsilon > 0$. Further, let $X^{(N)}$ denote the d-dimensional grid $\frac{1}{N}[N]^d$ where

$$[N]^d := \{n \in \mathbb{Z}^d_{\geq 0} \mid n_i < N,\ i \in \{1, \dots, d\}\}.$$

Here, n_i is the ith entry of vector n. We assume $N > 1$ throughout this work. We also introduce the following symmetric index set for the Fourier coefficients: $[[N]]^d = [[N]] \times \cdots \times [[N]]$, where

$$[[N]] := \begin{cases} \{-K, \dots, K\}, & (N = 2K + 1 \text{ is odd}), \\ \{-K, \dots, K-1\}, & (N = 2K \text{ is even}). \end{cases}$$

Irrespective of whether N is odd or even, $[[N]]^d$ contains N^d elements. For functions $u : \mathbb{T}^d \to \mathbb{R}^m$, we abuse notation slightly and use $\|u\|_{\ell^2(n \in [N]^d)}$ to indicate the quantity,

$$\|u\|_{\ell^2(n \in [N]^d)} := \Big(\sum_{n \in [N]^d} |u(x_n)|^2 \Big)^{1/2}.$$

This is a norm for the vector found by evaluating u at grid points. Note that for $x_n = \frac{1}{N} n$ where $n \in [N]^d$, it holds that $x_n \in \mathbb{T}^d$, and if $u \in L^2(\mathbb{T}^d)$ is Riemann integrable,

$$\lim_{N \to \infty} \frac{1}{N^{d/2}} \|u\|_{\ell^2(n \in [N]^d)} = \|u\|_{L^2(\mathbb{T}^d)}. \tag{4.2.2}$$

Finally, we define the FNO. We remark that this constitutes the standard definition of the FNO with the exception that we ask for smooth activation functions. At a high level, the FNO is a composition of layers, where the first and final layers are lifting and projection maps, and the internal layers are an activation function acting on the sum of an affine term, a nonlocal integral term, and a bias term. We emphasize that this FNO definition *does not involve a discretization*; it is a map between function spaces on a continuum. Recall the definition of the FNO in the introduction of this thesis in 1.3.1.

In the error analysis in the following section, we are interested in the discrepancy between taking the inner product in equation (1.3.1) on a grid instead of on a continuum — the errors due to *aliasing*. The above continuum definition is assumed in learning theory analysis of the FNO, but in practice, a discretized approximation is used. We consider the other parameters, including the mode count K, to be fixed and intrinsic to the FNO model considered, irrespective of which grid it is approximated on.

4.3 Main Results

Let $\mathcal{A}$ and $\mathcal{U}$ be the input and output Banach spaces in the FNO definition 1.3.1, and let the FNO model hyperparameters be fixed. Given a setting of the trainable FNO parameters θ, let $\Psi_{\text{FNO}} : \mathcal{A} \mapsto \mathcal{U}$ be the FNO obtained using the definition. *This definition does not involve a discretization.* Thus, any implementation of the FNO with the same hyperparameters and trainable θ must be some other map, denoted $\Psi^N_{\text{FNO}} : \mathcal{A} \mapsto \mathcal{U}$ that evaluates the L^2 inner product in equation (1.3.1) numerically on some grid points $X^{(N)}$ rather than at every point $x \in \mathbb{T}^d$ as Ψ_{FNO} does. In particular, Ψ^N_{FNO} exchanges the operator $\mathcal{K}_t$ defined in (1.3.1) for $\mathcal{K}^N_t$ such that

$$(\mathcal{K}^N_t v^N_t)(x_n) = \sum_{k \in [[K]]^d} \sum_{j=1}^{d_t} (P_t^{(k)})_j \mathsf{DFT}(v^N_t)(k) \mathrm{e}^{2\pi i \langle k, x_n \rangle} \in \mathbb{R}^{d_{t+1}}, \tag{4.3.1}$$

where DFT is the discrete Fourier transform; see Appendix C.1 for background. We refer to the output of each internal layer L as a *state value*. Starting from the same input $a \in \mathcal{A}$, $\Psi_{\text{FNO}}(a)$ and $\Psi^N_{\text{FNO}}(a)$ will have different state values, denoted v_t and v^N_t, respectively, as outputs of internal layer L_{t-1} for $t > 0$ despite having the exact same model parameters. This difference is important because in proofs concerning the FNO, only Ψ_{FNO} is considered, but Ψ_{FNO} is an *unimplementable* object in practice. If $\Psi^\dagger$ is the map of interest to be approximated using an FNO model, the overall approximation error of the implemented Ψ^N_{FNO} can be split into a contribution due to the numerical discretization and another contribution due to model discrepancy, as shown in (4.1.1). Theorem 4.3.2 bounds the discretization error component. The result takes into account both the initial errors that occur with each approximated inner product and their magnified effects as they propagate through the layers of the model. Despite this nonlinear propagation, we show that the approximate L^2 norm of the error after any number of layers decreases like N^{-s}, where s describes the Sobolev regularity of the input.

To prove Theorem 4.3.2, we assume the following.

***Assumption* 4.3.1.** For a fixed FNO with T layers:

(A1) There exists some $B \geq 1$ such that σ_t, σ_p, and σ_q possess continuous derivatives up to order s which are bounded by B.

(A2) Input set $\mathcal{A} \subset H^s(\mathbb{T}^d)$.

(A3) $1 \leq K < \frac{N}{2}$.

(A4) $s > \frac{d}{2}$.

(A5) There exists some $M \geq 1$ such that FNO parameters P_t, W_t, and b_t are each bounded above by M in the following norms: $\|P_t\|_F \leq M$, $\|W_t\|_2 \leq M$, and $|b_t| \leq M$ for all $t \in [0, \ldots, T-1]$. Furthermore, $\mathcal{P}$ and $\mathcal{Q}$ are bounded and smooth with $\|\mathcal{P}\|_{\text{NN}} \leq M$, and $\|\mathcal{Q}\|_{\text{NN}} \leq M$.

$\Diamond$

The main result is the following theorem concerning the behavior of the error with respect to the size of the discretization. To interpret the theorem statement in terms of norm-scaling on the left-hand side, recall (4.2.2).

***Theorem* 4.3.2.** Let Assumptions 4.3.1 hold. Let $\mathcal{A}_c$ be a compact set in $\mathcal{A}$. Let $v_t(a) := \mathsf{L}_t \circ \mathsf{L}_{t-1} \cdots \circ \mathsf{L}_0 \circ \mathcal{P}(a)$ with $\mathcal{P}$ and each L as defined in Definition 1.3.1. Similarly, let $v_t^N(a) := \mathsf{L}_t^N \circ \mathsf{L}_{t-1}^N \cdots \circ \mathsf{L}_0^N \circ \mathcal{P}(a)$ where $\mathsf{L}_j^N v_j^N = \sigma_j(W_j v_j^N + \mathcal{K}_j^N v_j + b_j)$ for $\mathcal{K}_j^N$ defined in (4.3.1) for each $0 \leq j \leq t$. Then

$$\sup_{a \in \mathcal{A}_c} \frac{1}{N^{d/2}} \|v_t(a) - v_t^N(a)\|_{\ell^2(n \in [N]^d)} \leq C N^{-s}, \tag{4.3.2}$$

where the constant C depends on B, M, d, s, t, and $\mathcal{A}_c$. $\Diamond$

The proof and exact form of the constant C in the above theorem are detailed in Appendix C.5.

We can also state the following variant of Theorem 4.3.2, which shows that the same convergence rate is obtained at the continuous level, when $v_t^N(x_n)$ is replaced by a trigonometric polynomial interpolant:

***Theorem* 4.3.3.** Let $p_t^N(x) = \sum_{k \in [[N]]^d} \mathsf{DFT}(v_t^N)(k) e^{2\pi i \langle k, x \rangle}$ denote the interpolating trigonometric polynomial of $\{v_t^N(x_n)\}_{n \in [N]^d}$. Let the assumptions of Theorem 4.3.2 hold. Then,

$$\sup_{a \in \mathcal{A}_c} \|v_t(a) - p_t^N(a)\|_{L^2(\mathbb{T}^d)} \leq C' N^{-s}. \tag{4.3.3}$$

Here, C' depends on B, M, d, s, t, and $\mathcal{A}$. $\Diamond$

***Remark* 4.3.4.** A consequence of Theorem 4.3.3 is that $\sup_{a \in \mathcal{A}_c} \|\Psi_{\text{FNO}}(a) - \Psi_{\text{FNO}}^N(a)\|_{L^2(\mathbb{T}^d)}$ also has a convergence rate of N^{-s} since $\mathcal{Q}$ is Lipschitz under Assumptions 4.3.1. $\Diamond$

The exact form of the constant C' may be found in the proof in Appendix C.6. A key element in the proof of Theorem 4.3.2 is to provide a bound on the Sobolev norm of the ground truth state $\|v_t\|_{H^s}$ at each layer. The following lemma accomplishes this for a single layer. The proof may be found in Appendix C.4. Under Assumptions 4.3.1, the following bounds hold:

- $\|v_{t+1}\|_\infty \leq \sigma_0 + BM(1 + \|v_t\|_\infty + K^{d/2}\|v_t\|_{L^2(\mathbb{T}^d)})$
- $|v_{t+1}|_s \leq BcM^s K^{ds/2}(1 + \|v_t\|_\infty)^s(1 + |v_t|_s)$

for some constant c dependent on d and s, where $\sigma_0 \coloneqq \max\{\max_{0\leq t\leq T} \sigma_t(0), 1\}$.

The result of Theorem 4.3.2 guarantees that the discretization error converges as grid resolution increases. The algebraic decay rate in a discrete L^2 norm is determined by the regularity of the input; this in turn builds on Lemma 4.3 which ensures that the regularity of the state is preserved through each layer of the FNO.

4.4 Numerical Experiments

In this section we present and discuss results from numerical experiments that empirically validate the results of Theorem 4.3.2. The L^2 error at each layer decreases like N^{-s} where s governs the input regularity and N is the discretization used to perform convolutions in the FNO implementation. For each FNO model in this section, we use a computation of a discrete FNO on a high resolution grid as the "ground truth;" this is standard practice in numerical analysis when the true solution is unobtainable. We compare states at each layer resulting from inputs of lower resolution with the state resulting from the ground truth. To obtain evaluations of v_ℓ at higher discretizations than N, the inverse Fourier transform operation is interpolated to additional grid points using trigonometric polynomial interpolation; Theorem 4.3.3 justifies this practice.

We perform experiments for inputs of varying regularity by generating Gaussian random field (GRF) inputs with prescribed smoothness H^{s-} for $s \in \{0.5, 1, 1.5, 2\}$. The GRF inputs are discretized for values of
$N \in \{32, 64, 128, 256, 512, 1024, 2048\}$ and $d = 2$. Grid size 2048 is used as the ground truth, and the relative error at layer ℓ for v_ℓ compared with the truth

$v_\ell^\dagger$ is computed with

$$\text{Relative Error} = \frac{\|v_\ell^\dagger - v_\ell\|_{\ell^2(n\in[2048]^d)}}{\|v_\ell^\dagger\|_{\ell^2(n\in[2048]^d)}}.$$

Finally, in FNO training, it is common practice to append positional information about the domain at each evaluation point in the form of Euclidean grid points; i.e. $(x_1, x_2) \in [0,1]^2$ for two dimensions. However, this grid information is not periodic, and an alternative is to append periodic grid information; i.e.$(\sin(x_1), \cos(x_1), \sin(x_2), \cos(x_2))$ for two dimensions. In these experiments, we also compare the error of models with these two different positional encodings.

We first discuss experiments on FNOs with random weights, and then discuss experiments on trained FNOs. In the random weights experiments, we present a few interesting experimental findings, namely, using ReLU activations or non-periodic position encodings negatively affects the discretization error decay as the theory predicts. In the trained network experiments, we explore the example of learning a gradient map to show that the model cannot learn a map with less regularity than the model allows. Finally, we propose an application of discretization subsampling to speed up operator learning training by leveraging adaptive grid sizes.

Experiments with random weights

In this subsection, we consider FNOs with random weights and study their discretization error and model stability with respect to perturbations of the inputs. All models are defined in spatial dimension $d = 2$, with $K = 12$ modes in each dimension, a width of 64, and 5 layers.

The default model has randomly initialized iid $\mathcal{U}(-\frac{1}{\sqrt{d_t}}, \frac{1}{\sqrt{d_t}})$ weights (uniformly distributed) for the affine and bias terms, where d_t is the layer width, and iid $\mathcal{U}(0, \frac{1}{d_t^2})$ spectral weights. Initializing the weights this way is the standard default for FNO. This model uses the GeLU activation function standard in FNO. Next we examine the use of ReLU activation instead of GeLU. Finally, we investigate non-periodic positional encoding.

Discretization error for random weights models The relative error of the state at each layer versus the discretization for inputs of varying regularity may be seen for the default model, the ReLU model, and the non-periodic

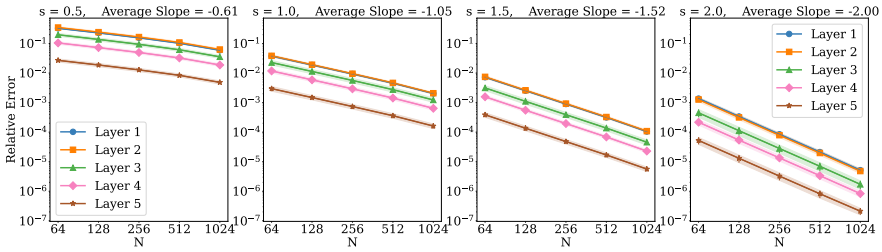


Figure 4.1: Relative error versus N and s for an FNO with default weight initialization.

position encoding model in Figures 4.1,4.2, and 4.3 respectively. In these figures, from left to right, $s \in \{0.5, 1, 1.5, 2\}$ where $v_0 \in H^{s-}$. The uncertainty shading indicates two standard deviations from the mean over five inputs to the FNO.

As can be seen in Figure 4.1 for the model with the default weight initialization, the empirical behavior of the error matches the behavior expected from Theorem 4.3.2. One question that arises from Figure 4.1 is why the error decreases as the number of layers increases; this is an effect of the magnitude of the weights. When the model weights are multiplied by 10, then the error begins to increase with the number of layers. A figure illustrating this phenomenon may be found in Appendix C.7, where additional weight initializations are explored as well.

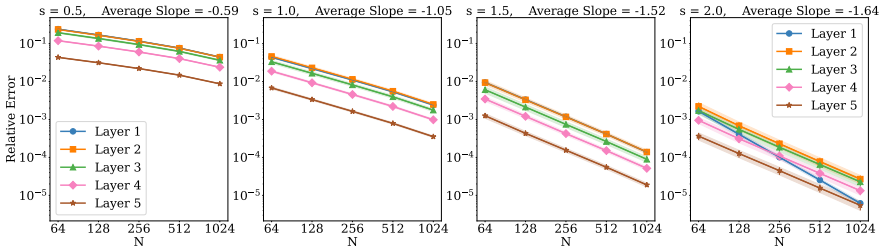


Figure 4.2: Relative error versus N and s for a default FNO with a ReLU activation.

The results shown in Figure 4.2 justify the use of the GeLU activation function, which belongs to C^∞, over the ReLU activation function, which is only Lipschitz. The figure shows that the benefit of having sufficiently smooth inputs is negated by the ReLU activation: the error decay is limited. Note that this effect does not occur for the first layer since at that point ReLU has been applied once, and the Fourier transform is not applied to the output of an activation function until the second layer. Additionally, we do not observe the effect of ReLU until the input has regularity greater than $s = 1.5$ since the

ReLU activation function has regularity of $s = 1.5$

A similar effect to the ReLU model occurs when non-periodic positional encoding information is appended to the input, as is standard in practical FNO usage; see Figure 4.3. Since this grid data has a jump discontinuity across the boundary of $[0, 1]^d$, it has regularity of $s = 0.5$, so the convergence rate never achieves N^{-1}. These results suggest caution when using positional encoding information with smooth input data; periodic positional encodings may be preferred.

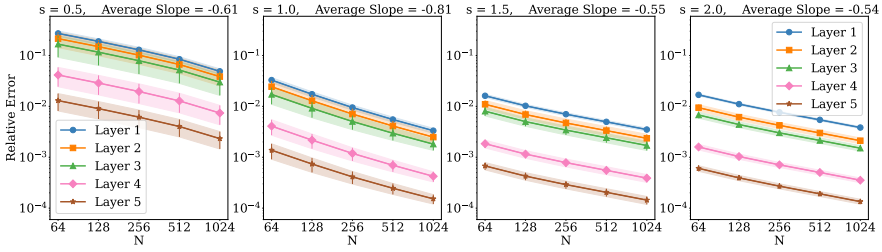
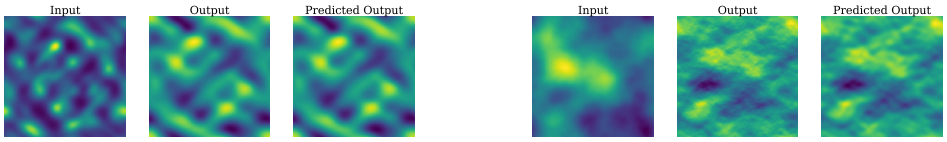


Figure 4.3: Relative error versus N and s for a default FNO with non-periodic position encoding appended to the input.

Experiments with trained networks

In this subsection, we consider two different maps and train FNOs on data from each map. The first map is a PDE solution map in two dimensions whose solution is at least as regular as the input function. The second map is a simple gradient, but in this setting the output data of the gradient is naturally less regular by one Sobolev smoothness exponent than that of the input function. In both experiments, periodic positional encoding information is appended to the inputs.



(a) Data for the PDE Solution FNO. (b) Data for the Gradient FNO.

Figure 4.4: Visualization of the input and output data for the trained model examples.

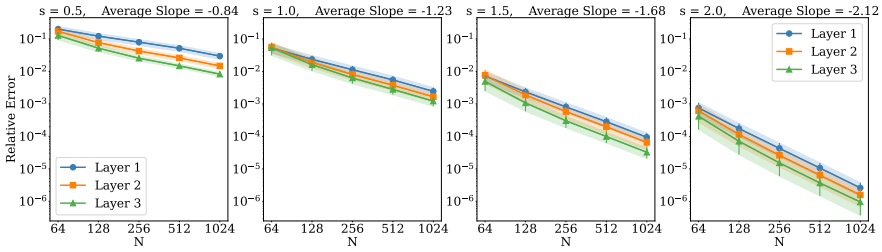


Figure 4.5: Error versus discretization for inputs of varying regularity for the

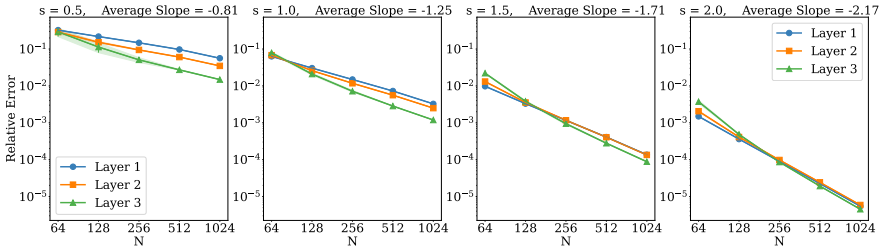


Figure 4.6: Error versus discretization for inputs of varying regularity for the FNO trained on data corresponding to a gradient map.

Example 1: PDE solution model In this example, we train an FNO to approximate the solution map of the following PDE:

$$\nabla \cdot (\nabla \chi A) = \nabla \cdot A, \quad y \in \mathbb{T}^2 \tag{4.4.1}$$

$$\chi \text{ is } 1-\text{periodic}, \int_{\mathbb{T}^2} \chi \, \mathrm{d}y = 0. \tag{4.4.2}$$

Here, the input $A : \mathbb{T}^2 \mapsto \mathbb{R}^{2\times 2}$ is symmetric positive definite at every point in the domain $\mathbb{T}^2$ and is bounded and coercive. For the output data we take the first component of $\chi : \mathbb{T}^2 \mapsto \mathbb{R}^2$. In our experiments the model is trained to $< 5\%$ relative L^2 test error. A visualization of the data is in Figure 4.4a.

The error versus discretization analysis can be seen in Figure 4.5. The error decreases slightly faster than predicted by the theory; a potential explanation is that the trained model itself has a smoothing effect that is not exploited in our analysis.

Example 2: gradient map In the final example, we train an FNO to approximate a simple gradient map $u \mapsto \nabla u$.

The training data consists of iid Gaussian random field inputs with regularity $s = 2$. Since a gradient reduces regularity, we expect the model outputs to approximate functions with regularity $s = 1$, which is at odds with the smoothness-preserving properties of the FNO described by theory.

The error versus discretization for inputs of various smoothness is shown in Figure 4.6. The error decreases according the the smoothness of the input despite the smoothness-decreasing properties of the data. Indeed, the model does produce more regular predicted outputs than the true gradient, as can be seen in Figure 4.4b where the predicted output is visibly smoother than the true output.

Speeding Up Training via Adaptive Subsampling

The fact that the FNO architecture and its parametrization are independent of the numerical discretization allows for increased flexibility. Specifically, it is possible to adaptively choose an optimal discretization for a given objective. Furthermore, the error decomposition (4.1.1) invites an exploitation to optimize computational training time.

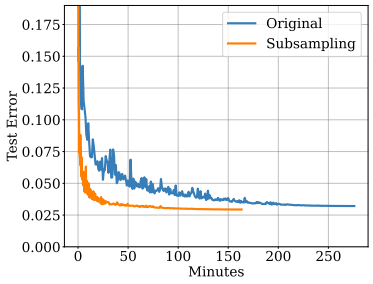


Figure 4.7: Adaptive grid refinement leads to greater training efficiency.

The basic idea of the proposed approach is that, during training, it is not necessary to compute model outputs to a numerical accuracy that is substantially better than the model discrepancy. This suggests an adaptive choice of the numerical discretization, where we employ a coarser grid during the early phase of training and refine the grid in later stages. In practice, we realize this idea by introducing a subsampling scheduler. The subsampling scheduler tracks a validation error on held out data and adaptively changes the numerical resolution via suitable subsampling of the training data. Starting from a coarse resolution, we iteratively double the grid size once the validation error plateaus.

We train FNO for the elliptic PDE (4.4.1) with and without the subsampling scheduler; details are contained in Appendix C.9. Compared to training without subsampling, training with a subsampling scheduler requires the same number of forward and backward passes over the network for the training and test set, plus an additional overhead due to the validation set. Since we are mainly interested in the training time, our choice of adding validation samples, rather than performing a training/validation split of the original training samples, ensures that computational timings are not skewed in favor of subsampling. Over the course of training, we iterate through the following grid

sizes: 32x32, 64x64, and 128x128. Our criterion for a plateau is that the validation error has not improved for 40 training epochs. The results of training with and without subsampling scheduler for the PDE solution model (4.4.1) are shown in Figure 4.7. We observe that training time can be substantially reduced with subsampling. This points to the potential benefits of developing adaptive numerical methods for model evaluation within operator learning.

4.5 Conclusions

In this chapter, we analyze the discretization error that results from implementation of Fourier Neural Operators (FNOs). We bound the L^2 norm of the error in Theorem 4.3.2, proving an upper bound that decreases asymptotically as N^{-s}, where N is the discretization in each dimension, and s is the input regularity. We show empirically that FNOs with random weights chosen as the default FNO weights for training behave almost exactly as the theory predicts. Furthermore, our theory and experiments justify the use of the GeLU activation function in FNO over ReLU, as the former preserves regularity. Additional analyses on trained models show that the error behaves less predictably in relation to our theory in the low-discretization regime. Finally, we use the decomposition of model error and discretization error to propose an adaptive subsampling algorithm for decreasing training time with operator learning. As FNOs become a more common tool in scientific machine learning, understanding the various sources of error is critical. By bounding FNO discretization error and demonstrating its behavior in numerical experiments, we understand its effect on learning and the potential to minimize computational costs by an adaptive choice of numerical resolution.

Chapter 5

AN OPERATOR LEARNING PERSPECTIVE ON PARAMETER-TO-OBSERVABLE MAPS

This chapter is adapted from the following publication:

[1] Daniel Zhengyu Huang, Nicholas H. Nelsen, and Margaret Trautner. "An operator learning perspective on parameter-to-observable maps". In: *Foundations of Data Science* 7.1 (2025), pp. 163–225. DOI: 10.3934/fods.2024037.

Computationally efficient surrogates for parametrized physical models play a crucial role in science and engineering. Operator learning provides data-driven surrogates that map between function spaces. However, instead of full-field measurements, often the available data are only finite-dimensional parametrizations of model inputs or finite observables of model outputs. Building on Fourier Neural Operators, this chapter introduces the Fourier Neural Mappings (FNMs) framework that is able to accommodate such finite-dimensional vector inputs or outputs. The chapter develops universal approximation theorems for the method. Moreover, in many applications the underlying parameter-to-observable (PtO) map is defined implicitly through an infinite-dimensional operator, such as the solution operator of a partial differential equation. A natural question is whether it is more data-efficient to learn the PtO map end-to-end or first learn the solution operator and subsequently compute the observable from the full-field solution. We explore this question numerically using the FNM framework via three nonlinear PtO maps and demonstrate the benefits of the operator learning perspective that this chapter adopts.

5.1 Introduction

Operator learning has emerged as a methodology that enables the machine learning of maps between spaces of functions. Many surrogate modeling tasks in areas such as uncertainty quantification, inverse problems, and design optimization involve a map between function spaces, such as the solution operator of a partial differential equation (PDE). However, the primary quantities of

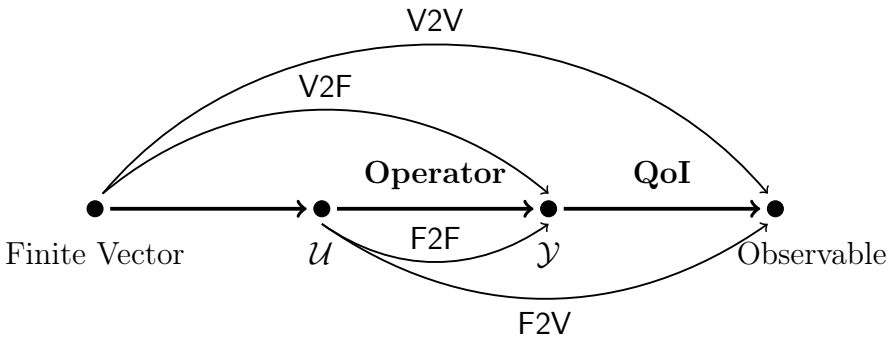


Figure 5.1: Illustration of the factorization of an underlying PtO map into a QoI and an operator between function spaces. Also shown are the four variants of input and output representations considered in this work. Here, $\mathcal{U}$ is an input function space and $\mathcal{Y}$ is an intermediate function space.

interest (QoI) in these tasks are usually just a finite number of design parameters or output observables. This may be because full-field data, such as initial conditions, boundary conditions, and solutions of PDEs, are not accessible from measurements or are too expensive to acquire. The prevailing approach then involves emulating the parameter-to-observable (PtO) map instead of the underlying solution map between function spaces. Yet, it is natural to wonder if the success of operator learning in the function-to-function setting can be brought to bear in this more realistic setting where inputs or outputs may necessarily be finite-dimensional vectors. To this end, the present chapter introduces Fourier Neural Mappings (FNMs) as a way to extend operator learning architectures such as the Fourier Neural Operator (FNO) to finite-dimensional input and output spaces in a manner compatible with the underlying operator between infinite-dimensional spaces. The admissible types of FNM models considered in this work are visualized in Figure 5.1.

Nevertheless, it is possible to accommodate finite-dimensional inputs or outputs through other means. For instance, one could lift a finite-dimensional input vector to a function by expanding in predetermined basis functions, apply traditional operator learning architectures to the full-field function space data, and then directly compute a known finite-dimensional QoI from the output function. In contrast, the end-to-end FNM approach in this work is fully data-driven and operates directly on finite-dimensional vector data without the need for pre- and postprocessing. A natural question is whether one of these two approaches achieves better accuracy than the other when the goal is to predict certain QoIs. In this chapter, we address this important question from

a numerical perspective. The theoretical perspective is addressed in [21], in the simplified setting of Bayesian linear functionals. Indeed, it has been empirically observed in various nonlinear problems ranging from electronic structure calculations [109] to metamaterial design [110] that data-driven methods that predict the full-field response of a system are superior to end-to-end approaches for the same downstream tasks or QoIs.

Throughout the chapter, we refer to learning a function-valued map as *full-field learning*. Given such a learned map, various known QoIs may be directly computed from the output of the map. On the other hand, we refer to the direct estimation of the map from an input to the observed QoI as *end-to-end learning*. This terminology distinguishes between output spaces. When either the input or output is a finite vector and the other is infinite-dimensional, we label the learning approach as "vector-to-function" (V2F) or "function-to-vector" (F2V), respectively, to avoid ambiguity. The abbreviations V2V and F2F for "vector-to-vector" and "function-to-function" are analogous.

Contributions

In this chapter, we make the following contributions.

(C1) We introduce FNMs as a function space architecture that is able to accommodate finite-dimensional vector inputs, outputs, or both.

(C2) We prove universal approximation theorems for FNMs.

(C3) We perform numerical experiments with FNMs in three different examples — an advection–diffusion equation, flow over an airfoil, and an elliptic homogenization problem — that show empirical evidence that the theoretical linear insight from [21] remain valid for nonlinear maps.

Related work

Several works have established neural operators as a viable tool for scientific machine learning. The general neural operator formalism is described in [52] and contains several subclasses including DeepONet [23], graph neural operator [111, 112], and FNO [19]. These architectures allow for function data evaluated at different grid points or resolutions to all be used with the same model. In particular, the FNO is primarily parametrized in Fourier space. It exploits the fact that the Fourier basis spans L^2 and uses the efficient Fast

Fourier Transform (FFT) algorithm for computations. The idea of parametrizing operators in Fourier space is explored in earlier works as well [69, 113]. The FNO has been shown to be applicable both to domains other than the torus and to nonuniform meshes [114, 115]. These neural operators have been used in various areas of application, including climate modeling [116], fracture mechanics [117], and catheter design [97]. In several of these applications, neural operators have been implemented with finite-dimensional vector inputs or outputs by using constant functions as substitutes for finite vectors, which is theoretically justified by statements of universal approximation [56], or by using other hand-designed maps. However, learning a constant function as a representation for a constant is unnatural and computationally wasteful; it is desirable to substitute a more suitable architecture. The present chapter develops FNMs that extend neural operators to this important setting while retaining desirable universal approximation properties.

The theory of scientific machine learning falls broadly into three tiers. In the first tier, universal approximation results [118, 4, 119, 120] use classical approximation theory to guarantee that the architecture is capable of representing maps from within a class of interest to any desired accuracy. Some of the proofs of these results contain constructive arguments, but the corresponding architectures are usually not as empirically effective as those that solely come with existence results. Examples of constructive arguments for operator approximation are contained in [100], which constructs ReLU neural networks, and [121], which uses randomized numerical linear algebra to sketch Green's functions for linear elliptic PDEs. Each of these works also comes equipped with convergence rates with respect to model size and data size, respectively; these rates form the second tier of operator learning theory. Many papers in this tier prove bounds on the required model size, i.e., parameter complexity [122, 89, 20, 123, 124, 125, 126, 127]. Some are able to obtain sample complexity bounds, although most results are restricted to linear or kernelized settings [128, 129, 130, 131, 132, 133, 134, 135, 136]. The third tier of theory describes the likelihood of actually obtaining an accurate approximation through optimization. While some results along these lines exist for linear models, linear maps, and constructive operators [131, 137, 138], they are absent for the class of neural operators optimized through variants of stochastic gradient descent (SGD). This is the class that has proven empirically most effective in applications thus far and is the class used in this work.

Recent work proposes and analyzes a kernelized deep learning method for nonlinear functionals [139]. The idea of such neural functionals, a subclass of the FNMs proposed in this work, is not new. One appearance is in the context of a function space discriminator for generative adversarial networks [140]. However, that work uses only a single bounded linear functional that is appended to the output of a FNO and is parametrized by a standard neural network. This is a special case of our FNMs for F2V maps. Another paper that shares similar ideas to ours is [141]. There, the authors also formulate a V2V neural network approach that maps through a latent 1D function space. However, their encoder and decoder maps are prescribed by hand-picked basis functions, while for FNMs the encoder and decoder maps are learned from data.

In this work, three potential applications are highlighted. The first application is an advection–diffusion model where the input is a velocity field and the output is the state at a fixed future time. This problem is considered a benchmark for scientific machine learning [142]. Some theoretical approximation rates for it have been developed for DeepONet in the F2F setting [122]. The second application centers on the compressible flow over an airfoil, i.e., a wing cross section. This experiment is explored for FNO in [114] and used as a shape optimization example in the F2F setting for DeepONet in [143] and for reduced basis networks in [144]. Several other related works devise V2V-based neural network approaches and novel training strategies for this aerodynamics problem [145, 146, 147, 148]. The third application involves learning the homogenized elasticity coefficient for a multiscale elliptic PDE. This example is explored in detail for FNO in [56] and for other constitutive laws in [29]. For the Darcy flow — or scalar coefficient — setting of this equation, other work adopts the F2F setting to efficiently compute QoIs [149]. For each of these applications, we compare the generalization error performance of all four F2F, F2V, V2F, and V2V variants of FNMs as well as standard fully-connected neural networks.

Outline

The remainder of this chapter is organized as follows. We define the architecture of FNMs as a slight adjustment of FNOs in Section 5.2 (Contribution (C1)) and confirm that FNMs retain desirable properties of FNOs such as universal approximation in Section 5.3 (Contribution (C2)). Section 5.5 provides numerical experiments that compare end-to-end and full-field learn-

ing with FNMs with both finite- and infinite-dimensional input space representations for predicting QoIs in several nonlinear PDE problems (Contribution (C3)). Concluding remarks are given in Section 5.6. All proofs are provided in Appendix D.1.

5.2 Neural mappings for finite-dimensional vector data

In this section, we recall the FNO architecture (Subsection 5.2) and describe modifications of it to form FNMs (Subsection 5.2).

A review of neural operators

Let $\mathcal{U} = \mathcal{U}(\mathcal{D}; \mathbb{R}^{d_u})$ and $\mathcal{Y} = \mathcal{Y}(\mathcal{D}; \mathbb{R}^{d_y})$ be Banach function spaces over Euclidean domain $\mathcal{D} \subset \mathbb{R}^d$. Finite-dimensional fully-connected neural networks are repeated compositions of affine mappings alternating with pointwise nonlinearities. To extend this framework to the infinite-dimensional function space setting, depth T neural operators from $\mathcal{U}$ to $\mathcal{Y}$ take the form

$$\Psi^{(\mathrm{NO})}(u) := \big(\mathcal{Q} \circ \mathscr{L}_T \circ \mathscr{L}_{T-1} \circ \cdots \circ \mathscr{L}_1 \circ \mathcal{S}\big)(u) \quad \text{for all} \quad u \in \mathcal{U}\,, \tag{5.2.1}$$

where $\mathcal{S}$ is a pointwise-defined local lifting operator, $\mathcal{Q}$ is a pointwise-defined local projection operator, and for each $t \in [T]$, the layer $\mathscr{L}_t \colon \mathcal{B}_{t-1} \to \mathcal{B}_t$ is a nonlinear map between appropriate Banach function spaces $\mathcal{B}_{t-1}(\mathcal{D}; \mathbb{R}^{d_{t-1}}) \subset L^2(\mathcal{D}; \mathbb{R}^{d_{t-1}})$ and $\mathcal{B}_t(\mathcal{D}; \mathbb{R}^{d_t}) \subset L^2(\mathcal{D}; \mathbb{R}^{d_t})$. The map $\mathscr{L}_t$ is the composition of a local (and usually nonlinear) operator with a nonlocal affine kernel integral operator [52].

The FNO is a specific instance of the class of neural operators (5.2.1). Let $\mathcal{D} = \mathbb{T}^d$. Then for FNO, the form of the layer $\mathscr{L}_t \colon \mathcal{B}_{t-1}(\mathbb{T}^d; \mathbb{R}^{d_{t-1}}) \to \mathcal{B}_t(\mathbb{T}^d; \mathbb{R}^{d_t})$ is given by $v \mapsto \mathscr{L}_t(v)$, where for any $x \in \mathbb{T}^d$, it holds that

$$\big(\mathscr{L}_t(v)\big)(x) = \sigma_t\big(W_t v(x) + (\mathcal{K}_t v)(x) + b_t(x)\big)\,. \tag{5.2.2}$$

In (5.2.2), $W_t \in \mathbb{R}^{d_t \times d_{t-1}}$ is a weight matrix, $b_t \colon \mathbb{T}^d \to \mathbb{R}^{d_t}$ is a bias function, and $\mathcal{K}_t$ is a convolution operator given, for $v \colon \mathbb{T}^d \to \mathbb{R}^{d_{t-1}}$ and any $x \in \mathbb{T}^d$, by the expression

$$(\mathcal{K}_t v)(x) = \left\{ \sum_{k \in \mathbb{Z}^d} \left(\sum_{j=1}^{d_{t-1}} (P_t^{(k)})_{\ell j} \langle \psi_k, v_j \rangle_{L^2(\mathbb{T}^d; \mathbb{C})} \right) \psi_k(x) \right\}_{\ell \in [d_t]} \in \mathbb{R}^{d_t}\,. \tag{5.2.3}$$

In the preceding display, the $\psi_k = e^{2\pi \mathrm{i} \langle k, \cdot \rangle_{\mathbb{R}^d}}$ are the complex Fourier basis elements of $L^2(\mathbb{T}^d; \mathbb{C})$ and $P_t^{(k)} \in \mathbb{C}^{d_t \times d_{t-1}}$ are the learnable parameters of the

integral operator $\mathcal{K}_t$ for each $k \in \mathbb{Z}^d$. The functions $\sigma_t \colon \mathbb{R} \to \mathbb{R}$ are nonlinear activations that act pointwise when applied to vectors. Additional details of more general versions and computational implementations of the FNO may be found in [20, 52, 114].

Though the internal FNO layers $\{\mathscr{L}_t\}$ in (5.2.2) and (5.2.3) are defined on the periodic domain $\mathbb{T}^d$, it is possible to apply the FNO to other d-dimensional domains $\mathcal{D} \neq \mathbb{T}^d$. Define Banach spaces $\mathcal{B}_{\text{in}}(\mathcal{D}; \mathbb{R}^{d_0})$ and $\mathcal{B}_{\text{out}}(\mathcal{D}; \mathbb{R}^{d_T})$ and introduce an operator $\mathcal{E} \colon \mathcal{B}_{\text{in}} \to \mathcal{B}_0$. Then, replace $\mathcal{S}$ in (5.2.1) with $\mathcal{E} \circ \mathcal{S}$. Similarly, let $\mathcal{R} \colon \mathcal{B}_T \to \mathcal{B}_{\text{out}}$ be an operator that maps back to functions on the desired domain $\mathcal{D}$ and replace $\mathcal{Q}$ in (5.2.1) with $\mathcal{Q} \circ \mathcal{R}$. These modifications to the lifting and projecting components yield the final FNO architecture

$$\Psi^{(\text{FNO})} = \mathcal{Q} \circ \mathcal{R} \circ \mathscr{L}_T \circ \mathscr{L}_{T-1} \circ \cdots \circ \mathscr{L}_1 \circ \mathcal{E} \circ \mathcal{S} \,. \tag{5.2.4}$$

In practice, the map $\mathcal{E}$ is usually represented by zero padding the input domain and $\mathcal{R}$ by restricting to the output domain of interest.

The neural mappings framework

The neural operator architecture described in Section 5.2 only accepts inputs, outputs, and intermediate states that are elements of function spaces. Finite-dimensional vector inputs, outputs, and states are not directly compatible with neural operators. We propose *neural mappings*, which lift this restriction through two fundamental building blocks. The first, linear functional layers, map from function space to finite dimensions. The second, linear decoder layers, map from finite dimensions to function space. We combine these two building blocks with standard iterative neural operator layers to form several classes of nonlinear and function space consistent architectures.

Instating the neural operator notation from Section 5.2, we define a *linear functional layer* $\mathscr{G} \colon \mathcal{B}_{T-1} \to \mathbb{R}^{d_T}$ and a *linear decoder layer* $\mathscr{D} \colon \mathbb{R}^{d_0} \to \mathcal{B}_1$ to be maps of the form

$$\begin{aligned} h \mapsto \mathscr{G} h &\coloneqq \int_{\mathcal{D}} \kappa(x) h(x) \, dx \,, \quad \text{where} \quad \kappa \colon \mathcal{D} \to \mathbb{R}^{d_T \times d_{T-1}} \,, \quad \text{and} \\ z \mapsto \mathscr{D} z &\coloneqq \kappa(\cdot) z \,, \quad \text{where} \quad \kappa \colon \mathcal{D} \to \mathbb{R}^{d_1 \times d_0} \,, \end{aligned} \tag{5.2.5}$$

respectively. The linear functional layer $\mathscr{G}$ takes a vector-valued function h and integrates it against a fixed matrix-valued function κ to produce a finite vector output. In duality to $\mathscr{G}$, the linear decoder layer $\mathscr{D}$ takes as input a

finite vector z and multiplies it by a fixed matrix-valued function κ to produce an output function. The functions κ are the sole learnable parameters of these two layers.

Although $\mathscr{G}$ and $\mathscr{D}$ may be incorporated into general neural operators (5.2.1), we will specialize our method to the FNO. In anticipation of this periodic setting, we view $\mathscr{G}$ as a Fourier linear functional layer by replacing $\mathcal{D}$ in (5.2.5) by the torus $\mathbb{T}^d$ and using Fourier series to expand $\mathscr{G}$ as

$$h \mapsto \mathscr{G}h = \left\{ \sum_{k\in\mathbb{Z}^d} \left(\sum_{j=1}^{d_{T-1}} P_{\ell j}^{(k)} \langle \psi_k, h_j \rangle_{L^2(\mathbb{T}^d;\mathbb{C})} \right) \right\}_{\ell\in[d_T]} \in \mathbb{R}^{d_T}, \tag{5.2.6}$$

where we recall that $\{\psi_k\}$ is the Fourier basis of $L^2(\mathbb{T}^d;\mathbb{C})$. In (5.2.6), the entries of the matrices $\{P^{(k)}\} \subset \mathbb{C}^{d_T\times d_{T-1}}$ correspond to the Fourier coefficients of the function κ in (5.2.5). The calculation leading to the convergent series formula (5.2.6) uses Parseval's theorem to equate the L^2 (5.2.5) and ℓ^2 (5.2.6) inner products. Similar calculations show that, on the torus $\mathbb{T}^d$, the map $\mathscr{D}$ takes the form

$$z \mapsto \mathscr{D}z = \left\{ \sum_{k\in\mathbb{Z}^d} \left(P^{(k)}z\right)_j \psi_k \right\}_{j\in[d_1]}, \quad \text{where} \quad P^{(k)} \in \mathbb{C}^{d_1\times d_0}. \tag{5.2.7}$$

Just like for the FNO kernel integral layers (5.2.3), the expressions (5.2.6) and (5.2.7) are efficiently implemented and learned in Fourier space.

We are now in a position to define the general FNMs architecture.

***Definition* 5.2.1** (Fourier Neural Mappings)**.** Let $Q\colon \mathbb{R}^{d_T} \to \mathbb{R}^{d_y}$ and $S\colon \mathbb{R}^{d_u} \to \mathbb{R}^{d_0}$ be finite-dimensional maps. For $\{\mathscr{L}_t\}$ defined as in (5.2.4) and $\mathscr{G}$ and $\mathscr{D}$ defined as in (5.2.6) and (5.2.7), let

$$\Psi^{(\mathrm{FNM})} := Q \circ \mathscr{G} \circ \mathscr{L}_{T-1} \circ \cdots \circ \mathscr{L}_2 \circ \mathscr{D} \circ S \tag{5.2.8}$$

be the base level map. The *Fourier Neural Mappings* architecture is composed of the following four main models that are obtained by modifying the base map:

(M-V2V) vector-to-vector (V2V): $\Psi^{(\mathrm{FNM})}$ in (5.2.8) as written, thus mapping finite vector inputs to finite vector outputs;

(M-V2F) vector-to-function (V2F): $\Psi^{(\mathrm{FNM})}$ with operator $\mathscr{G}$ in (5.2.8) replaced by $\mathcal{R} \circ \mathscr{L}_T$, where $\mathcal{R}$ and $\mathscr{L}_T$ are as in (5.2.4) and (5.2.2), respectively,

and Q in (5.2.8) now viewed as a pointwise-defined operator acting on vector-valued functions;

(M-F2V) function-to-vector (F2V): $\Psi^{(\mathrm{FNM})}$ with operator $\mathscr{D}$ in (5.2.8) replaced by $\mathscr{L}_1 \circ \mathcal{E}$, where $\mathscr{L}_1$ and $\mathcal{E}$ are as in (5.2.2) and (5.2.4), respectively, and S in (5.2.8) now viewed as a pointwise-defined operator acting on vector-valued functions;

(M-F2F) function-to-function (F2F): $\Psi^{(\mathrm{FNM})}$ with modifications (M-V2F) and (M-F2V), thus the resulting architecture is the standard FNO $\Psi^{(\mathrm{FNO})}$ (5.2.4).[1]

◊

When the (M-F2V) FNM is of primary interest, we sometimes call this architecture *Fourier Neural Functionals*. Similarly, we may also call the (M-V2F) FNM a *Fourier Neural Decoder*.

5.3 Universal approximation theory for Fourier Neural Mappings

In this section, we establish universal approximation theorems for FNMs; this is a confirmation that the architectures maintain this desirable property of neural operators. The results are stated for the cases of the F2V and V2F architectures; the case of V2V trivially follows. Similar results also hold for general neural mappings by invoking the appropriate universal approximation theorems for general neural operators from [20, Section 9.3] and for the topology induced by Lebesgue–Bochner norms, i.e., average error with respect to a probability measure supported on the input space. For more details regarding these extensions, see [52, Theorems 11–14, Section 9.3, pp. 55–57] and [20, pp. 12–14 and Theorem 18]. Our proofs, which are collected in Appendix D.1, use arguments based on constant functions that are similar to those used to prove universal approximation theorems at the level of operators.

The approximation theory in this section relies on the following assumption.

***Assumption* 5.3.1** (activation function)**.** All nonlinear layers $\{\mathscr{L}_t\}_{t=1}^T$ from (5.2.2) have the same non-polynomial and globally Lipschitz activation function $\sigma \in C^\infty(\mathbb{R};\mathbb{R})$. ◊

[1] Notice that yet another function-to-function FNM architecture is possible by exchanging the roles of $\mathscr{G}$ and $\mathscr{D}$ in (5.2.8); this is a nonlinear Fourier neural autoencoder.

We note that in practice, the final Fourier layer activation function is often set to be the identity. Moreover, the bias functions b_t in $\mathscr{L}_t$ are typically chosen to be constant functions. The universal approximation theory does not distinguish these differences. Additionally, to align with the existing theory developed in [20], our existence proofs rely on a reduction to the setting that

(*i*) the channel dimension d_t is constant across all layers, say $d_t = d_v \in \mathbb{N}$ for all $t \in [T]$, and

(*ii*) the maps S and Q in (5.2.8) are linear and act pointwise on functions.

These conditions are certainly special cases of nonconstant channel dimension and nonlinear lifting and projection maps, respectively. Hence, the forthcoming universality properties still hold for more sophisticated architectures that deviate from conditions (*i*) and (*ii*), such as those used in Section 5.5 in this work.

Our first result delivers a universal approximation result for Fourier Neural Functionals, i.e., the F2V setting. Appendix D.1 contains the proof.

***Theorem* 5.3.2** (universal approximation: function-to-vector mappings)**.** Let $s \geq 0$, $\mathcal{D} \subset \mathbb{R}^d$ be an open Lipschitz domain such that $\overline{\mathcal{D}} \subset (0,1)^d$, and $\mathcal{U} = H^s(\mathcal{D}; \mathbb{R}^{d_u})$. Let $\Psi^\dagger \colon \mathcal{U} \to \mathbb{R}^{d_y}$ be a continuous mapping. Let $K \subset \mathcal{U}$ be compact in $\mathcal{U}$. Under Assumption 5.3.1, for any $\varepsilon > 0$, there exist Fourier Neural Functionals $\Psi \colon \mathcal{U} \to \mathbb{R}^{d_y}$ of the form (5.2.8) with modification (M-F2V) such that

$$\sup_{u \in K} \left\| \Psi^\dagger(u) - \Psi(u) \right\|_{\mathbb{R}^{d_y}} < \varepsilon \,. \tag{5.3.1}$$

$\Diamond$

The approximation theorem for the Fourier Neural Decoder, i.e., the V2F case, is analogous.

***Theorem* 5.3.3** (universal approximation: vector-to-function mappings)**.** Let $t \geq 0$, $\mathcal{D} \subset \mathbb{R}^d$ be an open Lipschitz domain such that $\overline{\mathcal{D}} \subset (0,1)^d$, and $\mathcal{Y} = H^t(\mathcal{D}; \mathbb{R}^{d_y})$. Let $\Psi^\dagger \colon \mathbb{R}^{d_u} \to \mathcal{Y}$ be a continuous mapping. Let $\mathcal{Z} \subset \mathbb{R}^{d_u}$ be compact. Under Assumption 5.3.1, for any $\varepsilon > 0$, there exists a Fourier

Neural Decoder $\Psi\colon \mathbb{R}^{d_u} \to \mathcal{Y}$ of the form (5.2.8) with modification (M-V2F) such that

$$\sup_{z\in\mathcal{Z}} \left\| \Psi^\dagger(z) - \Psi(z) \right\|_{\mathcal{Y}} < \varepsilon\,. \tag{5.3.2}$$

$\Diamond$

The proof may also be found in Appendix D.1. While perhaps not surprising, the results in Theorems 5.3.2 and 5.3.3 nonetheless show that the proposed FNM architectures are sensible for the tasks of approximating continuous function-to-vector or vector-to-function mappings.

5.4 Summary of Linear Functional Regression Theory

In this section, we give an informal summmary of the theoretical findings of the paper this chapter is based on ([21]) on data efficiency of learning parameter-to-observable maps in the setting of Bayesian nonparametric linear functional regression. In the analysis, the input space is always infinite dimensional, and the comparison is done between learning a function-valued output (full-field learning) and learning a finite vector-valued output (end-to-end learning). Let H be an infinite-dimensional real separable Hilbert space and consider the linear functional map $f^\dagger = q^\dagger \circ L^\dagger : H \to \mathbb{R}$ for linear functional $f^\dagger$, the *quantity of interest* (QoI) map and self-adjoint linear operator $L^\dagger$, the forward map. The theory compares error convergence rates in terms of the number of data when obtaining a Bayesian posterior estimator of $f^\dagger$ (end-to-end learning) versus for $L^\dagger$ (full-field learning). The question at hand is whether it is more data efficient to learn an estimator of $f^\dagger$ directly or to learn an estimator of $L^\dagger$ and then apply a known QoI map $q^\dagger$.

The result in a simplified setting is visualized in Figure 5.2. The convergence rate exponent describes how the upper bound on expected error behaves with respect to the number of data samples N; for an exponent of -1, the upper bound on the expected error of the estimator behaves like $\lesssim N^{-1}$. The smoothness of the problem is governed by the eigenvalue decay of operator $L^\dagger$ and the covariance operator of the input data distribution. This figure shows that for more regular QoI maps $q^\dagger$, full-field learning has a better convergence rate exponent than end-to-end learning. The situation is reversed for less regular QoI maps. The crossover point where both approaches have the same bound on their convergence rates occurs for a QoI regularity exponent of $r = -\frac{1}{2}$;

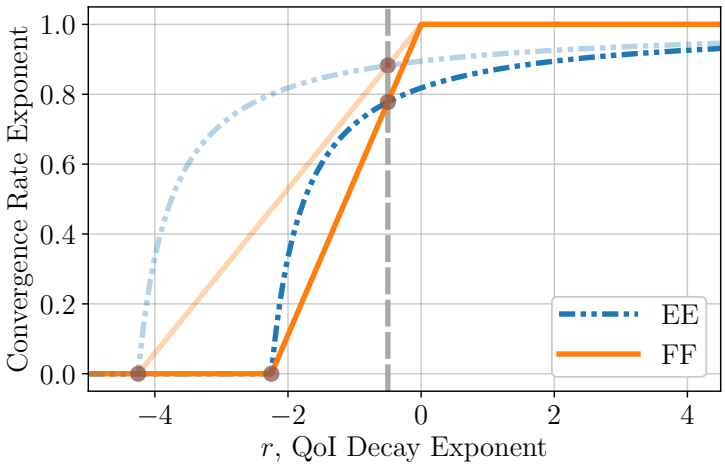


Figure 5.2: End-to-end vs. full-field convergence rate exponents as a function of QoI regularity exponent r. Larger exponents imply faster convergence rates. As the curves gets lighter, the smoothness of the problem increases. The vertical dashed line corresponds to $r = -1/2$, which is the transition point where end-to-end learning and full-field learning have the same rate.

this corresponds to the regularity of point evaluation. While these results give some intuition, we note that they are developed for the linear setting and only describe upper bounds on the expected error. We refer to [21] for the precise statements and proofs, but we reference the intuition summarized here in the following section containing numerical experiments.

5.5 Numerical experiments

We now perform numerical experiments with the proposed FNM architectures. These experiments have two main purposes. The first is to numerically implement and compare the various FNM models on several PtO maps of practical interest; the second is to qualitatively validate the theory described in 5.4 for such maps. We focus on nontrivial nonlinear problems with finite-dimensional observables that define the QoI maps. Although our linear theory from Section 5.4 does not apply to such nonlinear problems, we still observe qualitative validation of the main implications of the linear analysis. That is, for smooth enough QoIs, full-field learning is at least as data-efficient as end-to-end learning. Unlike the theory, however, our numerical results distinguish the two approaches only by constant factors and not by the actual convergence rates.

The continuum FNM architectures from Section 5.2 are implemented numeri-

cally by replacing all forward and inverse Fourier series calculations with their Discrete Fourier Transform counterparts. This enables fast summation of the series (5.2.3), (5.2.6), and (5.2.7) with the FFT. The inner products in these formulas are also computed with the FFT. In particular, the FFT performs Fourier space operations in the set $\{k \in \mathbb{Z}^d \colon \|k\|_{\ell^\infty([d];\mathbb{Z})} \leq K\}$ rather than over all $k \in \mathbb{Z}^d$. [2] The error that is produced by this approximation is analyzed in Chapter 4. In this case, we say that the FNM architecture has K modes. This is analogous to the mode truncation used in standard FNO layers (see, e.g., [19]). Additionally, since we work with real vector-valued functions, conjugate symmetry of the Fourier coefficients may be exploited to write the Fourier linear functional (5.2.6) and decoder (5.2.7) layers only in terms of the real part of the coefficients appearing in the summands. We also make a minor modification to the F2V and V2V FNMs. Since the discrete implementation of $\mathscr{G}$ in (5.2.6) requires discarding the higher frequencies in the input function, we define an auxiliary map $\mathscr{W} \colon h \mapsto \int_{\mathbb{T}^d} \mathsf{NN}(h(x))\,dx$ that makes use of all frequencies. Here $\mathsf{NN}(\,\cdot\,)$ is a one hidden layer fully-connected neural network (NN). Then we replace $\mathscr{G}$ in Definition 5.2.1 by the concatenated operator $(\mathscr{G}, \mathscr{W})^\top$.

Given a dataset of input–output pairs $\{(u_n, \widetilde{y}_n)\}_{n=1}^N$, we train a FNM Ψ_θ taking one of the forms given in Definition 5.2.1 (with the modifications from the preceding discussion) in a supervised manner by minimizing the average relative error

$$\frac{1}{N}\sum_{n=1}^{N} \frac{\|\widetilde{y}_n - \Psi_\theta(u_n)\|}{\|\widetilde{y}_n\|} \tag{5.5.1}$$

or the average absolute squared error

$$\frac{1}{N}\sum_{n=1}^{N} \|\widetilde{y}_n - \Psi_\theta(u_n)\|^2 \tag{5.5.2}$$

over the FNM's tunable parameters θ using mini-batch SGD with the ADAM optimizer. The choice of the loss function is dependent on the underlying problem. Moreover, the norm in the preceding displays are inferred from the space that the $\widetilde{y}_n$ takes values in (i.e., finite-dimensional vector or infinite-dimensional function output spaces). To avoid numerical instability in our actual computations, we add 10^{-6} to the denominator of the ratio in (5.5.1).

[2] In all numerical experiments to follow, $d = 1$ or $d = 2$.

***Remark* 5.5.1** (data discretization error)**.** In addition to the discretization error introduced by the discrete and implementable realizations of the continuum FNM architectures [150], there is another source of discretization error due to our choice of data generation procedure. Specifically, the training and test data in this section are generated from numerical solvers that discretely approximate an underlying continuum operator at a fixed resolution. The weights of the resulting trained FNMs have a complicated dependence on this discretization error. Although simpler operator learning architectures are stable to such errors [131, Example 3.9, pp. 5–6], no such results exist yet for neural operators. Furthermore, in line with most of the literature, the empirical convergence results that we numerically report in this section are for the test error with respect to the discretized operator or PtO map. Thus, there is an implicit assumption that this discretized operator is sufficiently resolved so that the computable but discrete test error is an accurate surrogate for the true but inaccessible test error with respect to the continuum operator. Alternative data acquisition strategies may mitigate these effects to some extent [151]. ◊

The numerical experiments are organized as follows. In Subsection 5.5, we extract the first four polynomial moments from the solution of a velocity-parametrized 2D advection–diffusion equation. Next, Subsection 5.5 considers the flow over an airfoil modeled by the steady compressible Euler equation. The PtO map sends the shape of the airfoil to the resultant drag and lift force vector. Last, we study an elliptic homogenization problem parametrized by material microstructure in Subsection 5.5. Here, the QoI returns the effective tensor of the material.

Moments of an advection–diffusion model

Our first model problem concerns a canonical advection–diffusion PDE in two spatial dimensions. This equation often arises in the environmental sciences and is useful for modeling the spread of passive tracers (e.g., contaminants, pollutants, aerosols), especially when the driving velocity field is coupled to another PDE such as the Navier–Stokes equation. Our setup is as follows. Let $\mathcal{D} = (0,1)^2$ be the spatial domain and n denote the unit inward normal vector to $\mathcal{D}$. For a prescribed time-independent velocity field $v\colon \mathcal{D} \to \mathbb{R}^2$, the state

$\phi\colon \mathcal{D} \times \mathbb{R}_{>0} \to \mathbb{R}$ solves

$$\begin{aligned} \partial_t \phi + \nabla \cdot (v\phi) - 0.05\Delta\phi &= g \quad \text{in} \quad \mathcal{D} \times \mathbb{R}_{>0}\,, \\ \mathsf{n} \cdot \nabla\phi &= 0 \quad \text{on} \quad \partial\mathcal{D} \times \mathbb{R}_{>0}\,, \\ \phi &= 0 \quad \text{on} \quad \mathcal{D} \times \{0\}\,. \end{aligned} \tag{5.5.3}$$

The time-independent source term g is a smoothed impulse located at $x_0 \coloneqq (0.2, 0.5)^\top$ and is defined for $x \in \mathcal{D}$ by

$$g(x) \coloneqq \frac{5}{2\pi(50)^{-2}} \exp\left(-\frac{\|x - x_0\|_{\mathbb{R}^2}^2}{2(50)^{-2}}\right).$$

We associate our input parameter with the velocity field v appearing in (5.5.3). Our parametrization takes the form

$$v = (u, 0)^\top\,, \quad \text{where} \quad u(x_1, x_2) = 3 + \sum_{j=1}^{d_{\mathrm{KL}}} \sqrt{\tau_j} z_j e_j(x_1) \tag{5.5.4}$$

for all $x = (x_1, x_2) \in \mathcal{D}$. Note that u is constant in the vertical x_2 direction. The eigenvalues $\{\tau_j\}_{j\in\mathbb{N}}$ and eigenfunctions $\{e_j\}_{j\in\mathbb{N}}$ correspond to the Mercer decomposition of a kernel obtained by restricting a Matérn covariance function over $\mathbb{R}$ to $(0, 1) \subset \mathbb{R}$. The covariance function has smoothness exponent 1.5 and lengthscale 0.25 [152]. We choose

$$z_j \overset{\text{i.i.d.}}{\sim} \mathsf{Uniform}([-1, 1]) \quad \text{for all} \quad j \in [d_{\mathrm{KL}}]\,.$$

Thus, up to normalization constants, the velocity field (5.5.4) is the (truncated) KL expansion of a subgaussian stochastic process. We take the input to either be the full x_1-velocity field $u\colon \mathcal{D} \to \mathbb{R}$ or the i.i.d. realizations $z \coloneqq (z_1, \ldots, z_{d_{\mathrm{KL}}})^\top$ of the random variables that affinely parametrize u.

Define the *nonlinear* QoI map $q^\dagger\colon L^4(\mathcal{D}; \mathbb{R}) \to \mathbb{R}^4$ as follows. First, for any $h \in L^2(\mathcal{D}; \mathbb{R})$, let

$$\overline{m}(h) \coloneqq \int_{\mathcal{D}} h(x)\,dx \quad \text{and} \quad \overline{s}(h) \coloneqq \left(\int_{\mathcal{D}} |h(x) - \overline{m}(h)|^2\,dx\right)^{1/2} \tag{5.5.5}$$

denote the mean and variance of the push-forward of the uniform distribution on $\mathcal{D} = (0, 1)^2$ under h, respectively. Then $q^\dagger = (q_1^\dagger, q_2^\dagger, q_3^\dagger, q_4^\dagger)^\top$ is given by

$$h \mapsto q^\dagger(h) \coloneqq \begin{pmatrix} \overline{m}(h) \\ \overline{s}(h) \\ \overline{s}(h)^{-3} \int_{\mathcal{D}} \big(h(x) - \overline{m}(h)\big)^3\,dx \\ -3 + \overline{s}(h)^{-4} \int_{\mathcal{D}} \big|h(x) - \overline{m}(h)\big|^4\,dx \end{pmatrix}. \tag{5.5.6}$$

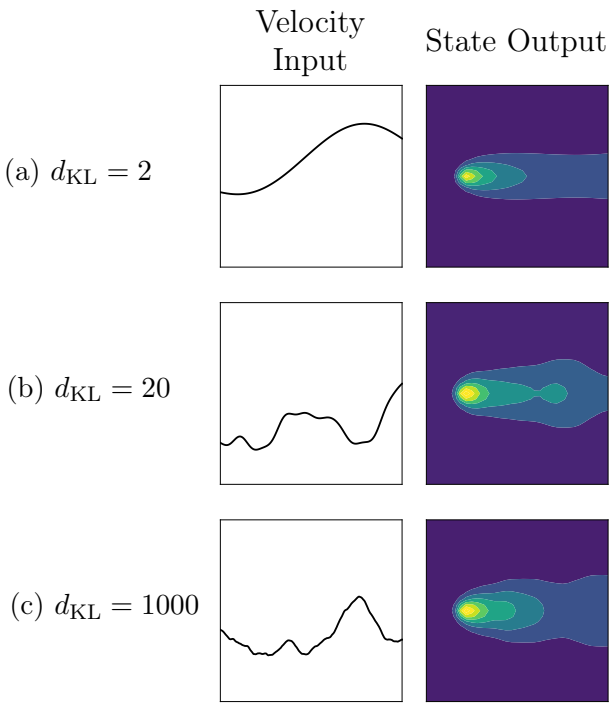
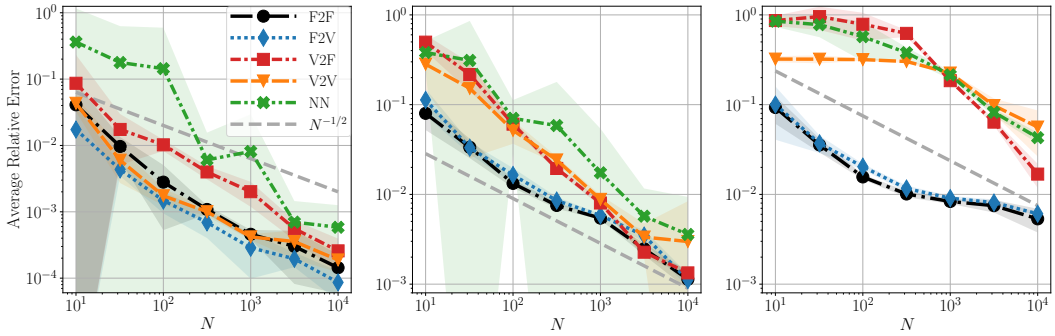


Figure 5.3: Visualization of the velocity-to-state map for the advection–diffusion model. Rows denote the dimension of the KL expansion of the velocity profile and columns display representative input and output fields.

Hence, $q_1^\dagger$ is the mean, $q_2^\dagger$ the standard deviation, $q_3^\dagger$ the skewness, and $q_4^\dagger$ the excess kurtosis. Our goal is to build FNM surrogates for the PtO map that sends the input representation (either the full velocity field or its finite number of i.i.d. coefficients) to the QoI values of the state ϕ at final time $t = 3/4$ (see Figure 5.3). Therefore, we train FNMs to approximate each of the following ground truth maps:

$$\begin{aligned}
&\Psi_{\mathrm{F2F}}^\dagger \colon u \mapsto \phi\big|_{t=3/4}\,, \\
&\Psi_{\mathrm{F2V}}^\dagger \colon u \mapsto q^\dagger\Big(\phi\big|_{t=3/4}\Big)\,, \\
&\Psi_{\mathrm{V2F}}^\dagger \colon z \mapsto \phi\big|_{t=3/4}\,, \quad \text{and} \\
&\Psi_{\mathrm{V2V}}^\dagger \colon z \mapsto q^\dagger\Big(\phi\big|_{t=3/4}\Big)\,.
\end{aligned}$$

The training data is obtained by solving (5.5.3) with a second-order Lagrange finite element method on a mesh of size 32×32 and Euler time step 0.01. For each $d_{\mathrm{KL}} \in \{2, 20, 1000\}$, we generate 10^4 i.i.d. data pairs for training, 1500 pairs for computing the test error (which is (5.5.1) over the 1500 test pairs instead of over the N training pairs), and 500 pairs for validation. All FNM



(a) $d_{\mathrm{KL}} = 2$ (b) $d_{\mathrm{KL}} = 20$ (c) $d_{\mathrm{KL}} = 1000$

Figure 5.4: Empirical sample complexity of FNM and NN architectures for the advection–diffusion PtO map (note that Figure 5.4a has a different vertical axis range). The shaded regions denote two standard deviations away from the mean of the test error over five realizations of the random training dataset indices, batch indices during SGD, and model parameter initializations.

models with 2D spatial input or output functions use 12 modes per dimension and a channel width of 32. For the V2V-FNM, we use a 1D latent function space with 12 modes and channel width of 96. We compare all FNM models to a standard fully-connected NN with 3 layers and constant hidden width 2048. These architecture settings were selected based on a hyperparameter search over the validation dataset for $d_{\mathrm{KL}} = 1000$ that mimics the parameter complexity experiments in [56, 94]. The models are trained on the relative loss (5.5.1) for 500 epochs in L^2 output space norm for functions and Euclidean norm for vectors. The optimizer settings include a minibatch size of 20, weight decay of 10^{-4}, and an initial learning rate of 10^{-3} which is halved every 100 epochs. We train 5 i.i.d. realizations of the models for various values of N and d_{KL} and report the results in Figure 5.4.

Figure 5.4 reveals several interesting trends. In general, training models to emulate the advection–diffusion PtO map with finite-dimensional vectors as input is more difficult than adopting function space input variants of the problem. The difficulty is further exacerbated as the dimension of the input vector (here, d_{KL}) increases. We hypothesis that this gap in performance would reduce if the vector input models received the weighted KL coefficients $\{\sqrt{\tau_j} z_j\}$ as input instead of the i.i.d. sequence $\{z_j\}$. This way the model would have access to decay information and hence an ordering of the coefficients. The standard finite-dimensional NN performs poorly across all KL expansion dimensions. The training of the NN is also quite erratic, as evidenced by the large green

shaded regions indicating large variance over multiple training runs. The output space seems to play less of a role than the input space. Indeed, the F2F and F2V FNMs with function space inputs generally achieve the lowest test error regardless of N and d_{KL}. The full-field F2F method slightly outperforms the end-to-end F2V method by a small constant factor (except for when $d_{\mathrm{KL}} = 2$). Since $q^\dagger$ is a smoothing QoI due to its integral definition, this observation aligns with the theoretical insights from Section 5.4. The fast convergence of some of the FNM models, especially for the low-dimensional cases $d_{\mathrm{KL}} = 2$ and $d_{\mathrm{KL}} = 20$, could potentially be explained by the lack of noise in the data, the smoothness of the QoIs, and the nonconvexity of the training procedure. When $d_{\mathrm{KL}} = 1000$, the problem is essentially infinite-dimensional. The function space input FNMs (F2F and F2V) exhibit a nonparametric decay of test error as expected.

Aerodynamic force exerted on an airfoil

Consider the following steady compressible Euler equation applied to an airfoil problem (see Figure 5.5), as introduced in [114]:

$$\begin{aligned} \nabla \cdot (\rho v) &= 0\,, \\ \nabla \cdot (\rho v v^\top + p\,\mathrm{Id}_{\mathbb{R}^2}) &= 0\,, \\ \nabla \cdot \big((E+p)v\big) &= 0\,. \end{aligned} \tag{5.5.7}$$

Here ρ is the fluid density, v is the velocity vector, p is the pressure, and E is the total energy. Equation (5.5.7) is equipped with the following far-field boundary conditions: $\rho_\infty = 1$, $p_\infty = 1$, $M_\infty = 0.8$, and $\mathsf{AoA} = 0$, where M_∞ is the Mach number and AoA is the angle of attack. This setup indicates that the flow condition is in the transonic regime. Additionally, the no-penetration condition $v \cdot \mathsf{n} = 0$ is imposed at the airfoil, where n represents the inward-pointing normal vector to the airfoil. Additional mathematical details about the setup may be found in [145, 146, 147, 148].

In this context, we are interested in solving the aforementioned 2D Euler equation to predict the drag and lift performance of different airfoil shapes. Building fast yet accurate surrogates for this task facilities aerodynamic shape optimization [144, 143] for various design goals, such as maximizing the lift to drag ratio [114]. The drag and lift QoIs, which only depend on the pressure

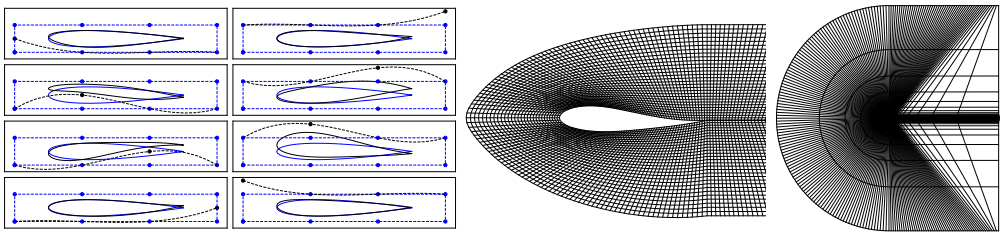


Figure 5.5: Flow over an airfoil. From left to right: visualization of the cubic design element and different airfoil configurations, guided by the displacement field of the control nodes; a close-up view of the C-grid surrounding the airfoil; the physical domain discretized by the C-grid.

on the airfoil, are given by the force vector

$$(\mathsf{Drag},\, \mathsf{Lift})^\top = \oint_{\mathcal{A}} p\mathsf{n}\, ds \in \mathbb{R}^2\,. \tag{5.5.8}$$

Here $\mathcal{A}$ denotes the closed curve defined by the union of the upper and lower surfaces of the airfoil. Different airfoil shapes are generated following the design element approach [153] (Figure 5.5). The initial NACA-0012 shape is embedded into a "cubic" design element featuring 8 control nodes, and the initial shape is morphed into a different one following the displacement field of the control nodes of the design element. The displacements of control nodes are restricted to the vertical direction only. Consequently, the intrinsic dimension of the input is 7, as displacing all nodes in the vertical direction by a constant value does not change the shape of the airfoil.

To generate the training data, we used the traditional second-order finite volume method with the implicit backward Euler time integrator. The process begins by generating a new airfoil shape. Subsequently, a C-grid mesh [154] consisting of 221×51 quadrilateral elements is created around the airfoil with adaptation near the airfoil. In total, we generated 2000 training data and 400 test data with the vertical displacements of each control node being sampled from a uniform distribution $\mathsf{Uniform}([-0.05, 0.05])$.

Next, we will define the operator learning problem (see Figure 5.6). In the 2D setting, we aim to learn the entire pressure field. Let $\mathcal{D}_a$ represent the irregular physical domain parametrized by a, indicating the shape of the airfoil. The domain $\mathcal{D}_a$ is discretized by a structured C-grid [154]. We introduce a latent space $\mathcal{D} = [0, 1]^2$ and the deformation map $\phi_a \colon \xi \to x(\xi)$ between $\mathcal{D}$ and $\mathcal{D}_a$. Here the deformation map has an analytical format and maps the uniform grid

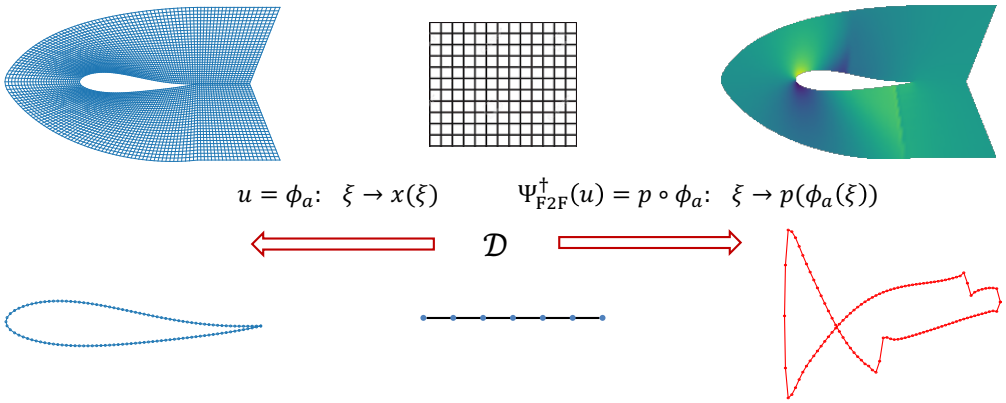


Figure 5.6: Flow over an airfoil. The 1D (bottom) and 2D (top) latent spaces are illustrated at the center; the input functions ϕ_a encoding the irregular physical domains, are shown on the left; and the output functions $p \circ \phi_a$ representing the pressure field on the irregular physical domains, are depicted on the right.

in $\mathcal{D}$ to the C-grid in $\mathcal{D}_a$. Subsequently, we formulate the operator learning problem in the latent space as

$$\Psi^\dagger_{\mathrm{F2F}} \colon \phi_a \to p \circ \phi_a \,. \tag{5.5.9}$$

In this equation, the deformation map ϕ_a is a function defined in $\mathcal{D}$, and $p \circ \phi_a$ represents the pressure function defined in $\mathcal{D}$. As mentioned previously, both lift and drag depend solely on the pressure distribution over the airfoil. Hence, we can alternatively formulate the learning problem in the 1D setting by focusing solely on learning the pressure distribution over the airfoil. We construct a one-dimensional latent space $\mathcal{D} = [0, 1]$ and also denote the deformation map as $\phi_a \colon \xi \to x(\xi)$ mapping from $\mathcal{D}$ to the shape of the airfoil. The corresponding operator learning problem in this 1D setting has the same form as (5.5.9). The ground truth maps $\Psi^\dagger_{\mathrm{F2V}}$, $\Psi^\dagger_{\mathrm{V2F}}$, and $\Psi^\dagger_{\mathrm{V2V}}$ are defined similarly, mapping either the deformation function ϕ_a or the 7-dimensional control node vector input to the pressure function or the QoI (5.5.8) itself. We use all four variants of the FNM architectures and a finite-dimensional NN to approximate these maps from data.

For each sample size N, five i.i.d. realizations of the models are trained on the relative loss (5.5.1) for 2000 epochs in L^2 output space norm for functions and Euclidean norm for vectors. All FNM models use 4 hidden layers, 12 modes per dimension, and a channel width of 128. We compare these models to a

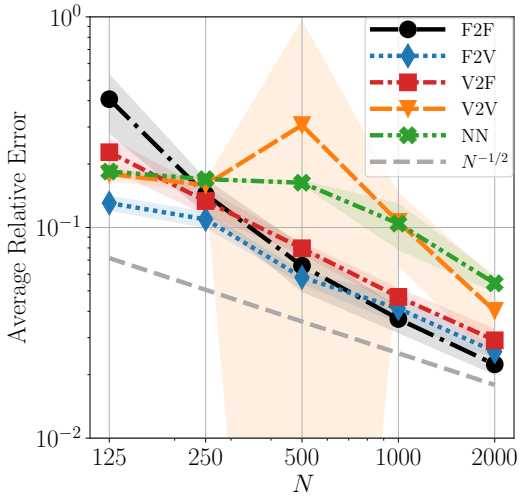


Figure 5.7: Flow over an airfoil. Comparative analysis of relative test error versus data size for the FNM and NN approaches. The shaded regions denote two standard deviations away from the mean of the test error over five realizations of the batch indices during SGD and model parameter initializations.

standard fully-connected NN with four layers and a hidden width of 128. In the case of FNM models, we observe that learning in the 1D setting consistently outperforms the 2D setting across all sizes of training data. Therefore, we only present results for the 1D setting. Moreover, this set of architectural hyperparameters with a large channel width of 128 in general outperforms other hyperparameter settings.

Figure 5.7 contains the results and reveals several trends. As the data volume N increases, all error curves decay at an algebraic rate that is slightly faster than $N^{-1/2}$. This may be due to the small sample sizes considered (under 2000 data pairs) or, especially since the training data is noise-free, could be evidence of a data-driven "superconvergence" effect similar to that observed for QoI computations in adjoint methods for PDEs [155]. Overall, emulating PtO maps by training models with finite-dimensional vectors as both input and output (V2V and NN) is more challenging for this problem than adopting function space variants (F2F, F2V, V2F). The standard finite-dimensional NN performs similarly to V2V.

Effective tensor for a multiscale elliptic equation

This example considers an equation that arises in elasticity in computational solid mechanics and relates the material properties on small scales to the ef-

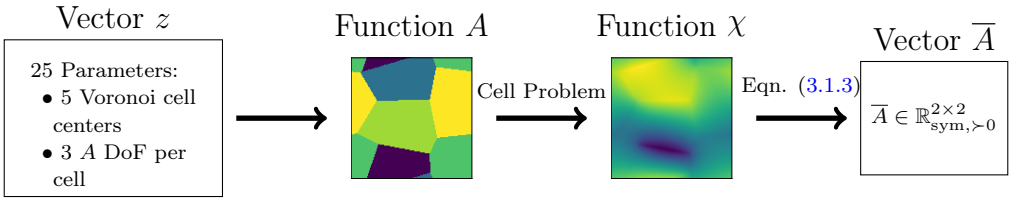


Figure 5.8: Diagram showing the homogenization experiment ground truth maps. The function A is parametrized by a finite vector z. The quantity of interest $\overline{A}$ (3.1.3) is computed from both the material function A and the solution χ to the cell problem (3.1.4). Note that both A and χ are functions on the torus $\mathbb{T}^2$.

fective property on a larger scale. Formally, we consider the following linear multiscale elliptic equation on a bounded domain $\mathcal{D} \subset \mathbb{R}^2$:

$$\begin{aligned} -\nabla \cdot (A^\epsilon \nabla u^\epsilon) &= g \quad \text{in} \quad \mathcal{D}\,, \\ u^\epsilon &= 0 \quad \text{on} \quad \partial\mathcal{D}\,. \end{aligned}$$

Here A^ϵ is given by $x \mapsto A^\epsilon(x) = A\left(\frac{x}{\epsilon}\right)$ for some $A\colon \mathbb{T}^2 \to \mathbb{R}^{2\times 2}_{\mathrm{sym},\succ 0}$ which is 1-periodic and positive definite. The source term is g. This equation contains fine-scale dependence through A^ϵ, which may be computationally expensive to evaluate without taking advantage of periodicity. The method of homogenization allows for elimination of the small scales in this manner and yields the homogenized equation

$$\begin{aligned} -\nabla \cdot \left(\overline{A} \nabla u\right) &= g \quad \text{in} \quad \mathcal{D}\,, \\ u &= 0 \quad \text{on} \quad \partial\mathcal{D}\,, \end{aligned}$$

where $\overline{A}$ is given by

$$\overline{A} = \int_{\mathbb{T}^2} \left(A(y) + A(y)\nabla\chi(y)^\top\right) \mathsf{d}y$$

and $\chi\colon \mathbb{T}^2 \to \mathbb{R}^2$ solves the cell problem

$$\begin{aligned} -\nabla \cdot \left((\nabla\chi)A\right) &= \nabla \cdot A \quad \text{in} \quad \mathbb{T}^2\,, \\ \int_{\mathbb{T}^2} \chi(y)\, \mathsf{d}y &= 0 \quad \text{and} \quad \chi \text{ is 1-periodic}\,. \end{aligned}$$

For $0 < \epsilon \ll 1$, the solution u^ϵ of (3.1.1) is approximated by the solution u of (3.1.2). The error between the solutions converges to zero as $\epsilon \to 0$ [26, 27].

The bottleneck step in obtaining the effective tensor $\overline{A}$, which is our QoI, is solving the cell problem (3.1.4). Learning the solution map $A \mapsto \chi$ in (3.1.4)

corresponds to the F2F setting and is explored in detail in Chapter 3. Alternately, one could learn the effective tensor $\overline{A}$ directly using the F2V-FNM architecture to approximate $A \mapsto \overline{A}$. Furthermore, though A is a function from $\mathbb{T}^2$ to $\mathbb{R}^{2\times 2}_{\text{sym},\succ 0}$, in certain cases it may have an exact finite vector parametrization. One example of this case is finite piecewise-constant Voronoi tessellations; A takes constant values on a fixed number of cells, and the cell centers uniquely determine the Voronoi geometry. Denoting these parameters as $z \in \mathbb{R}^{d_u}$ for appropriate $d_u \in \mathbb{N}$, one could also learn the V2F map $z \to \chi$ or the V2V map $z \to \overline{A}$. In this experiment, we compare the error in the QoI $\overline{A}$ using all four methods. A visualization of the possible maps is shown in Figure 5.8. Since our example is in two spatial dimensions, the five Voronoi cell centers have two components each. The symmetry of A yields three degrees of freedom (DoF) on each Voronoi cell. Altogether, this yields 25 parameters that comprise the finite vector input.

For training, we use the absolute squared loss in (5.5.2) with the H^1 norm for function output and Frobenius norm for vector output. Test error is also evaluated using this metric. Data are generated with a finite element solver using the method described in [56]; both A and χ are interpolated to a 128×128 grid, and the Voronoi geometry is randomly generated for each sample. The test set size is 500. Each map uses hyperparameters obtained via a grid search. For F2F, F2V, V2F, and V2V, the number of modes are 18, 12, 12, and 18, and the channel widths are 64, 96, 96, and 64, respectively. The fully-connected NN used as a comparison has a channel width of 576 and 2 hidden layers. As a consequence, all methods have a fixed model size of modes times width equaling 1152.

The results for the homogenization experiment in Figure 5.9 reinforce the theoretical intuition from Section 5.4 that learning with vector data results in higher error than learning with function data. Both the F2F and the F2V models approximately track the $N^{-1/2}$ rate, where N is the number of training data. On the other hand, the V2V model and NN model fail to attain this rate and saturate at the same level of roughly 10% error. The V2F map does achieve a slightly faster error decay rate than the V2V architecture for large enough sample sizes N, but it does not approach the $N^{-1/2}$ rate obtained by the F2F and F2V models. These rate differences occur when there is a difference in input dimension. On the other hand, for a difference in output dimension,

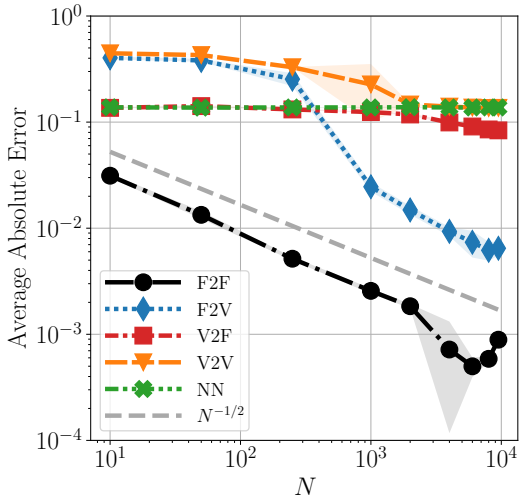


Figure 5.9: Elliptic homogenization problem. Absolute $\overline{A}$ error in the Frobenius norm versus data size for the FNM and NN architectures. The shaded regions denote two standard deviations away from the mean of the test error over five realizations of batch indices during SGD and model parameter initializations.

while both the F2F and F2V models reach roughly the same convergence rate, the F2V error remains an order of magnitude higher than the F2F error. We note that when measuring performance with relative test error instead, the qualitative behavior of Figure 5.9 remains the same.

5.6 Conclusion

This chapter proposes the Fourier Neural Mappings (FNMs) framework as an operator learning method for approximating parameter-to-observable (PtO) maps with finite-dimensional vector inputs or outputs, or both. Universal approximation theorems demonstrate that FNMs are well-suited for this task. Of central interest is the setting in which the PtO map factorizes into a vector-valued quantity of interest (QoI) map composed with a forward operator mapping between two function spaces. For this setting, the work introduces the end-to-end and full-field learning approaches. The end-to-end approach directly estimates the PtO map from its own input–output pairs, while the full-field approach estimates the forward map first and then plugs this estimator into the QoI. The chapter implements the FNM architectures for three nonlinear problems arising from environmental science, aerodynamics, and materials modeling. The numerical results support the linear theory and extend beyond

it by revealing the supremacy of function space representations of the input space over analogous finite-dimensional vector parametrizations.

Chapter 6

THEORY-TO-PRACTICE GAP IN OPERATOR LEARNING

This chapter is adapted from the following preprint:

[1] Philipp Grohs, Samuel Lanthaler, and Margaret Trautner. *Theory to Practice Gap for Neural Networks and Neural Operators*. 2025. arXiv: 2503.18219 [cs.LG].

This work studies the sampling complexity of learning with ReLU neural networks and neural operators. For mappings belonging to relevant approximation spaces, we derive upper bounds on the best-possible convergence rate of any learning algorithm, with respect to the number of samples. In the finite-dimensional case, these bounds imply a gap between the parametric and sampling complexities of learning, known as the *theory-to-practice gap*. In this work, a unified treatment of the theory-to-practice gap is achieved in a general L^p-setting, while at the same time improving available bounds in the literature. Furthermore, based on these results the theory-to-practice gap is extended to the infinite-dimensional setting of operator learning. Our results apply to Deep Operator Networks and integral kernel-based neural operators, including the Fourier neural operator. We show that the best-possible convergence rate in a Bochner L^p-norm is bounded by Monte-Carlo rates of order $1/p$.

6.1 Introduction

Deep learning has had remarkable success in a wide range of tasks such as speech recognition, computer vision, or natural language processing [156]. Increasingly this methodology is also used in the scientific domain, with applications to protein folding [157], plasma physics, [158] and numerical weather prediction [116, 159, 160]. However, the theoretical underpinnings of this field remain incomplete, and significant advances are still required to understand the empirical success of neural networks in these diverse applications. In many of these tasks, the goal is to approximate an unknown mapping based on a dataset consisting of input- and output-pairs, so-called *supervised learning*.

This has motivated a surge of theoretical work aimed at deepening our understanding of supervised learning with neural networks.

Theoretical error estimates for supervised learning are usually obtained by splitting the overall error into two contributions [161]: the approximation error and the generalization error. The approximation error measures the best-possible error that can be achieved by a given architecture. The generalization error bounds the difference between this best-possible error and the error achieved by empirical risk minimization, i.e. optimization based on a finite number of empirical data samples. This decomposition captures a trade-off between *model expressivity* (or parametric complexity), and generalization from a finite amount of data, i.e. the *sampling complexity*.

The study of the expressivity of neural networks dates back several decades to foundational work such as that of Cybenko [4] and Hornik, Stinchcombe and White [162], which focused on qualitative universality theorems. Motivated by the need to better explain the empirical efficiency of deep neural networks in applications, there has recently been increased interest in deriving quantitative approximation guarantees. These relate achievable approximation errors to key factors, such as the depth, width or choice of activation function [163, 164, 165, 166, 22]. What is striking about these results is that the derived approximation rates are generally far superior to the rates that are achievable by classical numerical representations. This fact makes deep learning methods potentially appealing for applications in numerical analysis where a high convergence rate is often desired.

The Theory-to-Practice Gap. Despite these encouraging results, any actual numerical approximation algorithm must operate with limited information on the function to be approximated — typically in the form of a finite number of samples of the function itself, or samples of a local (differential) operator applied to the function. It is therefore a key question whether the theoretically established approximation rates can be retained under such limited information.

For this reason, the sampling complexity of deep learning has also been of recent focus. Of particular relevance to the present work are the articles [22, 167, 168]. An informal summary of their results is as follows. First, define by $U^\alpha([0,1]^d)$ the set of functions on $[0,1]^d$ which, for any $n \in \mathbb{N}$, can be

approximated by a neural network ψ_n with at most n non-zero weights, and with approximation error $\|f - \psi_n\|_{L^\infty} \leq n^{-\alpha}$. We refer to the approximation rate α as the *parametric convergence rate*, since this rate holds with respect to the number of neural network parameters n.

Second, to address the sampling complexity of computing such an approximation, [22] considers (optimal) reconstruction methods aiming to reconstruct $f \in U^\alpha([0,1]^d)$ from point samples $f(x_1), \ldots, f(x_N)$, and examines limits on the best possible guaranteed convergence rate — *not in terms of the number of parameters but in the number of samples* N.

More formally, [22] asks for the optimal convergence rate $\beta_* > 0$ for which there exists a reconstruction method $A : f \mapsto Q(f(x_1), \ldots, f(x_N))$ defined in terms of a reconstruction mapping $Q : \mathbb{R}^N \to L^p([0,1]^d)$, with a convergence guarantee of the form $\sup_{f \in U^\alpha} \|f - A(f)\|_{L^p} \leq CN^{-\beta_*}$. Compared to the parametric convergence rate α, which describes the optimal convergence rate in terms of the number of parameters, the rate β_* is now in terms of the available information — the N function samples. This is of practical relevance, since a single function evaluation requires a certain minimal computational time, any upper bound on β_* yields a corresponding lower bound on the time to compute an accurate approximation for $f \in U^\alpha$.

A naive counting argument based on the number of degrees of freedom would suggest that $\beta_* = \alpha$ coincides with the optimal parametric convergence rate α. Indeed, it might be hoped that the determination of n parameters of the approximating neural network ψ_n requires $N = n$ point evaluations, implying that for $A(f) := \psi_n$, we have $\|f - A(f)\|_{L^\infty} \leq n^{-\alpha} = N^{-\alpha}$. As is well-known, this intuition is indeed correct for reconstruction by several popular methods, including polynomials, trigonometric polynomials, and certain kernel methods [169]. However, the results of [22] show that this intuition does not carry over to the sampling complexity of neural network approximation spaces: in this case, there is a gap between β_* and α. In fact, even in the limit $\alpha \to \infty$, the optimal sampling convergence rate β_* remains uniformly bounded and, for $p = \infty$ it even holds that $\beta_* \lesssim \frac{1}{d}$ which implies the existence of a curse of dimension. The gap between α and β_* is termed the *theory-to-practice gap* as it describes the discrepancy between the complexity of a theoretically possible approximation and one that can actually be computed from the information at hand.

A first contribution of this work is Theorem 6.2.2 which further extends and sharpens the bounds from [22] in the setting of finite-dimensional function approximation. Roughly speaking, we show that in a general L^p-setting, $\beta_* \leq \frac{1}{p} + \frac{1}{d}$ which means that for high input dimensions (i.e., large d), no actual algorithm is capable of beating the standard Monte-Carlo rate $\frac{1}{p}$ – irrespective of the parametric convergence rate α. Compared to results in [22] which, roughly speaking, established bounds of the form $\beta_* \leq \frac{1}{p} + 1$, this constitutes a significant improvement if the input dimension d is large.

Operator Learning. In addition to the aforementioned works on neural networks in finite-dimensions, there has also been increasing interest in operator learning [170]. The aim of operator learning is to approximate non-linear operators $\mathcal{G} : \mathcal{X} \to \mathcal{Y}$, mapping between infinite-dimensional function spaces $\mathcal{X}$ and $\mathcal{Y}$. In applications, such operators often arise as solution operators associated with a partial differential equation, but more general classes of operators can be considered. To approximate such $\mathcal{G}$, operator learning frameworks generalize neural networks to this infinite-dimensional setting. Empirically, the most successful approach is usually based on supervised learning from training data, and hence the same questions as discussed above also arise in this context.

Early work on operator learning dates back to a foundational paper by Chen and Chen [118]. Without any claim of completeness, we mention a number of works studying the parametric complexity of operator learning which aim to relate the model size to the achieved approximation accuracy, including both upper bounds on the required number of parameters, e.g. [102, 171, 172, 173], as well as lower bounds, e.g. [174, 175, 176, 177]. The data (or sampling) complexity of operator learning has been studied in [178, 133, 179]. In connection with the present work, we also highlight [180], where approximation spaces for the Fourier neural operator are introduced, and upper bounds on the sampling complexity of learning on these spaces are discussed. Closely related spaces will be discussed in this work. These spaces, which are an infinite-dimensional generalization of the approximation spaces U^α introduced in [22] (and described above), represent classes of Deep Operator Networks and kernel-integral based neural operators with a parametric approximation rate α. They will be introduced and studied in Sections 6.3 and 6.3.

In the context we are able to generalize the theory-to-practice gap to the infinite-dimensional setting: assuming access to an optimal algorithm for the reconstruction of the underlying mapping from N samples, we show that the best achievable convergence rate $N^{-\beta_*}$ on the relevant approximation spaces is upper bounded by $1/p$, if the reconstruction error is measured in a Bochner L^p-norm. These results are provided in Theorems 6.3.12 and 6.3.14 (Deep Operator Networks) and Theorems 6.3.20 and 6.3.22 (kernel-integral based neural operators) and they uncover fundamental limits on the convergence guarantees that are possible in the context of operator learning: one cannot improve on the standard Monte-Carlo rate, irrespective of how high the parametric convergence rate may be.

In summary, this work makes the following contributions:

- We give a unified treatment of the theory-to-practice gap in a general L^p-setting, valid for arbitrary $p \in [1, \infty]$, sharpening available bounds in the literature.
- We extend the theory-to-practice gap to the infinite-dimensional setting of operator learning. Our results apply to Deep Operator Networks and integral kernel-based neural operators, including the Fourier neural operator.
- For these architectures, we show that the optimal convergence rate in a Bochner L^p-norm is bounded by $\beta_* \leq 1/p$, for any $p \in [1, \infty)$.
- We furthermore show that $\beta_* = 0$ for *uniform* approximation over a compact set of input functions, i.e. no algebraic convergence rates are possible.

Overview

In Section 6.2 we first review neural network approximation spaces and relevant notions from sampling complexity theory. In Section 6.2, we then state and prove our main result in the finite dimensional setting (Theorem 6.2.2.) In Section 6.3, we extend the finite-dimensional result to operator learning. After a brief review of relevant concepts, we discuss the approximation-theoretic setting in Section 6.3. In Section 6.3 we state an abstract result, Proposition 6.3.6, which establishes a connection between the finite- and infinite-dimensional settings. Finally, based on this abstract result, we derive an infinite-dimensional

theory-to-practice gap for approximation in L^p- and sup-norms. Section 6.3 discusses Deep Operator Networks, resulting in Theorems 6.3.12 and 6.3.14, respectively. Section 6.3 discusses kernel-integral based neural operators, resulting in Theorems 6.3.20 and 6.3.22. We end with conclusions and further discussion in Section 6.4.

Notation

For a vector $v \in \mathbb{R}^d$, we indicate by $|v|$ the Euclidean norm, and for a matrix $A \in \mathbb{R}^{m\times d}$, we denote by $\|A\| = \sup_{|v|=1} |Av|$ its operator norm. Given a domain $D \subset \mathbb{R}^d$, we denote by $W^{1,\infty}(D;\mathbb{R}^m)$ the set of measurable functions $u : D \to \mathbb{R}^m$ with uniformly bounded values and uniformly bounded weak derivatives (Lebesgue almost everywhere). The corresponding norm is defined as

$$\|u\|_{W^{1,\infty}(D;\mathbb{R}^m)} = \|u\|_{L^\infty(D;\mathbb{R}^m)} + |u|_{W^{1,\infty}(D;\mathbb{R}^m)},$$

where $|u|_{W^{1,\infty}(D;\mathbb{R}^m)} = \operatorname{ess\,sup}_{x\in\Omega} \|Du(x)\|$ denotes the $W^{1,\infty}$ seminorm. Given a topological space $\mathcal{X}$, we will denote by $\mathcal{P}(\mathcal{X})$ the set of all probability measures on $\mathcal{X}$ under the Borel σ-algebra. For two measures μ and ν, on a measure space (Ω, Σ), we will write $\mu \geq \nu$ to indicate that $\mu(B) \geq \nu(B)$ for any element $B \in \Sigma$.

6.2 A generalized gap in finite dimension

The goal of this section is to derive new lower bounds on the sampling complexity of ReLU neural network approximation spaces, in a finite-dimensional setting. To describe the mathematical setting, we recall relevant notions from [22, Sect. 2.1 and 2.2], below.

ReLU neural networks

Following [22], a **neural network** is defined as a tuple $\psi = ((A_1, b_1), \ldots, (A_L, b_L))$, consisting of matrices $A_j \in \mathbb{R}^{d_j\times d_{j-1}}$ and bias vectors $b_j \in \mathbb{R}^{d_j}$. The number of layers $L(\psi) := L$ is the **depth** of ψ, and $W(\psi) := \sum_{j=1}^{L}(\|A_j\|_{\ell^0} + \|b_j\|_{\ell^0})$ is used to denote the **number of (nonzero) weights** of ψ. Here, the notation $\|A\|_{\ell^0}$ refers to the number of nonzero entries of a matrix (or vector) A. Finally, we write $d_{\text{in}}(\psi) := d_0$ and $d_{\text{out}}(\psi) := d_L$ for the **input and output dimension** of ψ, and we set $\|\psi\|_{\mathsf{NN}} := \max_{j=1,\ldots,L} \max\{\|A_j\|_\infty, \|b_j\|_\infty\}$, where $\|A\|_\infty := \max_{i,j} |A_{i,j}|$.

The function $R_\sigma\psi$ computed by ψ, is defined via an activation function σ. In this work, we restrict attention to ReLU networks, corresponding to $\sigma : \mathbb{R} \to \mathbb{R},\ x \mapsto \max\{0, x\}$, and acting componentwise on vectors. The **realization** $R_\sigma\psi : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ is then given by

$$R_\sigma\psi := T_L \circ (\sigma \circ T_{L-1}) \circ \cdots \circ (\sigma \circ T_1) \quad \text{where} \quad T_j : \begin{cases} \mathbb{R}^{d_{j-1}} & \to & \mathbb{R}^{d_j} \\ x & \mapsto & A_j x + b_j \end{cases}.$$

With slight abuse of notation, we usually do not distinguish notationally between ψ and its ReLU-realization $R_\sigma\psi$. Thus, we simply say that $\psi : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ is a (ReLU) neural network, when referring to the realization associated with a specific setting of the weights matrices and biases $\psi = ((A_1, b_1), \ldots, (A_L, b_L))$.

Neural network approximation spaces

We next summarize the relevant approximation spaces $A_\ell^{\alpha,\infty}(D)$ for a domain $D \subset \mathbb{R}^d$; they depend on a generalized 'smoothness' parameter $\alpha > 0$ and a depth-growth function $\ell = \ell(n)$. In short, $A_\ell^{\alpha,\infty}(D)$ contains all functions $f : D \to \mathbb{R}$ that can be uniformly approximated at rate $n^{-\alpha}$, by ReLU neural networks with at most n non-zero weights and biases, depth at most $\ell(n)$, and weight magnitude at most 1.

More precisely, given an input dimension $d \in \mathbb{N}$ and a non-decreasing depth-growth function $\ell : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$, we define

$$\Sigma_{d,n}^\ell := \left\{ \psi : \mathbb{R}^d \to \mathbb{R} \;\middle|\; \begin{array}{l} \psi \text{ NN with } d_{\text{in}}(\psi) = d, d_{\text{out}}(\psi) = 1, \\ W(\psi) \le n, L(\psi) \le \ell(n), \|\psi\|_{\mathsf{NN}} \le 1 \end{array} \right\}.$$

Then, given a measurable subset $D \subset \mathbb{R}^d$, $p \in [1, \infty]$, and $\alpha \in (0, \infty)$, for each measurable $f : D \to \mathbb{R}$, we define

$$\Gamma^{\alpha,p}(f) := \max\left\{ \|f\|_{L^p(D)},\ \sup_{n\in\mathbb{N}} \left[n^\alpha \cdot d_{p,D}\big(f, \Sigma_{d,n}^\ell\big) \right] \right\} \in [0, \infty],$$

where $d_{p,D}(f, \Sigma) := \inf_{g\in\Sigma} \|f - g\|_{L^p(D)}$.

As pointed out in [22], $\Gamma^{\alpha,p}$ is not a (quasi-)norm. However, one can define a neural network **approximation space quasi-norm** $\|\cdot\|_{A_\ell^{\alpha,p}}$ by

$$\|f\|_{A_\ell^{\alpha,p}} := \inf\{\theta > 0 : \Gamma^{\alpha,p}(f/\theta) \le 1\} \in [0, \infty],$$

giving rise to the **approximation space**

$$A_\ell^{\alpha,p} := A_\ell^{\alpha,p}(D) := \left\{ f \in L^p(D) : \|f\|_{A_\ell^{\alpha,p}} < \infty \right\}.$$

We denote by $U_\ell^{\alpha,p}(D) \subset A_\ell^{\alpha,p}(D)$ the unit ball,

$$U_\ell^{\alpha,p} := U_\ell^{\alpha,p}(D) := \Big\{ f \in L^p(D) : \|f\|_{A_\ell^{\alpha,p}} \leq 1 \Big\}.$$

As shown in [22], $U_\ell^{\alpha,p}(D)$ consists precisely of those f for which $d_{p,D}(f, \Sigma_{d,n}^\ell) \leq n^{-\alpha}$ for all $n \in \mathbb{N}$. We will focus on the special case $p = \infty$, in the following.

The following quantity is crucial in characterizing the sampling complexity of neural network approximation spaces [22]:

$$\ell^* := \sup_{n \in \mathbb{N}} \ell(n) \in \mathbb{N} \cup \{\infty\}. \tag{6.2.1}$$

It describes the maximal depth that is allowed for a neural network approximant of a given function.

***Remark* 6.2.1.** We will later extend the definition of the approximation spaces $A_\ell^{\alpha,\infty}$ to the infinite-dimensional operator learning setting. Specifically, in Section 6.3 we define $\mathbf{A}_{\ell,\mathrm{DON}}^{\alpha,\infty}$ for a class of DeepONets and in Section 6.3, we define $\mathbf{A}_{\ell,\mathrm{NO}}^{\alpha,\infty}$ for (integral-kernel based) Neural Operators. ◊

Sampling complexity

By definition, the approximation of a function $f \in A_\ell^{\alpha,\infty}(D)$ by neural networks can be achieved at a rate $n^{-\alpha}$, in terms of the required number of neural network parameters. This rate is fast, when $\alpha \gg 1$, thus allowing for efficient approximation in principle. Despite this fact, it has been shown in [22] that such high rates, in terms of the parameter count n, do not imply correspondingly high convergence rates when considering the required number of samples to train such neural networks. This phenomenon is referred to as the *theory-to-practice gap*.

Below, we recall mathematical notions from [22], which quantify the sampling complexity of a set of continuous functions $U \subset C(D)$, where $D \subset \mathbb{R}^d$ is a subdomain. We are mostly interested in the setting where $U = U_\ell^{\alpha,\infty}(D)$ is the unit ball in the relevant neural network approximation space.

The Deterministic Setting

Given $U \subset C(D)$, we now consider the approximation of $f \in U$ with respect to the $L^p(D)$-norm.

A map $A : U \to L^p(D)$ is called a *deterministic method using* $N \in \mathbb{N}$ *point measurements*, if there exists $\mathbf{x} = (x_1, \dots, x_N) \in D^N$ and a map $Q : \mathbb{R}^N \to L^p(D)$ such that

$$A(f) = Q(f(x_1), \dots, f(x_N)) \quad \forall\, f \in U.$$

The set of all deterministic methods using N point measurements will be denoted by $\mathrm{Alg}_N(U, L^p(D))$.

We define the error of A for approximation in L^p as

$$e(A, U, L^p(D)) := \sup_{f \in U} \|f - A(f)\|_{L^p(D)}.$$

The **optimal error** for (deterministic) approximation in L^p using N point samples is then

$$e_N^{\mathrm{det}}(U, L^p(D)) := \inf_{A \in \mathrm{Alg}_N(U, L^p(D))} e(A, U, L^p(D)).$$

Finally, the **optimal order of convergence** for (deterministic) approximation in L^p using N point samples is

$$\beta_*^{\mathrm{det}}(U, L^p(D)) := \sup\left\{ \beta \geq 0 : \begin{array}{l} \exists\, C > 0 \,\text{s.t.}\ \forall N \in \mathbb{N}, \\ e_N^{\mathrm{det}}(U, L^p(D)) \leq C \cdot N^{-\beta} \end{array} \right\}. \tag{6.2.2}$$

The Randomized Setting

Generalizing the deterministic setting above, we consider randomized algorithms. A *randomized method using* $N \in \mathbb{N}$ *point measurements (in expectation)* is a tuple (A, N) consisting of a family $A = (A_\omega)_{\omega \in \Omega}$ of maps $A_\omega : U \to L^p(D)$ indexed by a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a measurable function $\mathsf{N} : \Omega \to \mathbb{N}$ with the following properties:

1. for each $f \in U$, the map $\Omega \to L^p(D), \omega \mapsto A_\omega(f)$ is measurable with respect to the Borel σ-algebra on D,
2. for each $\omega \in \Omega$, we have $A_\omega \in \mathrm{Alg}_{\mathsf{N}(\omega)}(U, L^p(D))$,
3. $\mathbb{E}_\omega[\mathsf{N}(\omega)] \leq N$.

We say that (A, N) is *strongly measurable* if the map $\Omega \times U \to L^p(D), (\omega, f) \mapsto A_\omega(f)$ is measurable, where $U \subset C(D)$ is equipped with the Borel σ-algebra induced by the uniform norm. We denote the set of all strongly measurable (A, N) satisfying the above properties by $\mathrm{Alg}_N^*(U, L^p(D))$.

The **expected error** of a randomized algorithm (A, N) for approximation in L^p is defined as

$$e((A, \mathsf{N}), U, L^p(D)) := \sup_{f \in U} \mathbb{E}_\omega \big[\|f - A_\omega(f)\|_{L^p(D)} \big]. \tag{6.2.3}$$

The **optimal randomized error** for approximation in L^p using N point samples (in expectation) is

$$e_N^{\mathrm{ran}}(U, L^p(D)) := \inf_{(A,\mathsf{N}) \in \mathrm{Alg}_N^*(U, L^p(D))} e((A, \mathsf{N}), U, L^p(D)). \tag{6.2.4}$$

Finally, the **optimal randomized order** for approximation in L^p using point samples is

$$\beta_*(U, L^p(D)) := \sup \left\{ \beta \geq 0 : \begin{array}{l} \exists\, C > 0 \text{ s.t. } \forall N \in \mathbb{N}, \\ e_N^{\mathrm{ran}}(U, L^p(D)) \leq C \cdot N^{-\beta} \end{array} \right\}. \tag{6.2.5}$$

We point out that a deterministic method is a special case of a randomized method, and hence

$$\beta_*^{\mathrm{det}}(U, L^p(D)) \leq \beta_*(U, L^p(D)). \tag{6.2.6}$$

Since our aim is to derive upper bounds on these convergence rates, we may restrict attention to $\beta_*(U, L^p(D))$, implying corresponding bounds also for $\beta_*^{\mathrm{det}}(U, L^p(D))$.

In the following, we in particular derive upper bounds for the exponents $\beta_*(U_\ell^{\alpha,\infty}, L^p)$, where $U_\ell^{\alpha,\infty} = U_\ell^{\alpha,\infty}([0,1]^d)$ and $L^p = L^p([0,1]^d)$ with $1 \leq p \leq \infty$.

Theory-to-practice gap

It was shown in [22, Theorem 1.1 and 1.2], that the best possible convergence rate of any reconstruction method on $U_\ell^{\alpha,\infty} := U_\ell^{\alpha,\infty}([0,1]^d)$ based on point samples is upper bounded as follows. When the reconstruction error is measured with respect to the $L^\infty([0,1]^d)$ norm, the rate is upper bounded by

$$\beta_*(U_\ell^{\alpha,\infty}, L^\infty) \leq \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor}. \tag{6.2.7}$$

When measuring the reconstruction error with respect to the $L^2([0,1]^d)$ norm, the bound becomes

$$\beta_*(U_\ell^{\alpha,\infty}, L^2) \leq \frac{1}{2} + \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor}. \tag{6.2.8}$$

These are the tightest known upper bounds on the convergence rates for the most relevant range $\alpha \geq \max(2, \lfloor \ell^*/2 \rfloor)$; for smaller values of α refined estimates are given in [22, Theorems 5.1 and 7.1]. Astonishingly, even in the limit $\alpha \to \infty$, the best possible reconstruction rates are upper bounded by $1/d$ and $3/2$, respectively.

Our first goal in the next subsection is to extend and sharpen the bounds in (6.2.7) and (6.2.8) for arbitrary $d \in \mathbb{N}$ and $p \in [1, \infty]$, with a unified proof.

Main result in finite dimension We prove the following theorem:

***Theorem* 6.2.2.** Let $p \in [1, \infty]$. Let $\ell : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ be non-decreasing with $\ell^* \geq 3$. Given $d \in \mathbb{N}$ and $\alpha \in (0, \infty)$, consider

$$U := U_\ell^{\alpha,\infty}(\mathbb{R}^d)|_{[0,1]^d} := \left\{ f|_{[0,1]^d} \,\middle|\, f \in U_\ell^{\alpha,\infty}(\mathbb{R}^d) \right\},$$

such that $U \subset C([0,1]^d)$. Then

$$\beta_*(U, L^p([0,1]^d)) \leq \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor}. \tag{6.2.9}$$

◊

Since the restriction $U = U_\ell^{\alpha,\infty}(\mathbb{R}^d)|_{[0,1]^d}$ is a subset of $U_\ell^{\alpha,\infty}([0,1]^d)$, Theorem 6.2.2 implies the following corollary:

***Corollary* 6.2.3.** Denote $U_\ell^{\alpha,\infty} = U_\ell^{\alpha,\infty}([0,1]^d)$. Under the assumptions of Theorem 6.2.2, we have

$$\beta_*(U_\ell^{\alpha,\infty}, L^p([0,1]^d)) \leq \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor}.$$

◊

***Remark* 6.2.4.** The upper bound (6.2.9) implies in particular, that the best possible convergence rate of any randomized or deterministic reconstruction method is upper bounded by $\beta \leq \frac{1}{p} + \frac{1}{d}$. In high-dimensional applications, this upper bound is approximately $\frac{1}{p}$, implying that algorithms cannot be expected to converge at substantially faster than Monte-Carlo rates. ◊

Proof of Theorem 6.2.2

Our proof of Theorem 6.2.2 is based on several technical lemmas and ideas from [22]. The main novelty here is a different approach to combining these ingredients: while the derivation in [22] relies on a linear combination of (many) hat functions and combinatorial arguments, our proof will be instead be based on a random placement of a single hat function, and a probabilistic argument.

Outline Before detailing the proof of Theorem 6.2.2, we first outline the general idea on the domain $[0,1]^d$. Our first observation is that, for any choice of evaluation points $x_1, \dots, x_N \in [0,1]^d$, there exists a void with inner diameter of order $N^{-1/d}$; more precisely, we show that, independently of the choice of $x_1, \dots, x_N$, for a randomly drawn $y \sim \mathrm{Unif}([0,1]^d)$, we have

$$\min_{j=1,\dots,N} |y - x_j| \geq \frac{1}{4} N^{-1/d}, \tag{6.2.10}$$

with probability at least $1/2$.

Given the inevitable presence of such a void, we are then tempted to place a function g with support inside this void. If we assume that $\|g\|_{L^\infty} \leq 1$, or indeed that g is the characteristic function of a cube in this void, then such g can have an L^p-norm as large as $\|g\|_{L^p} \sim N^{-1/p}$. Given only point values at $x_1, \dots, x_N$, such g will be indistinguishable from the zero-function. Thus, if A is a reconstruction method relying only on the point values at $x_1, \dots, x_N$, then $A(g) = A(0)$. It follows that $\|g\|_{L^p} \leq \|g - A(g)\|_{L^p} + \|0 - A(0)\|_{L^p}$ and hence at least one of $\|g - A(g)\|_{L^p}$ or $\|0 - A(0)\|_{L^p}$ must be on the order of magnitude of $\|g\|_{L^p([0,1]^d)} \sim N^{-1/p}$. As a consequence, the achievable convergence rate β of a reconstruction method on characteristic functions is fundamentally limited to $\beta \leq \frac{1}{p}$.

To link the above observation with reconstruction methods on neural network approximation spaces, we will recall (cp. Lemma 6.2.6, below) that ReLU neural networks can efficiently approximate certain localized functions $\vartheta_{M,y}$. These functions are locally supported inside a cube of side-length $r := 1/M$ with center $y \in [0,1]^d$. In our proof, the $\vartheta_{M,y}$ with $M \sim N^{1/d}$ will act as a replacement of the characteristic function g of the outline, above. The main difference with the characteristic function is that the gradient of $\vartheta_{M,y}$ cannot be arbitrarily large, if we constrain it to belong to the unit ball $U_\ell^{\alpha,\infty}$ in the approximation space. This limit on the gradient introduces an additional

correction in our upper bound, which finally will take the form $\beta_* \leq \frac{1}{p} +$ (correction depending on α, ℓ).

Details We now proceed with the detailed proof of Theorem 6.2.2. For any placement of evaluation points $x_1, \dots, x_N \in [0,1]^d$, we first show that a point $y \in [0,1]^d$ picked uniformly at random has a positive chance of sitting in a "void" with interior diameter $\sim N^{-1/d}$.

***Lemma* 6.2.5** (Existence of a void)**.** Let $d, N \in \mathbb{N}$. Let $x_1, \dots, x_N \in [0,1]^d$ be given. Consider $y \sim \mathrm{Unif}([0,1]^d)$ drawn uniformly at random. Then

$$\mathrm{Prob}_y\left[\min_{j=1,\dots,N} |y - x_j|_\infty > \frac{1}{4}N^{-1/d}\right] \geq \frac{1}{2}. \tag{6.2.11}$$

$\Diamond$

Proof. Let us fix $r > 0$ for the moment. Applying a union bound, we note that

$$\begin{aligned}\mathrm{Prob}_y\left[\min_{j=1,\dots,N} |y - x_j|_\infty \leq r\right] &= \mathrm{Prob}_y\left[y \in \textstyle\bigcup_{j=1}^N \big(x_j + [-r,r]^d\big)\right] \\ &\leq \sum_{j=1}^N \mathrm{Vol}\big(x_j + [-r,r]^d\big) = N(2r)^d.\end{aligned}$$

It thus follows that, if $r := \frac{1}{4}N^{-1/d} \leq \frac{1}{2}(2N)^{-1/d}$, then

$$\mathrm{Prob}_y\left[\min_{j=1,\dots,N} |y - x_j| \leq r\right] \leq \frac{1}{2}.$$

Hence, we must have $\mathrm{Prob}_y[\min_{j=1,\dots,N} |y - x_j| > r] \geq \frac{1}{2}$, as claimed. □

We now state the following fundamental result, which follows from [22, Lemma 3.4]. The main insight of this lemma is that ReLU neural networks can efficiently approximate a localized function $\vartheta_{M,y} : \mathbb{R}^d \to [0,1]$, which is supported in a cube of side-length $2/M$ around y, and which "fills in" a significant fraction of this cube.

***Lemma* 6.2.6** (Localized networks)**.** Given $d \in \mathbb{N}$, $M \geq 1$ and $y \in [0,1]^d$, there exists a function $\vartheta_{M,y} : \mathbb{R}^d \to [0,1]$ with the following properties:

- $\vartheta_{M,y}(x)$ depends continuously on x and y; in fact, we have that $\vartheta_{M,y}(x) = \vartheta_M(x - y)$ is a shift of a neural network ϑ_M.

- $\vartheta_{M,y}(x) = 0$ whenever $|x - y|_\infty \geq 1/M$.
- For any $p \in [1, \infty]$, there exists $C = C(d, p) > 0$ satisfying,

$$\|\vartheta_{M,y}\|_{L^p([0,1]^d)} \geq CM^{-d/p},$$

- There exists a constant $\kappa = \kappa(\gamma, \alpha, d, \ell, c) > 0$, such that

$$g_{M,y} := \kappa M^{-\alpha/(\alpha+\lfloor \ell^*/2 \rfloor)} \vartheta_{M,y} \in U_\ell^{\alpha,\infty}(\mathbb{R}^d).$$

$\Diamond$

Proof. The existence of $\vartheta_{M,y}$ and the other properties are shown in [22], see in particular [22, Lemma 3.4]. $\square$

We also state the following result, which is a minor variant of [22, Lemma 2.3].

***Lemma* 6.2.7.** Let $D \subset \mathbb{R}^d$ and $\emptyset \neq U \subset C(D)$ be bounded, and let $1 \leq p \leq \infty$. Assume that there exists $\lambda \in [0, \infty)$, $\kappa > 0$, such that for every $N \in \mathbb{N}$, there exists a probability space $(\Xi, \mathbb{P})$ and a random variable $\psi : \Xi \to U$, $\xi \mapsto \psi_\xi$, such that

$$\mathbb{E}_\xi \big[\|\psi_\xi - A(\psi_\xi)\|_{L^p(D)} \big] \geq \kappa N^{-\lambda}, \quad \forall\, A \in \mathrm{Alg}_N^{\mathrm{det}}(U, L^p(D)).$$

Then $\beta_*(U, L^p(D)) \leq \lambda$. $\Diamond$

Proof. This follows by the same reasoning as [22, Lemma 2.3]. $\square$

Our main interest in this result is when $U = U_\ell^{\alpha,\infty}(\mathbb{R}^d)|_{[0,1]^d}$. Combining the results from Lemmas 6.2.5, 6.2.6, and 6.2.7, we now come to the proof of Theorem 6.2.2.

Proof of Theorem 6.2.2. Let $d, N \in \mathbb{N}$ be given. Recall that $U := U_\ell^{\alpha,\infty}(\mathbb{R}^d)|_{[0,1]^d}$. Our goal is to apply Lemma 6.2.7 to deterministic reconstruction methods based on N point-values,

$$A \in \mathrm{Alg}_N^{\mathrm{det}}(U, L^p([0,1]^d)).$$

To construct a suitable probability space $(\Xi, \mathbb{P})$ and random function $\Xi \to U$, we first define a probability space as $\Xi := [0,1]^d \times \{-1, +1\}$ endowed with

the Borel σ-algebra and define $\mathbb{P} := \mathrm{Unif}([0,1]^d) \otimes \mathrm{Unif}(\{-1,+1\})$. We will denote elements of Ξ as $\xi = (y,\sigma)$. This defines our probability space.

We next fix

$$M := 4N^{1/d}. \tag{6.2.12}$$

With this choice of M, we then define $\psi_\xi := \psi_{(y,\sigma)} := \sigma g_{M,y} \in U$, where $g_{M,y}$ is the function of Lemma 6.2.6. Thus, $\psi_\xi = \psi_{(y,\sigma)}$ is a random function, given by a random shift of a localized neural network ϑ_M by $y \sim \mathrm{Unif}([0,1]^d)$ and a random choice of sign $\sigma \sim \mathrm{Unif}(\{-1,+1\})$. This defines our random variable

$$\psi : \Xi \to U. \tag{6.2.13}$$

Invoking Lemma 6.2.7, it will suffice to prove the following claim, formulated as a lemma for later reference:

***Lemma* 6.2.8.** Let $\psi : \Xi \to U$, $\xi \mapsto \psi_\xi$ be the random variable defined above (6.2.13), where $U := U_\ell^{\alpha,\infty}(\mathbb{R}^d)|_{[0,1]^d}$. There exists a constant κ, independent of N, such that

$$\mathbb{E}_\xi\big[\|\psi_\xi - A(\psi_\xi)\|_{L^p([0,1]^d)}\big] \geq \kappa N^{-\lambda}, \tag{6.2.14}$$

for all $A \in \mathrm{Alg}^{\mathrm{det}}(U, L^p([0,1]^d))$ and where $\lambda := \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor}$. ◊

The claim of Theorem 6.2.2 is then immediate from Lemma 6.2.7 and 6.2.8.

Proof of Lemma 6.2.8. Let $A \in \mathrm{Alg}^{\mathrm{det}}(U_\ell^{\alpha,\infty}(\mathbb{R}^d), L^p([0,1]^d))$ be given. Let $Q : \mathbb{R}^N \to L^p([0,1]^d)$ be the reconstruction mapping associated with A and let $x_1, \dots, x_N \in \mathbb{R}^d$ denote the associated evaluation points, such that $A(f) = Q(f(x_1), \dots, f(x_N))$. We first observe that, whatever the choice of $x_1, \dots, x_N$, we have, by Lemma 6.2.5 and our choice of $M = 4N^{1/d}$ in (6.2.12),

$$\mathrm{Prob}_y\left[\min_{j=1,\dots,N} |y - x_j| > 1/M\right] \geq \frac{1}{2}.$$

Now consider the random event

$$\begin{aligned} E &= \big\{(y,\sigma) \in \Xi \,\big|\, \psi_{(y,\sigma)}(x_1) = \cdots = \psi_{(y,\sigma)}(x_N) = 0\big\} \\ &= \{(y,\sigma) \in \Xi \,|\, \sigma g_{M,y}(x_1) = \cdots = \sigma g_{M,y}(x_N) = 0\}, \end{aligned}$$

where the randomness is introduced by $y \sim \mathrm{Unif}([0,1]^d)$ and $\sigma \sim \mathrm{Unif}\{-1,+1\}$. Since $g_{M,y}$ is supported in the shifted cube $(y + [-1/M, 1/M]^d)$ (cp. Lemma 6.2.6), it follows that

$$\mathrm{Prob}_{y,\sigma}[E] \geq \mathrm{Prob}_y\left[\min_{j=1,\dots,N} |y - x_j| > 1/M\right] \geq \frac{1}{2}.$$

We can now finish the proof. To this end, we first observe that

$$\begin{aligned}
&\mathbb{E}_\xi\big[\|\psi_\xi - A(\psi_\xi)\|_{L^p([0,1]^d)}\big] \\
&\qquad \geq \mathbb{E}_\xi\big[\|\psi_\xi - A(\psi_\xi)\|_{L^p([0,1]^d)} \mid E\big]\,\mathrm{Prob}_\xi[E] \\
&\qquad \geq \frac{1}{2}\mathbb{E}_\xi\big[\|\psi_\xi - A(\psi_\xi)\|_{L^p([0,1]^d)} \mid E\big] \\
&\qquad = \frac{1}{2}\mathbb{E}_\xi\big[\|\psi_\xi - A(0)\|_{L^p([0,1]^d)} \mid E\big]. \qquad (6.2.15)
\end{aligned}$$

We next note that the random variable $\psi_\xi = \sigma g_{M,y}$, conditioned on E, has the same distribution as $-\psi_\xi = -\sigma g_{M,y}$ conditioned on E; this follows from the fact that E and $\mathbb{P}$ are invariant under replacement $\sigma \to -\sigma$. Furthermore, it follows from Lemma 6.2.6 that there exists a constant $C = C(\alpha, d, p, \ell) > 0$, such that

$$\|\psi_\xi\|_{L^p([0,1]^d)} = \|g_{M,y}\|_{L^p([0,1]^d)} \geq C M^{-d/p - \alpha/(\alpha + \lfloor \ell^*/2 \rfloor)}, \qquad (6.2.16)$$

for all $M \geq 1$ and $y \in [0,1]^d$. Hence, it follows with

$$\lambda := \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor},$$

that

$$\begin{aligned}
2CM^{-d\lambda} &\leq 2\mathbb{E}_\xi\big[\|\psi_\xi\|_{L^p([0,1]^d)} \mid E\big] && \text{(by (6.2.16))} \\
&= \mathbb{E}_\xi\big[\|\psi_\xi - (-\psi_\xi)\|_{L^p([0,1]^d)} \mid E\big] \\
&\leq \mathbb{E}_\xi\big[\|\psi_\xi - A(0)\|_{L^p([0,1]^d)} \mid E\big] \\
&\qquad + \mathbb{E}_\xi\big[\|-\psi_\xi - A(0)\|_{L^p([0,1]^d)} \mid E\big] && \text{(triangle ineq.)} \\
&= 2\mathbb{E}_\xi\big[\|\psi_\xi - A(0)\|_{L^p([0,1]^d)} \mid E\big] && \text{(invar. } \sigma \to -\sigma\text{)} \\
&\leq 4\mathbb{E}_\xi\big[\|\psi_\xi - A(\psi_\xi)\|_{L^p([0,1]^d)}\big]. && \text{(by (6.2.15))}
\end{aligned}$$

Thus, we have shown that

$$\mathbb{E}_\xi\big[\|\psi_\xi - A(\psi_\xi)\|_{L^p([0,1]^d)}\big] \geq \frac{C}{2} M^{-d\lambda}.$$

This implies a bound of the desired form (6.2.14), since

$$\frac{C}{2}M^{-d\lambda} = \frac{C}{2}\left(4N^{1/d}\right)^{-d\lambda} =: \kappa N^{-\lambda},$$

where we recall $\lambda = -\frac{1}{p} - \frac{1}{d}\cdot\frac{\alpha}{\alpha+\lfloor \ell^*/2\rfloor}$, and where the constant κ depends on α, d, p, ℓ, but is independent of N. This is (6.2.14) and concludes our proof. □

6.3 Extension to operator Learning

We now consider the extension of the theory-to-practice gap to operator learning, i.e. the data-driven approximation of operators mapping between infinite-dimensional function spaces. The finite-dimensional upper bound (6.2.9) suggests that in the infinite-dimensional limit, $d \to \infty$, the best convergence rate in L^p should be upper bounded by $\frac{1}{p}$, *independently of* α *and* $\lfloor \ell^*/2\rfloor$.

Our goal in this second half of the chapter is to rigorously state and prove such a theory-to-practice gap for two prototypical classes of neural operators: deep operator networks, as discussed in [23, 87], and a class of integral-kernel based neural operators [52, 181]. Before stating our main results in the operator-learning setting, we will first summarize the overall objective of operator learning, and discuss suitable infinite-dimensional replacements of the spaces $L^p([0,1]^d)$, for $1 \leq p \leq \infty$ in this context. This is followed by a definition of the aforementioned prototypical neural operator frameworks, their associated operator approximation spaces, and the statement of our main results, establishing a theory-to-practice gap in infinite dimensions.

Operator Learning

In the following, we will be interested in the sampling complexity of operator learning. To simplify our discussion, we will only consider the special case $\mathcal{Y} = \mathbb{R}$, i.e. we consider the data-driven approximation of non-linear functionals $\mathcal{G} : \mathcal{X} \to \mathbb{R}$. Since our analysis concerns *lower bounds* on the sampling complexity (or *upper bounds* on the optimal convergence rates), the results continue to hold if the output space is replaced by a more general $\mathcal{Y}$ (finite-dimensional or infinite-dimensional). Thus, this reduction can be made without loss of generality. For our analysis, only the fact that the input space $\mathcal{X}$ is infinite-dimensional will be relevant.

Sampling complexity of Operator Learning

With this operator learning setting in mind, we first point out that the discussion of the sampling complexity of Section 6.2 carries over with only minor changes. For convenience, we repeat the main elements here.

We now consider a set of continuous operators $\mathbf{U} \subset C(\mathcal{X}) = C(\mathcal{X}; \mathbb{R})$ on a separable Banach space $\mathcal{X}$. We fix a Banach space of operators $\mathbf{V}$, equipped with a norm $\|\cdot\|_{\mathbf{V}}$. In analogy with the finite-dimensional setting, a map $\mathcal{A} : \mathbf{U} \to \mathbf{V}$ will be called a deterministic method using $N \in \mathbb{N}$ point measurements, if there exists $\mathbf{u} = (u_1, \ldots, u_N) \in \mathcal{X}^N$, and a map $\mathcal{Q} : \mathbb{R}^N \to \mathbf{V}$, such that

$$\mathcal{A}(\mathcal{G}) = \mathcal{Q}(\mathcal{G}(u_1), \ldots, \mathcal{G}(u_N)), \quad \forall\, \mathcal{G} \in \mathbf{U}.$$

We recall that each $\mathcal{G} \in \mathbf{U}$ is a continuous operator $\mathcal{G} : \mathcal{X} \to \mathbb{R}$, and hence the above expression is well-defined. Consistent with our earlier discussion, the set of all deterministic methods using N point measurements will be denoted by $\mathrm{Alg}_N(\mathbf{U}, \mathbf{V})$. It will be assumed that $\mathbf{U} \subset \mathbf{V}$ with a canonical embedding.

The approximation error of $\mathcal{A}$ in $\mathbf{V}$ is defined as

$$e(\mathcal{A}, \mathbf{U}, \mathbf{V}) = \sup_{\mathcal{G} \in \mathbf{U}} \|\mathcal{A}(\mathcal{G}) - \mathcal{G}\|_{\mathbf{V}},$$

and the optimal error is $e_N^{\mathrm{det}}(\mathbf{U}, \mathbf{V}) = \inf_{\mathcal{A} \in \mathrm{Alg}_N^{\mathrm{det}}(\mathbf{U}, \mathbf{V})} e(\mathcal{A}, \mathbf{U}, \mathbf{V})$. The optimal order of convergence for deterministic approximation in $\mathbf{V}$ using N point samples is defined as in (6.2.2).

Randomized methods for operator learning, as well as the corresponding errors and the optimal randomized order (6.2.5) are similarly defined, in analogy with Section 6.2; we here only recall that a randomized method using N point measurement (in expectation) is a tuple $(\mathcal{A}, \mathsf{N})$ consisting of a family $\mathcal{A} = (\mathcal{A}_\omega), \omega \in \Omega$ of maps $\mathcal{A}_\omega : \mathbf{U} \to \mathbf{V}$ indexed by a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a measurable function $\mathsf{N} : \Omega \to \mathbb{N}$ with the same properties as described there, and with $\mathcal{A}_\omega \in \mathrm{Alg}_{\mathsf{N}(\omega)}(\mathbf{U}, \mathbf{V})$. We continue to denote by $\mathrm{Alg}_N(\mathbf{U}, \mathbf{V})$ the set of all strongly measurable randomized methods with $\mathbb{E}_\omega[\mathsf{N}(\omega)] \leq N$ expected point evaluations.

In the following, we will derive upper bounds on $\beta_*(\mathbf{U}, \mathbf{V})$, where $\mathbf{U}$ is the unit ball in an operator learning approximation space, and $\mathbf{V}$ is a space of operators with distance measured by either an L^p-norm with $1 \leq p < \infty$, or by a sup-norm. We describe the relevant setting in the next section.

Input functions in infinite dimensions

In the finite-dimensional case, the domain $[0,1]^d$ is widely considered as a canonical prototype, and results obtained for $[0,1]^d$ usually extend to more general bounded domains $D \subset \mathbb{R}^d$, under mild assumptions. In the infinite-dimensional setting, there is no longer such a canonical choice. Therefore, we preface our discussion of the infinite-dimensional theory-to-practice gap with the introduction of suitable alternatives to the spaces $L^p([0,1]^d)$ and $C([0,1]^d)$ (the latter corresponding to the limit case $p = \infty$) to be considered in this work.

Approximation in $\mathbf{V} = L^p(\mu)$ When the approximation error is measured in an L^p-norm for $1 \leq p < \infty$, we consider the following prototypical setting: We are given a probability measure μ on the input function space $\mathcal{X}$. We then consider the Banach space of p-integrable (real-valued) operators $\mathbf{V} = L^p(\mu)$, with the following L^p-norm:

$$\|\mathcal{G}\|_{L^p(\mu)} := \mathbb{E}_{u\sim\mu}[|\mathcal{G}(u)|^p]^{1/p}.$$

Thus, in the infinite-dimensional setting, the probability measure μ replaces the Lebesgue measure on $[0,1]^d$. To ensure that μ is "truly infinite-dimensional", we will make the following assumption.

***Assumption* 6.3.1.** There exist bi-orthogonal sequences $\{e_j\}_{j\in\mathbb{N}} \subset \mathcal{X}$ and $\{e_j^*\}_{j\in\mathbb{N}} \subset \mathcal{X}^*$, such that $e_j^*(e_k) = \delta_{jk}$ for all $j,k \in \mathbb{N}$, and such that the probability measure $\mu \in \mathcal{P}(\mathcal{X})$ can be written as the law of

$$u = \sum_{j=1}^{\infty} Z_j e_j, \quad \text{with } (Z_1, Z_2, \dots) \sim \textstyle\prod_{j=1}^{\infty} \rho_j(z_j)\, \mathsf{d}z_j, \tag{6.3.1}$$

Here, the components Z_j are independent, real-valued random variables, and the law of Z_j has probability density $\rho_j \in L^1(\mathbb{R};[0,\infty))$. For each $j \in \mathbb{N}$, we assume that there exists a non-empty interval $I_j \subset \mathbb{R}$ such that $\operatorname{ess\,inf}_{z_j\in I_j} \rho_j(z_j) > 0$. We assume that the series (6.3.1) converges with probability 1, i.e. that the essential supports of the ρ_j decay suitably fast. ◊

A decomposition of μ Let $\mu \in \mathcal{P}(\mathcal{X})$ be a probability measure satisfying Assumption 6.3.1. In the following, it will be useful to consider $u \sim \mu$ as being parametrized by the coefficients in the expansion (6.3.1). Let $d \in \mathbb{N}$ be fixed

for the following discussion. With a slight abuse of notation, we will then write, for $y = (y_1, \dots, y_d) \in \mathbb{R}^d$ and $\xi = \sum_{j>d} \xi_j e_j \in \mathcal{X}$,

$$u(y; \xi) = \xi + \sum_{j=1}^{d} y_j e_j, \quad \big(y \in \mathbb{R}^d,\ \xi \in \Omega_d\big), \tag{6.3.2}$$

where we have introduced

$$\Omega_d := \{\xi \in \mathcal{X} \,|\, e_1^*(\xi) = \cdots = e_d^*(\xi) = 0\} \subset \mathcal{X}. \tag{6.3.3}$$

Associated with y, ξ are the following probability measures $\mu_d \in \mathcal{P}(\mathbb{R}^d)$ and $\mu_d^\perp \in \mathcal{P}(\Omega_d)$, respectively: first, μ_d is defined by

$$\mu_d := \prod_{j=1}^{d} \rho_j(z_j)\, \mathsf{d}z_j. \tag{6.3.4}$$

Second, $\mu_d^\perp$ is defined as the law of the random variable,

$$\xi = \sum_{j>d} \xi_j e_j, \quad \text{where } (\xi_{d+1}, \xi_{d+2}, \dots) \sim \prod_{j>d} \rho_j(z_j)\, \mathsf{d}z_j. \tag{6.3.5}$$

By assumption on μ, the random variable (6.3.1) is equal in law to (6.3.2), when $y \sim \mu_d$ and $\xi \sim \mu_d^\perp$ are sampled independently. We can thus think of μ as essentially equivalent to the product measure $\mu_d \otimes \mu_d^\perp$ for any $d \in \mathbb{N}$. This structure will be convenient for our derivation of the infinite-dimensional theory-to-practice gap.

Technically, our analysis only requires that the following d-dependent assumption holds for any $d \in \mathbb{N}$.

***Assumption* 6.3.1$^{(\mathrm{d})}$.** There exist linearly independent $\{e_1, \dots, e_d\} \subset \mathcal{X}$ with bi-orthogonal $\{e_1^*, \dots, e_d^*\} \subset \mathcal{X}^*$, such that $\mu \in \mathcal{P}(\mathcal{X})$ is the law of

$$u = \xi + \sum_{j=1}^{d} y_j e_j, \quad (y_1, \dots, y_d) \sim \mu_d,\ \xi \sim \mu_d^\perp,$$

where for $\mu_d \in \mathcal{P}(\mathbb{R}^d)$ and $\mu_d^\perp \in \mathcal{P}(\Omega_d)$:

(a) the law of $(y_1, \dots y_d)$ is of the form $\mu_d = \prod_{j=1}^d \rho_j(z_j)\, \mathsf{d}z_j$, for $\rho_j \in L^1(\mathbb{R})$,

(b) there exist intervals $I_j \subset \mathbb{R}$ such that $\operatorname{ess\,inf}_{z_j \in I_j} \rho_j(z_j) > 0$ for $j = 1, \dots, d$,

(c) ξ satisfies $e_1^*(\xi) = \cdots = e_d^*(\xi) = 0$ almost surely.

$\Diamond$

Defining ξ by (6.3.5), one readily observes that Assumption 6.3.1$^{(d)}$ is implied by Assumption 6.3.1. For later reference, we also note the following lemma.

***Lemma* 6.3.2.** Let $d \in \mathbb{N}$ be given. Let $\mu \in \mathcal{P}(\mathcal{X})$ satisfy Assumption 6.3.1 or Assumption 6.3.1$^{(d)}$. There exists $B > 0$, such that $\mu_d^\perp(\|\xi\|_\mathcal{X} \leq B) > 0$. $\Diamond$

Proof. Note that if $u \sim \mu$, then $\xi \equiv u - \sum_{j=1}^d e_j^*(u)e_j$ is a well-defined random variable, with law $\mu_d^\perp \in \mathcal{P}(\Omega_d)$ (by definition). In particular, we have $\|\xi\|_\mathcal{X} < \infty$ almost surely. Since $\mu_d^\perp(\|\xi\|_\mathcal{X} < \infty) = \lim_{B\to\infty} \mu_d^\perp(\|\xi\|_\mathcal{X} \leq B)$, the claim is immediate. $\square$

This concludes our discussion of the $L^p(\mu)$-setting considered in this work.

Approximation in $\mathbf{V} = C(\mathcal{K})$ When the goal is to approximate $\mathcal{G} \in \mathbf{U}$ uniformly, i.e. with respect to the L^∞-norm, we restrict attention to a compact set $\mathcal{K} \subset \mathcal{X}$. In this case, we consider the Banach space of continuous operators $\mathbf{V} = C(\mathcal{K})$, endowed with the sup-norm:

$$\|\mathcal{G}\|_{C(\mathcal{K})} := \sup_{u\in\mathcal{K}} |\mathcal{G}(u)|.$$

Thus, in the infinite-dimensional setting, the compact set $\mathcal{K}$ replaces the unit cube $[0,1]^d$. To ensure that $\mathcal{K}$ is truly infinite-dimensional, and we make the following assumption.

***Assumption* 6.3.3.** We assume that the set $\mathcal{K}$ is (i) *convex* and (ii) there does *not* exist a finite-dimensional subspace $\mathcal{X}_0 \subset \mathcal{X}$ containing $\mathcal{K}$. $\Diamond$

We can relate this uniform setting to the $L^p(\mu)$-setting described above:

***Proposition* 6.3.4.** Let $\mathcal{K}$ satisfy Assumption 6.3.3. Then for any $d \in \mathbb{N}$, there exists a probability measure $\mu \in \mathcal{P}(\mathcal{X})$, with $\mathrm{supp}(\mu) \subset \mathcal{K}$, satisfying Assumption 6.3.1$^{(d)}$; more specifically, μ is the law of

$$u = e_0 + \sum_{j=1}^d y_j e_j, \qquad (y_1, \ldots, y_d) \sim \prod_{j=1}^d \mathrm{Unif}([0,1]).$$

Here $e_1, \dots, e_d \in \mathcal{X}$ are linearly independent elements for which there exist bi-orthogonal elements $e_1^*, \dots, e_d^* \in \mathcal{X}^*$, and $e_0 \in \mathcal{K}$ is such that $e_1^*(e_0) = \cdots = e_d^*(e_0) = 0$. ◊

Proof. Let $d \in \mathbb{N}$ be given. Our aim is to construct $\mu \in \mathcal{P}(\mathcal{X})$ as in the claim of Proposition 6.3.4. By assumption, $\mathcal{K}$ is infinite-dimensional. It follows that there exist $v_0, \dots, v_d \in \mathcal{K}$ which are linearly independent. Given such a choice, we set $e_0 := v_0$, and $e_j := \frac{1}{d}(v_j - v_0)$ for $j = 1, \dots, d$. Since $v_0, \dots, v_d$ are linearly independent, it follows that also $e_0, \dots, e_d$ are linearly independent. Furthermore, for any finite set of linearly independent vectors, we can find bi-orthogonal elements $e_0^*, \dots, e_d^* \in \mathcal{X}^*$, such that $e_j^*(e_k) = \delta_{jk}$, $j, k = 0, \dots, d$. In particular, for this choice we then have $e_1^*(e_0) = \cdots = e_d^*(e_0) = 0$.

We now define $\mu \in \mathcal{P}(\mathcal{X})$ as the law of $u := e_0 + \sum_{j=1}^d y_j e_j$, with $y_1, \dots, y_d \overset{iid}{\sim} \mathrm{Unif}([0,1])$. This μ trivially satisfies Assumption 6.3.1(d). To prove Proposition 6.3.4, it thus only remains to show that $\mathrm{supp}(\mu) \subset \mathcal{K}$. To see this, we observe that

$$\begin{aligned} u = e_0 + \sum_{j=1}^d y_j e_j &= v_0 + \frac{1}{d}\sum_{j=1}^d y_j (v_j - v_0) \\ &= \left(1 - \frac{1}{d}\sum_{j=1}^d y_j\right) v_0 + \sum_{j=1}^d \frac{y_j}{d} v_j =: \sum_{j=0}^d \lambda_j v_j, \end{aligned}$$

where $\lambda_0 := 1 - \frac{1}{d}\sum_{j=1}^d y_j$, $\lambda_j := \frac{1}{d} y_j$ for $j = 1, \dots, d$. The last sum identifies u as a convex combination of $v_0, \dots, v_d \in \mathcal{K}$: Indeed, since $y_j \in [0,1]$, for $j = 1, \dots, d$, it follows that $\lambda_0, \dots, \lambda_d \geq 0$ and we also have $\sum_{j=0}^d \lambda_j = 1$ by definition of λ_0. It now follows from the convexity of $\mathcal{K}$ that $u \in \mathcal{K}$ for any choice of $y_1, \dots, y_d \in [0,1]$, and hence $\mathrm{supp}(\mu) \subset \mathcal{K}$. This concludes the proof. □

A consequence of Proposition 6.3.4 is that approximation of a continuous operator $\mathcal{G} : \mathcal{K} \to \mathcal{Y}$ with respect to the $C(\mathcal{K})$-norm is at least as difficult as approximation with respect to the $L^p(\mu)$-norm, for *any* $p \in [1, \infty)$, and where μ satisfies Assumption 6.3.1(d). Indeed, for any reconstruction method $\mathcal{A} : \mathbf{U} \to C(\mathcal{K})$, we have

$$\|\mathcal{G} - \mathcal{A}(\mathcal{G})\|_{L^p(\mu)} \leq \|\mathcal{G} - \mathcal{A}(\mathcal{G})\|_{L^\infty(\mu)} \leq \|\mathcal{G} - \mathcal{A}(\mathcal{G})\|_{C(\mathcal{K})},$$

where the last inequality follows from $\text{supp}(\mu) \subset \mathcal{K}$. This implies that the optimal convergence rate with respect to $C(\mathcal{K})$ can be at most as large as the optimal convergence rate with respect to $L^p(\mu)$:

***Lemma* 6.3.5.** Let $\mathcal{K} \subset \mathcal{X}$ be compact. If $\mu \in \mathcal{P}(\mathcal{X})$ is a probability measure such that $\text{supp}(\mu) \subset \mathcal{K}$, then

$$\beta_*(\mathbf{U}, C(\mathcal{K})) \leq \beta_*(\mathbf{U}, L^p(\mu)). \tag{6.3.6}$$

$\Diamond$

This simple observation will allow us to deduce results about the uniform setting from corresponding results in the $L^p(\mu)$ setting, and derive estimates on $\beta_*(\mathbf{U}, C(\mathcal{K}))$ by passing to the limit $p \to \infty$.

From Finite to Infinite Dimensions

Neural operators $\Psi : \mathcal{X} \to \mathbb{R}$, approximating $\mathcal{G} : \mathcal{X} \to \mathbb{R}$, can often be interpreted as a composition of two mappings, $\Psi(u) = \psi \circ \mathcal{E}(u)$. Here, $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$ is an encoder, which maps the infinite-dimensional input into a finite-dimensional latent space, and $\psi : \mathbb{R}^d \to \mathbb{R}$ is the realization of a finite-dimensional neural network. The latent dimension $d \in \mathbb{N}$ is a hyperparameter of the architecture.

The general idea behind our proof of the infinite-dimensional theory-to-practice gap is the following: If the relevant approximation space $\mathbf{U} \subset C(\mathcal{X})$ contains compositions of the form $f \circ \mathcal{E} : \mathcal{X} \to \mathbb{R}$ where $f \in U_\ell^{\alpha,\infty}$ belongs to the d-dimensional neural network approximation space, then the mapping

$$U_\ell^{\alpha,\infty} \to \mathbf{U}, \quad f \mapsto f \circ \mathcal{E},$$

defines an embedding of $U_\ell^{\alpha,\infty}$ into $\mathbf{U}$. Thus, the sampling complexity of $\mathbf{U}$ should be at least as large as that of $U_\ell^{\alpha,\infty}$. As a consequence, any d-dimensional upper bound on convergence rate implies a corresponding bound in the infinite-dimensional case. This general intuition is confirmed and made precise by the following proposition:

***Proposition* 6.3.6** (L^p setting)**.** Let $\mathcal{X}$ be a separable Banach space, let $\mu \in \mathcal{P}(\mathcal{X})$ be a probability measure on $\mathcal{X}$, and let $\mathbf{U} \subset C(\mathcal{X})$ be a set of continuous operators. Assume that there is an encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$ and a constant $c > 0$,

such that

$$\mathcal{E}_{\#}\mu \geq c \cdot \mathrm{Unif}([0,1]^d). \tag{6.3.7}$$

If for some $\alpha > 0$ and $\ell : \mathbb{N} \to \mathbb{N}$, we have

$$\left\{ f \circ \mathcal{E} : \mathcal{X} \to \mathbb{R} \,\middle|\, f \in U_\ell^{\alpha,\infty}(\mathbb{R}^d) \right\} \subset \mathbf{U}, \tag{6.3.8}$$

then

$$\beta_*(\mathbf{U}, L^p(\mu)) \leq \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor}. \tag{6.3.9}$$

$\Diamond$

Our proof of Proposition 6.3.6 will be based on the following two lemmas. The first lemma shows that, under the assumptions of Proposition 6.3.6 any deterministic reconstruction method $\mathcal{A} \in \mathrm{Alg}^{\mathrm{det}}(\mathbf{U}, L^p(\mu))$ in infinite dimensions induces an associated finite-dimensional reconstruction method $A \in \mathrm{Alg}^{\mathrm{det}}(U_\ell^{\alpha,\infty}, L^p([0,1]^d))$.

***Lemma* 6.3.7.** With the notation and under assumptions (6.3.7) and (6.3.8) of Proposition 6.3.6, the following holds: For any $\mathcal{A} \in \mathrm{Alg}^{\mathrm{det}}(\mathbf{U}, L^p(\mu))$, there exists $A \in \mathrm{Alg}^{\mathrm{det}}(U_\ell^{\alpha,\infty}, L^p([0,1]^d))$, such that

$$\|f \circ \mathcal{E} - \mathcal{A}(f \circ \mathcal{E})\|_{L^p(\mu)} \geq c \|f - A(f)\|_{L^p([0,1]^d)}, \quad \forall\, f \in U_\ell^{\alpha,\infty}. \tag{6.3.10}$$

Here $c > 0$ is the constant of (6.3.7). $\Diamond$

Proof Sketch. The detailed proof of Lemma 6.3.7 is included in Appendix E.2. The basic idea of the proof is the following: By definition, $\mathcal{A}$ is of the form $\mathcal{A}(\mathcal{G}) = \mathcal{Q}(\mathcal{G}(u_1), \ldots, \mathcal{G}(u_N))$ for some sampling points $u_1, \ldots, u_N \in \mathcal{X}$ and $\mathcal{Q} : \mathbb{R}^N \to L^p(\mu)$. We want to construct $A : U_\ell^{\alpha,\infty} \to L^p([0,1]^d)$, of the form $A(f) = Q(f(x_1), \ldots, f(x_N))$, where $Q : \mathbb{R}^N \to L^p([0,1]^d)$. The canonical choice of the sampling points $x_1, \ldots, x_N \in \mathbb{R}^d$ is via composition with the encoder, $x_j := \mathcal{E}(u_j)$. The main remaining question is then how to construct the mapping $Q : \mathbb{R}^N \to L^p([0,1]^d)$ from $\mathcal{Q} : \mathbb{R}^N \to L^p(\mu)$. A first idea is that this reconstruction could satisfy

$$Q(y_1, \ldots, y_N)(\mathcal{E}(u)) := \mathcal{Q}(y_1, \ldots, y_N)(u), \quad \forall\, u \in \mathcal{X},\ \forall\, (y_1, \ldots, y_N) \in \mathbb{R}^N.$$

However, this is not well-defined, since different u will generally map to the same $x = \mathcal{E}(u)$. The improved guess is the following: Fix $x \in \mathbb{R}^d$ and consider

a random variable $u \sim \mu$. We now condition on the event that $\mathcal{E}(u) = x$. This gives a conditional distribution on the input function space. We then *average* the reconstruction $\mathcal{Q}(y_1, \dots, y_N)(u)$ in u over this conditional distribution, i.e. define

$$Q(y_1, \dots, y_N)(x) := \mathbb{E}_{u\sim\mu}[\mathcal{Q}(y_1, \dots, y_N)(u) \,|\, \mathcal{E}(u) = x].$$

This is well-defined, and due to Jensen's inequality the conditional averaging on the right-hand side turns out to reduce the reconstruction error of A compared to $\mathcal{A}$. The detailed calculations are provided in Appendix E.2. □

The last lemma is in anticipation of our next result, Lemma 6.3.8. The final result Proposition 6.3.6 will then be an immediate consequence.

***Lemma* 6.3.8.** With the notation and under assumptions (6.3.7) and (6.3.8) of Proposition 6.3.6, the following holds. There exists $\kappa > 0$, such that for every $N \in \mathbb{N}$, there exists a probability space $(\Xi, \mathbb{P})$ and a random variable $\Psi : \Xi \to \mathbf{U}$, $\xi \mapsto \Psi_\xi$, such that

$$\mathbb{E}_\xi\big[\|\Psi_\xi - \mathcal{A}(\Psi_\xi)\|_{L^p(\mu)}\big] \geq \kappa N^{-\lambda}, \quad \forall\, \mathcal{A} \in \mathrm{Alg}^{\mathrm{det}}(\mathbf{U}, L^p(\mu)), \tag{6.3.11}$$

where $\lambda = \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor}$. ◊

Proof. By Lemma 6.2.8, there exists a constant $\kappa > 0$, such that for any $N \in \mathbb{N}$, there exists a probability space $(\Xi, \mathbb{P})$ and random variable $\psi : \Xi \to U_\ell^{\alpha,\infty}$, $\xi \mapsto \psi_\xi$, such that

$$\mathbb{E}_\xi\big[\|\psi_\xi - A(\psi_\xi)\|_{L^p([0,1]^d)}\big] \geq \kappa N^{-\lambda}, \quad \forall\, A \in \mathrm{Alg}^{\mathrm{det}}(U_\ell^{\alpha,\infty}, L^p([0,1]^d)),$$

and where $\lambda := \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor}$. We use ψ_ξ to define a new random variable $\Psi : \Xi \to \mathbf{U}$, $\Psi_\xi := \psi_\xi \circ \mathcal{E}$. We claim that (6.3.11) holds for this Ψ_ξ.

To see this, let $\mathcal{A} \in \mathrm{Alg}^{\mathrm{det}}(\mathbf{U}, L^p(\mu))$ be given. By Lemma 6.3.7, there exists $A \in \mathrm{Alg}^{\mathrm{det}}(\mathbf{U}, L^p([0,1]^d))$, such that

$$\|\psi_\xi \circ \mathcal{E} - \mathcal{A}(\psi_\xi \circ \mathcal{E})\|_{L^p(\mu)} \geq c\|\psi_\xi - A(\psi_\xi)\|_{L^p([0,1]^d)}, \quad \forall\, \xi \in \Xi.$$

Here $c > 0$ is the constant appearing in (6.3.7) (and is independent of N). Taking expectations over ξ, it follows that

$$\mathbb{E}_\xi\big[\|\Psi_\xi - \mathcal{A}(\Psi_\xi)\|_{L^p(\mu)}\big] \geq c\mathbb{E}_\xi\big[\|\psi_\xi - A(\psi_\xi)\|_{L^p([0,1]^d)}\big].$$

By construction of ψ_ξ, the right hand side is lower bounded by $c\kappa N^{-\lambda}$, with $\kappa > 0$ independent of N and $\lambda = \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell^*/2 \rfloor}$. Thus it follows that

$$\mathbb{E}_\xi \left[\|\Psi_\xi - \mathcal{A}(\Psi_\xi)\|_{L^p(\mu)} \right] \geq c\kappa N^{-\lambda},$$

where $c, \kappa > 0$ are independent of N. Replacing $c\kappa$ by κ, the claimed bound (6.3.11) follows. □

The proof of Proposition 6.3.6 is now immediate.

Proof of Proposition 6.3.6. Proposition 6.3.6 follows from Lemma 6.3.8 and Lemma 6.2.8. □

Deep Operator Networks (DeepONet)

In this section, we state and prove a theory-to-practice gap for a general family of "DeepONet" architectures. We recall that we are interested in the approximation of operators $\mathcal{G} : \mathcal{X} \to \mathbb{R}$. In this setting, we define these DeepONet architectures to be of the form $\Psi = \psi \circ \mathcal{L}$, combining a linear encoder $\mathcal{L} : \mathcal{X} \to \mathbb{R}^d$ with a feedforward neural network $\psi : \mathbb{R}^d \to \mathbb{R}$. The next three paragraphs provide a precise description of the considered architecture, define relevant approximation spaces, and prove an infinite-dimensional theory-to-practice gap for these architectures.

Architecture Fix a sequence of continuous linear functionals $\ell_1, \ell_2, \cdots : \mathcal{X} \to \mathbb{R}$. For $d_0 \in \mathbb{N}$, we denote by $\mathcal{L}_{d_0}$ the linear encoder $\mathcal{L}_{d_0} : \mathcal{X} \to \mathbb{R}^{d_0}$, $\mathcal{L}_{d_0}(u) := (\ell_1(u), \ldots, \ell_{d_0}(u))$. The **encoder-net** $\Psi : \mathcal{X} \to \mathbb{R}$ associated with a neural network $\psi : \mathbb{R}^{d_0} \to \mathbb{R}$ is a mapping of the form $\Psi(u) = \psi \circ \mathcal{L}_{d_0}(u)$.

To ensure universality of the resulting operator learning architecture, we will make the (minimal) assumption that

$$\text{span}\{\ell_j : \mathcal{X} \to \mathbb{R} \,|\, j \in \mathbb{N}\} \subset \mathcal{X}^* \text{ is dense}, \tag{6.3.12}$$

where we recall that $\mathcal{X}^*$ denotes the continuous dual of $\mathcal{X}$. Throughout the following discussion, we will consider the sequence $(\ell_j)_{j \in \mathbb{N}} \subset \mathcal{X}^*$ fixed, and it will be assumed that (6.3.12) holds without further mention.

DeepONet Approximation Space We can now define spaces $\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{DON}}$ for DeepONets. To this end, we introduce

$$\Sigma^{\ell}_{n,\mathrm{DON}} := \left\{ \Psi = \psi \circ \mathcal{L}_{d_0} : \begin{array}{l} \psi \text{ NN with } d_{\mathrm{in}}(\psi) = d_0, d_{\mathrm{out}}(\psi) = 1, \\ \max\{W(\psi), d_0\} \leq n, L(\psi) \leq \ell(n), \|\psi\|_{\mathsf{NN}} \leq 1 \end{array} \right\}.$$

Then, given $\alpha \in (0,\infty)$, for each continuous (non-linear) operator $\mathcal{G} : \mathcal{X} \to \mathbb{R}$, we define

$$\Gamma^{\alpha,\infty}_{\mathrm{DON}}(\mathcal{G}) := \max\left\{ \sup_{u\in\mathcal{X}} \|\mathcal{G}(u)\|,\ \sup_{n\in\mathbb{N}} \left[n^{\alpha} \cdot d_{\infty}\big(\mathcal{G}, \Sigma^{\ell}_{n,\mathrm{DON}}\big)\right] \right\} \in [0,\infty],$$

where $d_{\infty}(\mathcal{G},\Sigma) := \inf_{\Psi\in\Sigma} \sup_{u\in\mathcal{X}} \|\mathcal{G}(u) - \Psi(u)\|$. We can define a DeepONet **approximation space quasi-norm** $\|\cdot\|_{\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{DON}}}$ by

$$\|\mathcal{G}\|_{\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{DON}}} := \inf\{\theta > 0 : \Gamma^{\alpha,\infty}_{\mathrm{DON}}(\mathcal{G}/\theta) \leq 1\} \in [0,\infty],$$

giving rise to the DeepONet **approximation space**

$$\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{DON}} := \left\{ \mathcal{G} \in C(\mathcal{X}) : \|\mathcal{G}\|_{\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{DON}}} < \infty \right\}.$$

Encoder Construction The following is a useful technical lemma, which will be applied to construct suitable encoders $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$. It shows that if a finite-dimensional map $F : V \subset \mathbb{R}^d \to \mathbb{R}^d$ is sufficiently close to the identity, then the image of V must "fill out" a non-empty open set $V_0 \subset \mathbb{R}^d$.

***Lemma* 6.3.9.** Let $V \subset \mathbb{R}^d$ be a non-empty domain. There exist constants $\epsilon_0, c_0 > 0$ and a non-empty open subset $V_0 \subset V$ with the following property: For any Lipschitz-continuous function $F : V \to \mathbb{R}^d$, satisfying

$$\|F - \mathrm{id}\|_{W^{1,\infty}(V)} \leq \epsilon_0,$$

where $\mathrm{id} : V \to \mathbb{R}^d$, $\mathrm{id}(y) = y$ denotes the identity mapping, it follows that $F(V) \supset V_0$, and

$$F_{\#}\mathrm{Unif}(V) \geq c_0 \mathrm{Unif}(V_0).$$

$\Diamond$

The result of Lemma 6.3.9 follows as a consequence of the contraction mapping theorem; we provide a detailed proof in Appendix E.2. Our goal in this section is to construct encoders $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$ which "fill out" the set $[0,1]^d$. The link with the finite-dimensional setting of Lemma 6.3.9 is made by identifying

$\mathcal{X} \simeq \mathbb{R}^d \times \Omega_d$, as in the decomposition of μ in (6.3.2), and by considering the second factor as a parameter. This leads us to study parametrized mappings, $F : V \times \Omega_d \to \mathbb{R}^d$, $(y, \xi) \mapsto F_\xi(y) = F(y; \xi)$, with the parameter $\xi \in \Omega_d$ a random variable. The following result derives a similar result as Lemma 6.3.9 in this parametrized setting:

***Lemma* 6.3.10.** Let $V \subset \mathbb{R}^d$ be a non-empty domain, and let $\epsilon_0, c_0 > 0$ and $V_0 \subset V$ denote the constants and set of Lemma 6.3.9, respectively. Let $(\Omega, \mathbb{P})$ be a probability space, and assume that $F : V \times \Omega \to \mathbb{R}^d$, $(y, \xi) \mapsto F(y; \xi)$ is measurable. Assume furthermore, that there exists $K \subset \Omega$, such that the mapping $F_\xi : V \to \mathbb{R}^d$, $y \mapsto F_\xi(y) := F(y; \xi)$ is Lipschitz for each $\xi \in K$, and

$$\|F_\xi - \mathrm{id}\|_{W^{1,\infty}(V)} \leq \epsilon_0, \quad \forall\, \xi \in K. \tag{6.3.13}$$

Then the push-forward under F of the product measure $\mathrm{Unif}(V) \otimes \mathbb{P}$ on $V \times \Omega$ satisfies

$$F_\#(\mathrm{Unif}(V) \otimes \mathbb{P}) \geq c_0 \mathbb{P}(K)\, \mathrm{Unif}(V_0).$$

$\lozenge$

A detailed proof of Lemma 6.3.10 is given in Appendix E.2. Let now $\{\ell_k\} \subset \mathcal{X}^*$ be a set of encoding functionals, such that $\mathrm{span}\{\ell_k\} \subset \mathcal{X}^*$ is dense. Our first goal is to use the $\{\ell_k\}$ to construct an encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$, such that $\mathcal{E}_\# \mu \geq c\mathrm{Unif}([0,1]^d)$.

To this end, let us momentarily fix $\delta > 0$. Then by density, there exists $d_0 \in \mathbb{N}$ and coefficients c_{jk} for $j \in [d]$, $k \in [d_0]$, such that

$$\left\| e_j^* - \sum_{k=1}^{d_0} c_{jk} \ell_k \right\|_{\mathcal{X}^*} \leq \delta, \quad \forall\, j = 1, \ldots, d. \tag{6.3.14}$$

Since $e_j^*(u(y; \xi)) = y_j$, (6.3.14) allows us to approximate a "projection" onto y_j. Motivated by this, we now define the encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$, $\mathcal{E}(u) := (\mathcal{E}_1(u), \ldots, \mathcal{E}_d(u))$, via

$$\mathcal{E}_j(u) = b_j + \sum_{k=1}^{d_0} a_{jk} \ell_k(u), \tag{6.3.15}$$

for coefficients a_{jk} and bias b_j, for $j \in [d]$ and $k \in [d_0]$, to be determined.

***Proposition* 6.3.11.** Assume that $\mu \in \mathcal{P}(\mathcal{X})$ satisfies Assumption 6.3.1[(d)] for $d \in \mathbb{N}$. Then there exists an encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$ of the form (6.3.15) and constant $c > 0$, such that

$$\mathcal{E}_{\#}\mu \geq c\,\mathrm{Unif}([0,1]^d). \tag{6.3.16}$$

$\Diamond$

The proof of Proposition 6.3.11 is a straight-forward, albeit somewhat tedious, consequence of Lemma 6.3.10 and the fact that encoders of the form (6.3.15) are dense in the space of all affine encoders with d-dimensional range. The proof relies on the assumption that the linear functionals $\{\ell_k\}_{k\in\mathbb{N}}$ are dense in $\mathcal{X}^*$. We include the detailed argument in Appendix E.2.

Theory-to-Practice Gap We can now state a theory-to-practice gap for the unit ball $\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}$ in the DeepONet approximation space $\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{DON}}$:

***Theorem* 6.3.12** (DeepONet theory-to-practice gap)**.** Let $p \in [1,\infty]$. Let $\ell : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ be non-decreasing with $\ell^* \geq 4$. Assume that $\mu \in \mathcal{P}(\mathcal{X})$ satisfies Assumption 6.3.1. Then for any $\alpha > 0$, we have

$$\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, L^p(\mu)) \leq \frac{1}{p}. \tag{6.3.17}$$

$\Diamond$

We recall that the typical Monte-Carlo (MC) approximation rate in the L^p-norm is $\beta_{MC} = 1/p$, reducing to the well-known $1/2$-rate with respect to the L^2-norm. Theorem 6.3.12 shows that, *independently of* $\alpha > 0$ *and the depth* $\ell^* \geq 4$, it is not possible to achieve better-than-MC rates by any approximation method on the relevant approximation spaces $\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{DON}}$.

Proof. Fix $d \in \mathbb{N}$, and let $\ell : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ be a non-decreasing function with $\ell^* \geq 4$. We denote $\widetilde{\ell} := \ell - 3$, so that $\widetilde{\ell}^* \geq 3$, as in the assumptions of the *finite dimensional* theory-to-practice gap, Theorem 6.2.2. Let $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$ be the encoder of Proposition 6.3.11, such that $\mathcal{E}_{\#}\mu \geq c\,\mathrm{Unif}([0,1]^d)$. Let $\gamma_d \cdot \mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}$ denote re-scaling of $\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}$ by a constant scaling factor γ_d. By Lemma 6.3.13, which we state below after the proof, there exists a constant $\gamma_d \geq 1$, such that we have

$$\left\{ f \circ \mathcal{E} \,\middle|\, f \in U^{\alpha,\infty}_{\widetilde{\ell}}(\mathbb{R}^d) \right\} \subset \gamma_d \cdot \mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}. \tag{6.3.18}$$

Assuming (6.3.18), the claim of Theorem 6.3.12 then follows immediately from Proposition 6.3.6. Indeed, defining $\mathbf{U} := \gamma_d \cdot \mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}$, that proposition implies that

$$\beta_*(\gamma_d \cdot \mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, L^p(\mu)) \leq \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \widetilde{\ell}^*/2 \rfloor}.$$

However, it follows from the definition that β_* is invariant under re-scaling,

$$\beta_*(\gamma_d \cdot \mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, L^p(\mu)) = \beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, L^p(\mu)).$$

Thus, recalling also $\widetilde{\ell}^* = \ell^* - 1$, we have

$$\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, L^p(\mu)) \leq \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor (\ell^* - 1)/2 \rfloor}. \tag{6.3.19}$$

Since $d \in \mathbb{N}$ was arbitrary and the left-hand side is independent of d, we can take the infimum over all $d \in \mathbb{N}$ on the right to conclude that

$$\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, L^p(\mu)) \leq \frac{1}{p}.$$

This is (6.3.17).

□

***Lemma* 6.3.13.** Let $D \subset \mathbb{R}^d$, $\ell : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ non-decreasing and $f \in U^{\alpha,\infty}_{\ell}(D)$. Then for every $e \in \mathbb{N}$, $C \in \mathbb{R}^{d \times e}$ and $b \in \mathbb{R}^d$ there is $R \in (0, \infty)$ with $f(C \cdot + b) \in R \cdot U^{\alpha,\infty}_{\ell}(E)$, where $E := \{x \in \mathbb{R}^e : \ Cx + b \in D\} \subset \mathbb{R}^e$. ◊

The detailed proof of Lemma 6.3.13 is given in Appendix E.2. We also state the following theory-to-practice gap for uniform approximation over compact $\mathcal{K}$:

***Theorem* 6.3.14** (DeepONet; uniform theory-to-practice gap)**.** Let $\ell : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ be non-decreasing with $\ell^* \geq 4$. Assume that $\mathcal{K} \subset \mathcal{X}$ is a compact set satisfying Assumption 6.3.3. Then for any $\alpha > 0$, we have

$$\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, C(\mathcal{K})) = 0.$$

◊

Proof. By Proposition 6.3.4, for any $d \in \mathbb{N}$, there exists a probability measure $\mu \in \mathcal{P}(\mathcal{X})$, with $\mathrm{supp}(\mu) \subset \mathcal{K}$, and μ satisfies Assumption 6.3.1[(d)]. By

Proposition 6.3.11, there exists a DeepONet encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$ and constant $c > 0$, such that

$$\mathcal{E}_{\#}\mu \geq c\,\mathrm{Unif}([0,1]^d).$$

Following the steps in the proof of Theorem 6.3.12, leading up to (6.3.19), it follows that for any $p \in [1, \infty]$, we have

$$\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, L^p(\mu)) \leq \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor (\ell^* - 1)/2 \rfloor}.$$

We now recall that

$$\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, C(\mathcal{K})) \leq \beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, L^p(\mu)).$$

This inequality is (6.3.6) and follows from the fact that uniform approximation over $\mathcal{K}$ is a more stringent criterion than $L^p(\mu)$ approximation with respect to μ, owing to the fact that $\mathrm{supp}(\mu) \subset \mathcal{K}$. Thus, we have

$$\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, C(\mathcal{K})) \leq \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor (\ell^* - 1)/2 \rfloor}.$$

This holds for any $d \in \mathbb{N}$ and $p \in [1, \infty)$. The convergence rate $\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{DON}}, C(\mathcal{K}))$ on the left depends on α and ℓ, but is independent of p and d. Thus, upon letting $d, p \to \infty$, the claim follows. □

Integral-kernel Neural Operators

In this section, we state and prove a theory-to-practice gap for a general family of integral-kernel neural operator (NO) architectures [52, 19]. Again, we are interested in the approximation of operators $\mathcal{G} : \mathcal{X} \to \mathbb{R}$. In this setting, we define integral-kernel NO architectures to be of the form $\Psi = \mathcal{Q} \circ \mathcal{L}_L \circ \ldots \mathcal{L}_1 \circ \mathcal{R}$, combining a lifting layer $\mathcal{R}$, hidden layers $\mathcal{L}_1, \ldots, \mathcal{L}_L$ and an output layer $\mathcal{Q}$. The next three paragraphs provide a precise description of the considered architecture, define relevant approximation spaces, and prove an infinite-dimensional theory-to-practice gap.

Architecture The following is a minimal architecture shared by all/most variants of integral kernel-based neural operators [181]. For notational simplicity, we focus on real-valued input and output functions. All results extend readily to the more general vector-valued case.

***Definition* 6.3.15** (Averaging Neural Operator)**.** Let $\mathcal{X}(D;\mathbb{R})$, $\mathcal{Y}(D;\mathbb{R})$, and $\mathcal{V}(D;\mathbb{R}^{d_c})$ be spaces of functions on Lipschitz domain $D \subset \mathbb{R}^{d_D}$. An averaging neural operator (ANO) $\Psi : \mathcal{X}(D;\mathbb{R}) \to \mathcal{Y}(D;\mathbb{R})$ of depth L takes the form

$$\Psi(u) = \mathcal{Q} \circ \mathcal{L}_L \circ \dots \mathcal{L}_1 \circ \mathcal{R}(u), \tag{6.3.20}$$

where $u \in \mathcal{X}(D;\mathbb{R})$ and $x \in \mathbb{R}^d$. In addition, the pointwise lifting operators $\mathcal{R}$ and $\mathcal{Q}$ are obtained by composition with shallow ReLU neural networks; i.e. there exist neural networks $R : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^{d_c}$ and $Q : \mathbb{R}^{d_c} \to \mathbb{R}$, of depth $L(R) = L(Q) = 2$, such that

$$\mathcal{R}(u)(x) = R(u(x), x), \qquad \mathcal{Q}(v)(x) = Q(v(x)). \tag{6.3.21}$$

Finally, the hidden layers $\mathcal{L}_j : \mathcal{V}(D;\mathbb{R}^{d_c}) \to \mathcal{V}(D;\mathbb{R}^{d_c})$ take the form

$$\mathcal{L}_j(v)(x) = \sigma\left(W_j v(x) + b_j + \fint_D v(y)\, \mathsf{d}y\right), \tag{6.3.22}$$

where $W_j \in \mathbb{R}^{d_c \times d_c}$ is a matrix and $b_j \in \mathbb{R}^{d_c}$ a bias. ◊

Generalizing our definitions of quantities of interest from Section 6.2, we will denote by $L(\Psi) := L$ the depth (number of hidden layers) of an ANO, and we denote by $W(\Psi) = W(R) + \sum_{j=1}^{L}(\|W_j\|_{\ell^0} + \|b_j\|_{\ell^0}) + W(Q)$ the total number of non-zero parameters of the architecture. Furthermore, we define $\|\Psi\|_{\mathsf{NN}} := \max\{\|W_j\|_\infty, \|b_j\|_\infty, \|R\|_{\mathsf{NN}}, \|Q\|_{\mathsf{NN}}\}$ as the maximal weight magnitude.

***Remark* 6.3.16.** The ANO introduced above is a special case of a more general family of kernel-based neural operators introduced in [52]. Its theoretical significance is that most instantiations of such neural operators contain the ANO as a special case, with a specific tuning of the weights. For example, the FNO defined in 1.3.1 uses the same general structure, but employs hidden layers of the form

$$\mathcal{L}(v)(x) = \sigma\left(Wv(x) + b + \int \kappa(x-y)v(y)\, dy\right),$$

where the integral kernel κ is convolutional, and

$$\kappa(x) = \sum_{|k| \le k_{\max}} \widehat{\kappa}_k e^{ik\cdot x}$$

is parametrized by the coefficients $\widehat{\kappa}_k \in \mathbb{C}^{d_c \times d_c}$ in its (truncated) Fourier expansion. Thus, the ANO can be obtained from the FNO upon setting $\widehat{\kappa}_k \equiv 0$ for $k \neq 0$ and $\widehat{\kappa}_0 = I_{d_c \times d_c}/\mathrm{vol}(D)$. ◊

NO Approximation Space We now define the relevant approximation spaces $\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{NO}}$ for (averaging) neural operators. Assume we are given function spaces $\mathcal{X}(D) = \mathcal{X}(D;\mathbb{R})$ and $\mathcal{Y}(D) = \mathcal{Y}(D;\mathbb{R})$. In our discussion, we will assume that $\mathcal{X}(D) \subset L^\infty(D)$ is an infinite-dimensional Banach space on Lipschitz domain $D \subset \mathbb{R}^{d_D}$, and we will assume that $\mathcal{Y}(D)$ contains all constant functions. We now introduce

$$\Sigma^{\ell}_{n,\mathrm{NO}} := \left\{ \Psi : \mathcal{X} \to \mathcal{Y} : \begin{array}{l} \Psi \text{ is an ANO with } W(\Psi) \leq n, \\ L(\Psi) \leq \ell(n), \|\Psi\|_{\mathsf{NN}} \leq 1 \end{array} \right\}.$$

***Remark* 6.3.17.** Part of the definition of $\Sigma^{\ell}_{n,\mathrm{NO}}$ is that for any $\Psi \in \Sigma^{\ell}_{n,\mathrm{NO}}$, we must have $\Psi(\mathcal{X}) \subset \mathcal{Y}$. Non-trivial Ψ exist, since we can readily construct averaging neural operators Ψ of the form (6.3.20), for which the output $\Psi(u)$ is a *constant-valued function*, for any input $u \in \mathcal{X}$. Since $\mathcal{Y}$ contains constant functions by assumption, this implies that such Ψ defines a map $\Psi : \mathcal{X} \to \mathcal{Y}$. In our proofs, we will only ever consider Ψ of this form, thus our results hold even when $\mathcal{Y} = \mathbb{R}$. ◊

Given $\alpha \in (0,\infty)$, for each continuous (non-linear) operator $\mathcal{G} : \mathcal{X} \to \mathcal{Y}$, we define

$$\Gamma^{\alpha,\infty}_{\mathrm{NO}}(\mathcal{G}) := \max\left\{ \sup_{u\in\mathcal{X}} \|\mathcal{G}(u)\|,\ \sup_{n\in\mathbb{N}} \left[n^\alpha \cdot d_\infty\big(\mathcal{G}, \Sigma^{\ell}_{n,\mathrm{NO}}\big)\right] \right\} \in [0,\infty],$$

where $d_\infty(\mathcal{G},\Sigma) := \inf_{\Psi\in\Sigma} \sup_{u\in\mathcal{X}} \|\mathcal{G}(u) - \Psi(u)\|_{\mathcal{Y}}$. We define a NO **approximation space quasi-norm** $\|\cdot\|_{\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{NO}}}$ by

$$\|\mathcal{G}\|_{\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{NO}}} := \inf\{\theta > 0 : \Gamma^{\alpha,\infty}_{\mathrm{DON}}(\mathcal{G}/\theta) \leq 1\} \in [0,\infty],$$

giving rise to the NO **approximation space**

$$\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{NO}} := \left\{ \mathcal{G} \in C(\mathcal{X};\mathcal{Y}) : \|\mathcal{G}\|_{\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{NO}}} < \infty \right\}.$$

We again denote by $\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{NO}}$ the unit ball in $\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{NO}}$.

***Remark* 6.3.18.** As pointed out in Remark 6.3.16, the ANO can be obtained by a special setting of the weights in the FNO. As a consequence, it can be shown that $\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{NO}} \subset \mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{FNO}}$, where $\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{FNO}}$ denotes the relevant approximation space for FNO, which can be defined in analogy to $\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{NO}}$. Based on this relationship, it could be shown that

$$\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{FNO}}, L^p(\mu)) \leq \beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{NO}}, L^p(\mu)),$$

and hence any upper bound on $\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{NO}}, L^p(\mu))$ implies a corresponding upper bound for the FNO. ◊

Encoder Construction Let $\mu \in \mathcal{P}(\mathcal{X})$ be a probability measure on $\mathcal{X}$. We recall that $L^1(D)$ is a subset of the dual of $L^\infty(D)$ under the natural pairing,

$$\langle u, e^* \rangle = \int_D u(x) e^*(x) \ \mathsf{d}x, \quad \forall u \in L^\infty(D),\ e^* \in L^1(D).$$

The following proposition constructs an encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$, whose existence will imply a theory-to-practice gap for the averaging neural operator.

***Proposition* 6.3.19.** Let μ satisfy Assumption 6.3.1(d) for $d \in \mathbb{N}$, with bi-orthogonal elements $e_j^* \in L^1(D)$. Then there exists an encoder $\mathcal{E} : \mathcal{X}(D) \to \mathbb{R}^d$,

$$\mathcal{E}(u) = \fint_D R(u(x), x) \ \mathsf{d}x, \tag{6.3.23}$$

with $R : \mathbb{R} \times \mathbb{R}^{d_D} \to \mathbb{R}^d$ a shallow ReLU neural network, and constant $c > 0$ dependent on d, such that

$$\mathcal{E}_{\#}\mu \geq c \cdot \mathrm{Unif}([0,1]^d). \tag{6.3.24}$$

◊

Proof. It will suffice to show that there exists a neural network $R : \mathbb{R} \times \mathbb{R}^{d_D} \to \mathbb{R}^d$ and encoder of the form (6.3.23), such that for *some* non-empty open set $V_0 \subset \mathbb{R}^d$ and constant $c > 0$, we have

$$\mathcal{E}_{\#}\mu \geq c \cdot \mathrm{Unif}(V_0). \tag{6.3.25}$$

Indeed, given such V_0, there exists a scaling factor $\gamma > 0$ and shift $b \in \mathbb{R}^d$, such that $[0,1]^d \subset \gamma \cdot V_0 + b$. Replacing the neural network $R(\eta, x)$ by $\widetilde{R}(\eta, x) := \gamma \cdot R(\eta, x) + b$, it is then immediate that the encoder $\widetilde{\mathcal{E}} : \mathcal{X}(D) \to \mathbb{R}^d$ defined by $\widetilde{\mathcal{E}}(u) = \fint_D \widetilde{R}(u(x), x) \ \mathsf{d}x$ satisfies a lower bound of the form (6.3.24).

To prove the existence of an encoder $\mathcal{E}$ satisfying (6.3.25), we recall that, by Assumption 6.3.1(d), $u \sim \mathcal{P}(\mathcal{X})$ is of the form

$$u(x; y, \xi) = \xi(x) + \sum_{j=1}^{d} y_j e_j(x),$$

where $e_1, \dots, e_d$ are linearly independent with bi-orthogonal elements $e_1^*, \dots, e_d^*$, and the coefficients $y_j \sim \rho_j(y)\,\mathsf{d}y$ are independent. Furthermore, $\xi \sim \mu_d^\perp$ is a random function such that $e_1^*(\xi) = \cdots = e_d^*(\xi) = 0$. In the following we consider ξ a random "parameter" and denote the law of ξ by $\mathbb{P} := \mu_d^\perp$. By assumption the dual elements e_j^* are represented by a function in L^1.

Under our assumptions, there exist non-empty intervals $I_j \subset \mathbb{R}$ and constant $c_\rho > 0$, such that $\operatorname{ess\,inf}_{I_j} \rho_j(z) \geq c_\rho$ for all $j = 1, \dots, d$. We may assume without loss of generality that I_j is a bounded interval, and fix a constant $c_V > 0$ such that $I_j \subset [-c_V, c_V]$ for all $j = 1, \dots, d$. Let $V = \prod_{j=1}^d I_j \subset \mathbb{R}^d$. We now choose $B > 0$, such that $\mathrm{Prob}(\|\xi\|_{L^\infty} \leq B) > 0$. This is possible because of the assumed inclusion $\mathcal{X} \subset L^\infty$ and Lemma 6.3.2. For this choice of $B > 0$, we define the random event

$$\mathcal{K} := \{\|\xi\|_{L^\infty} \leq B\},$$

so that $\mathbb{P}(\mathcal{K}) > 0$. Note that this bound on B implies that for all $y \in V$ and $\xi \in K$, we have

$$\|u(\,\cdot\,; y, \xi)\|_{L^\infty(D)} \leq B + dc_V \max_{j=1,\dots,d} \|e_j\|_{L^\infty(D)} =: B'.$$

For fixed $\xi \in K$, define the maps $F_\xi^\dagger, F_\xi : \mathbb{R}^d \to \mathbb{R}^d$, as follows:

$$\begin{cases} F_\xi^\dagger(y) := \left\{ \fint_D u(x; y, \xi)\, e_k^*(x)\, \mathsf{d}x \right\}_{k=1}^d = y \\ F_\xi(y) := \fint_D R(u(x; y, \xi), x)\, \mathsf{d}x, \end{cases} \tag{6.3.26}$$

where R is a ReLU neural network mapping $\mathbb{R} \times \mathbb{R}^{d_D}$ to $\mathbb{R}^d$. Let $R^\dagger$ be defined by $R^\dagger(\eta, x) = \{\eta\, e_k^*(x)\}_{k=1}^d$. Then by Corollary E.1.2, for any $\epsilon > 0$, there exists a ReLU neural network R such that

$$\fint_D \|R(\cdot, x) - R^\dagger(\cdot, x)\|_{W^{1,\infty}([-B', B']; \mathbb{R}^d)}\, \mathsf{d}x \leq \epsilon. \tag{6.3.27}$$

Identify R with a ReLU neural network achieving this bound. Note that $F_\xi^\dagger \equiv \mathrm{id}$ is exactly the identity on $\mathbb{R}^d$ for all $\xi \in K$. Given the constant $\epsilon_0 > 0$ of Lemma 6.3.10, We seek to show that for sufficiently small $\epsilon > 0$ in (6.3.27), we can ensure that

$$\|F_\xi - F_\xi^\dagger\|_{W^{1,\infty}(V)} = \|F_\xi - \mathrm{id}\|_{W^{1,\infty}(V)} \leq \epsilon_0, \quad \forall\, \xi \in K. \tag{6.3.28}$$

By Lemma 6.3.10, this entails that there exist $V_0 \subset V$ and $c_0 > 0$, such that

$$F_{\#}(\mathrm{Unif}(Y) \otimes \mathbb{P}) \geq c_0 \mathbb{P}(K)\, \mathrm{Unif}(V_0),$$

where $\mathbb{P}$ denotes the law of ξ and $F(y, \xi) \coloneqq F_\xi(y)$. The claim then follows by observing that

$$\mathcal{E}_{\#}\mu = F_{\#}(\mu_d \otimes \mathbb{P}) \geq c_\rho^d\, F_{\#}(\mathrm{Unif}(Y) \otimes \mathbb{P}) \geq c_\rho^d c_0 \mathbb{P}(K)\, \mathrm{Unif}(V_0).$$

Since $\mathbb{P}(K) > 0$ by construction, the claim then follows with constant $c := c_\rho^d c_0 \mathbb{P}(K)$. It therefore remains to show that (6.3.27) for sufficiently small $\epsilon > 0$ implies (6.3.28).

In the remainder of this proof, we show that this holds for $\epsilon := \epsilon_0/2d$. By (6.3.27), we have $\|\partial_\eta (R - R^\dagger)(\cdot, x)\|_{L^\infty([-B', B'])} < \epsilon$, where η refers to the first argument of R and $R^\dagger$. Then for $\xi \in K$ and $y, y' \in V$, we have

$$\|u(\,\cdot\,; y, \xi)\|_{L^\infty(D)},\ \|u(\,\cdot\,; y, \xi)\|_{L^\infty(D)} \leq B',$$

and hence

$$\|F_\xi - F_\xi^\dagger\|_{L^\infty(V)} \leq \fint_D \|R(\,\cdot\,, x) - R^\dagger(\,\cdot\,, x)\|_{L^\infty([-B', B'])}\, \mathsf{d}x \leq \epsilon = \epsilon_0/2d.$$

To estimate $\|DF_\xi - DF_\xi^\dagger\|_{L^\infty(V)}$, we recall that, due to the convexity of the d-dimensional cube V, the $W^{1,\infty}(V)$ seminorm is equal to the Lipschitz seminorm:

$$\|DF_\xi\|_{L^\infty(V)} = \sup_{y, y' \in V} \frac{|F_\xi(y) - F_\xi(y')|}{|y - y'|}.$$

We now bound, for $y, y' \in V$:

$$\begin{aligned}
&|(F_\xi - F_\xi^\dagger)(y) - (F_\xi - F_\xi^\dagger)(y')| \\
&\qquad = \left| \fint_D (R - R^\dagger)(u(x; y, \xi), x) - (R - R^\dagger)(u(x; y', \xi), x)\, \mathsf{d}x \right| \\
&\qquad \leq \fint_D \left| (R - R^\dagger)(u(x; y, \xi), x) - (R - R^\dagger)(u(x; y', \xi), x) \right| \mathsf{d}x \\
&\qquad \leq \fint_D |u(x; y, \xi) - u(x; y', \xi)| \|\partial_\eta (R - R^\dagger)(\cdot, x)\|_{L^\infty([-B', B'])}\, \mathsf{d}x \\
&\qquad = \fint_D \left| \sum_{j=1}^d (y_j - y_j') e_j(x) \right| \|\partial_\eta (R - R^\dagger)(\cdot, x)\|_{L^\infty([-B', B'])}\, \mathsf{d}x \\
&\qquad \leq \fint_D \sum_{j=1}^d |y_j - y_j'| \|\partial_\eta (R - R^\dagger)(\cdot, x)\|_{L^\infty([-B', B'])}\, \mathsf{d}x \\
&\qquad \leq \sqrt{d} |y - y'| \epsilon.
\end{aligned}$$

With our choice of $\epsilon = \epsilon_0/2d$, this implies that

$$\|DF_\xi - DF_\xi^\dagger\|_{L^\infty(V)} = \sup_{y,y' \in V} \frac{|(F_\xi - F_\xi^\dagger)(y) - (F_\xi - F_\xi^\dagger)(y')|}{|y - y'|} \leq \epsilon\sqrt{d} \leq \epsilon_0/2.$$

Combining both estimates, we have shown that

$$\fint_D \|R(\,\cdot\,, x) - R^\dagger(\,\cdot\,; x)\|_{W^{1,\infty}([-B',B'])}\, \mathsf{d}x \leq \epsilon,$$

for $\epsilon = \epsilon_0/2d$, implies

$$\|F_\xi - F_\xi^\dagger\|_{W^{1,\infty}(V)} \leq \epsilon_0.$$

This is what we set out to show, and concludes our proof of Proposition 6.3.19. □

Theory-to-Practice Gap We can now state a theory-to-practice gap for the unit ball $\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{NO}}$ in the NO approximation space $\mathbf{A}^{\alpha,\infty}_{\ell,\mathrm{NO}}$:

***Theorem* 6.3.20** (NO theory-to-practice gap)**.** Let $p \in [1,\infty]$. Let $\ell : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ be non-decreasing with $\ell^* \geq 4$. Let $\mathcal{X}(D) \subset L^\infty(D)$ be a Banach space on Lipschitz domain $D \subset \mathbb{R}^{d_D}$. Assume that $\mu \in \mathcal{P}(\mathcal{X})$ satisfies Assumption 6.3.1 with bi-orthogonal elements $\{e_j^*\}_{j\in\mathbb{N}} \subset L^1(D)$. Then for any $\alpha > 0$, we have

$$\beta_*(\mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{NO}}, L^p(\mu)) \leq \frac{1}{p}. \tag{6.3.29}$$

◊

The proof of Theorem 6.3.20 relies on the following lemma:

***Lemma* 6.3.21.** Let $\mathcal{X}(D) \subset L^\infty(D)$ be a Banach space on Lipschitz domain $D \subset \mathbb{R}^{d_D}$, and let $\mu \in \mathcal{P}(\mathcal{X})$ be a probability measure on $\mathcal{X}$. Assume that $\mathcal{Y}(D)$ contains all constant functions and μ satisfies Assumption 6.3.1[(d)] for $d \in \mathbb{N}$, with bi-orthogonal elements $e_1^*, \ldots, e_d^* \in L^1(D)$. Let $\mathcal{E} : \mathcal{X}(D) \to \mathbb{R}^d$ be the encoder of Proposition 6.3.19. Fix a (finite) constant $\ell_0 \leq \ell^* + 2$. There exists a constant $\gamma = \gamma(d, \mathcal{E}, \alpha, \ell_0) > 0$, such that

$$\left\{ f \circ \mathcal{E} \,\middle|\, f \in U^{\alpha,\infty}_{\ell_0}(\mathbb{R}^d) \right\} \subset \gamma \cdot \mathbf{U}^{\alpha,\infty}_{\ell,\mathrm{NO}}. \tag{6.3.30}$$

◊

A proof of this lemma is given in Appendix E.2. We now come to the proof of Theorem 6.3.20.

Proof of Theorem 6.3.20. Fix $d \in \mathbb{N}$, and let $\ell : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ be a non-decreasing function. Clearly, we have $\ell^* \geq 1$. We define $\ell_0 := 3$, so that the *finite dimensional* theory-to-practice gap, Theorem 6.2.2 applies to $U_{\ell_0}^{\alpha,\infty}(\mathbb{R}^d)$. Let $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$ be an encoder as in Proposition 6.3.19, with $\mathcal{E}_{\#}\mu \geq c\,\mathrm{Unif}([0,1]^d)$. By (6.3.30), there exists a constant $\gamma > 0$, such that

$$\left\{ f \circ \mathcal{E} \,\middle|\, f \in U_{\ell_0}^{\alpha,\infty}(\mathbb{R}^d) \right\} \subset \gamma \cdot \mathbf{U}_{\ell,\mathrm{NO}}^{\alpha,\infty}.$$

The claim of Theorem 6.3.20 then follows again from Proposition 6.3.6, as in the proof of Theorem 6.3.12. Indeed, Proposition 6.3.6, and the fact that $\beta_*(\gamma \cdot \mathbf{U}_{\ell,\mathrm{NO}}^{\alpha,\infty}, L^p(\mu)) = \beta_*(\mathbf{U}_{\ell,\mathrm{NO}}^{\alpha,\infty}, L^p(\mu))$, imply that

$$\beta_*(\mathbf{U}_{\ell,\mathrm{NO}}^{\alpha,\infty}, L^p(\mu)) \leq \frac{1}{p} + \frac{1}{d} \cdot \frac{\alpha}{\alpha + \lfloor \ell_0/2 \rfloor}.$$

Since the left-hand side is independent of d, we let $d \to \infty$, to obtain (6.3.29). □

We also state the following theory-to-practice gap for uniform approximation over compact $\mathcal{K}$.

***Theorem* 6.3.22** (NO; uniform theory-to-practice gap)**.** Let $\ell : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ be non-decreasing with $\ell^* \geq 4$. Let $\mathcal{X}(D) \subset L^\infty(D)$ be a Banach space on Lipschitz domain $D \subset \mathbb{R}^{d_D}$. Assume that $\mathcal{K} \subset \mathcal{X}$ is a compact set satisfying Assumption 6.3.3. Then for any $\alpha > 0$, we have

$$\beta_*(\mathbf{U}_{\ell,\mathrm{NO}}^{\alpha,\infty}, C(\mathcal{K})) = 0.$$

◊

Proof. The proof is analogous to the argument for the uniform theory-to-practice gap for DeepONet, except that the DeepONet encoder construction, Proposition 6.3.11, is replaced by the NO encoder construction in 6.3.19, and the relevant inclusion is the one identified in Lemma 6.3.21. □

6.4 Conclusion

In conclusion, this work has rigorously examined the theory-to-practice gap in both finite-dimensional and infinite-dimensional settings, resulting in rigorous bounds on achievable convergence rates for general reconstruction methods based on point-values. By deriving upper bounds on the optimal rate β_*, we have uncovered the inherent constraints of learning on relevant neural network and neural operator approximation spaces. In the finite-dimensional case, our contributions include a unified treatment of the theory-to-practice gap for approximation errors measured in general L^p-spaces for arbitrary $p \in [1, \infty]$ and dimension $d \in \mathbb{N}$. Furthermore, we extend the theory-to-practice gap to infinite-dimensional operator learning frameworks, and derive results for prominent architectures such as Deep Operator Networks and integral kernel-based neural operators, such as the Fourier neural operator (FNO). Notably, for operator learning we establish that the optimal convergence rate in a Bochner L^p-norm satisfies $\beta_* \leq 1/p$, while no algebraic convergence is possible ($\beta_* = 0$) for uniform approximation on infinite-dimensional compact input sets. These findings highlight some intrinsic limitations of these data-driven methodologies and provide a clearer understanding of the theoretical bounds shaping practical applications.

There are several interesting avenues for future work, two of which we briefly mention in closing. One open problem is to study the theory-to-practice gap under additional constraints, e.g. on spaces of the form $\mathbf{A}_\ell^{\alpha,\infty} \cap \mathrm{Lip}$. We expect that the theory-to-practice gap will persist essentially unchanged even when introducing additional regularity constraints. Another open problem is to extend this gap beyond ReLU activations. This is specifically relevant for operator learning, where popular implementations of e.g. FNO usually use a smooth variant of ReLU, such as GeLU. To date, even in finite dimensions, no theory-to-practice gap is known for such smooth activation functions. Since our proofs rely on the homogeneity of ReLU, the path to such an extension is not immediately obvious.

BIBLIOGRAPHY

[1] Thomas G Dietterich. "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15.

[2] Allan David Gordon. *Classification*. CRC Press, 1999.

[3] Todd K Moon. "The expectation-maximization algorithm". In: *IEEE Signal processing magazine* 13.6 (1996), pp. 47–60.

[4] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[5] Waseem Rawat and Zenghui Wang. "Deep convolutional neural networks for image classification: A comprehensive review". In: *Neural computation* 29.9 (2017), pp. 2352–2449.

[6] Li Deng, Geoffrey Hinton, and Brian Kingsbury. "New types of deep neural network learning for speech recognition and related applications: An overview". In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 8599–8603.

[7] Gerald Tesauro et al. "Temporal difference learning and TD-Gammon". In: *Communications of the ACM* 38.3 (1995), pp. 58–68.

[8] Maziar Raissi. "Deep hidden physics models: Deep learning of nonlinear partial differential equations". In: *Journal of Machine Learning Research* 19.25 (2018), pp. 1–24.

[9] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational physics* 378 (2019), pp. 686–707.

[10] Franco Scarselli and Ah Chung Tsoi. "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results". In: *Neural networks* 11.1 (1998), pp. 15–37.

[11] Andrew R Barron. "Approximation and estimation bounds for artificial neural networks". In: *Machine learning* 14 (1994), pp. 115–133.

[12] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. "Visualizing the loss landscape of neural nets". In: *Advances in neural information processing systems* 31 (2018).

[13] William D Fries, Xiaolong He, and Youngsoo Choi. "Lasdi: Parametric latent space dynamics identification". In: *Computer Methods in Applied Mechanics and Engineering* 399 (2022), p. 115436.

[14] Kévin Garanger, Julie Kraus, and Julian J Rimoli. "Symmetry-enforcing neural networks with applications to constitutive modeling". In: *Extreme Mechanics Letters* 71 (2024), p. 102188.

[15] Kaushik Bhattacharya, Burigede Liu, Andrew M. Stuart, and Margaret Trautner. "Learning Markovian homogenized models in viscoelasticity". In: *Multiscale Modeling & Simulation* 21.2 (2023), pp. 641–679. DOI: 10.1137/22M149920.

[16] Burigede Liu, Eric Ocegueda, Margaret Trautner, Andrew M. Stuart, and Kaushik Bhattacharya. "Learning macroscopic internal variables and history dependence from microscopic models". In: *Journal of the Mechanics and Physics of Solids* 178 (2023), p. 105329. DOI: 10.1016/j.jmps.2023.105329.

[17] James R Rice. "Inelastic constitutive relations for solids: an internal-variable theory and its application to metal plasticity". In: *Journal of the Mechanics and Physics of Solids* 19.6 (1971), pp. 433–455.

[18] Kaushik Bhattacharya, Lianghao Cao, George Stepaniants, Andrew Stuart, and Margaret Trautner. *Learning Memory and Material Dependent Constitutive Laws*. 2025. arXiv: 2502.05463 [math.NA].

[19] Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. "Fourier neural operator for parametric partial differential equations". In: *International Conference on Learning Representations* (2021).

[20] Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. "On universal approximation and error bounds for Fourier neural operators". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 13237–13312.

[21] Daniel Zhengyu Huang, Nicholas H. Nelsen, and Margaret Trautner. "An operator learning perspective on parameter-to-observable maps". In: *Foundations of Data Science* 7.1 (2025), pp. 163–225. DOI: 10.3934/fods.2024037.

[22] Philipp Grohs and Felix Voigtlaender. "Proof of the theory-to-practice gap in deep learning via sampling complexity bounds for neural network approximation spaces". In: *Foundations of Computational Mathematics* (2023), pp. 1–59.

[23] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators". In: *Nature machine intelligence* 3.3 (2021), pp. 218–229.

[24] Juan C Simo. "Numerical analysis and simulation of plasticity". In: *Handbook of numerical analysis* 6 (1998), pp. 183–499.

[25] Juan C Simo and Thomas JR Hughes. *Computational inelasticity*. Vol. 7. Springer Science & Business Media, 2006.

[26] Alain Bensoussan, Jacques-Louis Lions, and George Papanicolaou. *Asymptotic analysis for periodic structures*. Vol. 374. American Mathematical Society, 2011.

[27] Grigoris Pavliotis and Andrew M. Stuart. *Multiscale methods: averaging and homogenization*. Springer Science & Business Media, 2008.

[28] Gilles A Francfort and Pierre M Suquet. "Homogenization and mechanical dissipation in thermoviscoelasticity". In: *Archive for Rational Mechanics and Analysis* 96.3 (1986), pp. 265–293.

[29] Burigede Liu, Nikola B. Kovachki, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, Andrew M. Stuart, and Kaushik Bhattacharya. "A learning-based multiscale method and its application to inelastic impact problems". In: *Journal of the Mechanics and Physics of Solids* 158, 104668 (2022).

[30] Morton E Gurtin, Eliot Fried, and Lallit" Anand. *The Mechanics and Thermodynamics of Continua*. Cambridge University Press, 2010.

[31] A.J.M. Spencer. *Continuum Mechanics*. Essex: Longman Group U.K., 1980.

[32] Oscar Gonzalez and Andrew M. Stuart. *A first course in continuum mechanics*. Vol. 42. Cambridge University Press, 2008.

[33] Richard C. Christensen. *Theory of Viscoelasticity*. Academic Press, 1982.

[34] Roderic S. Lakes. *Viscoelastic Solids*. C.R.C Press, 1998.

[35] Jacob Fish. *Multiscale methods: bridging the scales in science and engineering*. Oxford University Press on Demand, 2010.

[36] Ellad B. Tadmor. *Modeling Materials: Continuum, Atomistic and Multiscale Techniques*. Cambridge University Press, 2012.

[37] Erik Van Der Giessen, Peter A Schultz, Nicolas Bertin, Vasily V Bulatov, Wei Cai, Gábor Csányi, Stephen M Foiles, Marc GD Geers, Carlos González, Markus Hütter, et al. "Roadmap on multiscale materials modeling". In: *Modelling and Simulation in Materials Science and Engineering* 28.4 (2020), p. 043001.

[38] Rob Phillips. *Crystals, defects and microstructures: modeling across scales*. Cambridge University Press, 2001.

[39] Enrique Sánchez-Palencia. "Non-homogeneous media and vibration theory". In: *Lecture notes in physics* 127 (1980).

[40] Doina Cioranescu and Patrizia Donato. *An Introduction To Homogenization*. Vol. 17. Oxford university press Oxford, 1999.

[41] Grégoire Allaire. "Homogenization and two-scale convergence". In: *SIAM Journal on Mathematical Analysis* 23.6 (1992), pp. 1482–1518.

[42] Graeme W. Milton. *The Theory of Composites*. Cambridge University Press, 2002.

[43] Renald Brenner and Pierre Suquet. "Overall response of viscoelastic composites and polycrystals: exact asymptotic relations and approximate estimates". In: *International Journal of Solids and Structures* 50.10 (2013), pp. 1824–1838.

[44] Luc Tartar. "Memory effects and homogenization". In: *Archive for Rational Mechanics and Analysis* 111 (1990), pp. 121–133.

[45] T I Zohdi and P Wriggers. *Introduction to computational micromechanics*. Springer Verlag, 2005.

[46] Hervé Moulinec and Pierre Suquet. "A numerical method for computing the overall response of nonlinear composites with complex microstructure". In: *Computer methods in Applied Mechanics and Engineering* 157.1-2 (1998), pp. 69–94.

[47] N. Mishra, J. Vondřejc, and J. Zeman. "A comparative study on low-memory iterative solvers for FFT-based homogenization of periodic media". In: *Journal of Computational Physics* 321 (2016), pp. 151–168.

[48] H. Moulinec, P. Suqeut, and G. Milton. "Convergence of iterative methods based on Neumann series for composite materials: Theory and practice". In: *International Journal of Numerical Methods in Engineering* 114 (2018), pp. 1103–1130.

[49] M Mozaffar, R Bostanabad, W Chen, K Ehmann, Jian Cao, and MA Bessa. "Deep learning predicts path-dependent plasticity". In: *Proceedings of the National Academy of Sciences* 116.52 (2019), pp. 26414–26420.

[50] L. Wu, N.G. Kilingar, and L. Noels. "A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths". In: *Computer Methods in Applied Mechanics and Engineering* 369 (2020), p. 113234.

[51] Faisal As' ad, Philip Avery, and Charbel Farhat. "A mechanics-informed artificial neural network approach in data-driven constitutive modeling". In: *International Journal for Numerical Methods in Engineering* 123.12 (2022), pp. 2738–2759.

[52] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. "Neural operator: Learning maps between function spaces with applications to pdes". In: *Journal of Machine Learning Research* 24.89 (2023), pp. 1–97.

[53] F Ghavamian and A Simone. "Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network". In: *Computer Methods in Applied Mechanics and Engineering* 357 (2019), p. 112594.

[54] Zeliang Liu, CT Wu, and M391298807188754 Koishi. "A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials". In: *Computer Methods in Applied Mechanics and Engineering* 345 (2019), pp. 1138–1168.

[55] Dana Bishara, Yuxi Xie, Wing Kam Liu, and Shaofan Li. "A state-of-the-art review on machine learning-based multiscale modeling, simulation, homogenization and design of materials". In: *Archives of computational methods in engineering* 30.1 (2023), pp. 191–222.

[56] Kaushik Bhattacharya, Nikola B. Kovachki, Aakila Rajan, Andrew M. Stuart, and Margaret Trautner. "Learning homogenization for elliptic operators". In: *SIAM Journal on Numerical Analysis* 62.4 (2024), pp. 1844–1873. DOI: 10.1137/23M1585015.

[57] Giovanni Leoni. *A first course in Sobolev spaces*. American Mathematical Soc., 2017.

[58] Enrique Sanchez-Palencia and André Zaoui. "Homogenization techniques for composite media". In: *Homogenization techniques for composite media* 272 (1987).

[59] Xavier Blanc and Claude Le Bris. *Homogenization Theory for Multiscale Problems: An Introduction*. Vol. 21. Springer Nature, 2023.

[60] Morton E Gurtin. *An Introduction To Continuum Mechanics*. Academic press, 1982.

[61] Thomas Y Hou and Xiao-Hui Wu. "A multiscale finite element method for elliptic problems in composite materials and porous media". In: *Journal of computational physics* 134.1 (1997), pp. 169–189.

[62] Andrea Bonito, Ronald A DeVore, and Ricardo H Nochetto. "Adaptive finite element methods for elliptic problems with discontinuous coefficients". In: *SIAM Journal on Numerical Analysis* 51.6 (2013), pp. 3106–3134.

[63] Ricardo H Nochetto, Kunibert G Siebert, and Andreas Veeser. "Theory of adaptive finite element methods: an introduction". In: *Multiscale, Nonlinear and Adaptive Approximation: Dedicated to Wolfgang Dahmen on the Occasion of his 60th Birthday*. Springer. 2009, pp. 409–542.

[64] Houman Owhadi and Lei Zhang. "Numerical homogenization of the acoustic wave equations with a continuum of scales". In: *Computer Methods in Applied Mechanics and Engineering* 198.3-4 (2008), pp. 397–406.

[65] Albert Cohen, Ronald DeVore, and Christoph Schwab. "Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs". In: *Foundations of Computational Mathematics* 10.6 (2010), pp. 615–646.

[66] Albert Cohen, Ronald Devore, and Christoph Schwab. "Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's". In: *Analysis and Applications* 9.01 (2011), pp. 11–47.

[67] Abdellah Chkifa, Albert Cohen, Ronald DeVore, and Christoph Schwab. "Sparse adaptive Taylor approximation algorithms for parametric and stochastic elliptic PDEs". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 47.1 (2013), pp. 253–280.

[68] Kaushik Bhattacharya, Bamdad Hosseini, Nikola B Kovachki, and Andrew M. Stuart. "Model reduction and neural networks for parametric PDEs". In: *The SMAI journal of computational mathematics* 7 (2021), pp. 121–157.

[69] Nicholas H Nelsen and Andrew M. Stuart. "The random feature model for input-output maps between Banach spaces". In: *SIAM Journal on Scientific Computing* 43.5 (2021), A3212–A3243.

[70] Pau Batlle, Matthieu Darcy, Bamdad Hosseini, and Houman Owhadi. "Kernel methods are competitive for operator learning". In: *Journal of Computational Physics* 496 (2024), p. 112549.

[71] Carlo Marcati and Christoph Schwab. "Exponential convergence of deep operator networks for elliptic partial differential equations". In: *SIAM Journal on Numerical Analysis* 61.3 (2023), pp. 1513–1545.

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[73] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[74] Kaushik Bhattacharya. "Phase boundary propagation in a heterogeneous body". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 455.1982 (1999), pp. 757–766.

[75] Xin Liu, Su Tian, Fei Tao, and Wenbin Yu. "A review of artificial neural networks in the constitutive modeling of composite materials". In: *Composites Part B: Engineering* 224 (2021), p. 109152.

[76] Jan N Fuhg and Nikolaos Bouklas. "On physics-informed data-driven isotropic and anisotropic constitutive models through probabilistic machine learning and space-filling sampling". In: *Computer Methods in Applied Mechanics and Engineering* 394 (2022).

[77] Alexandre M Tartakovsky, C Ortiz Marrero, Paris Perdikaris, Guzel D Tartakovsky, and David Barajas-Solano. "Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems". In: *Water Resources Research* 56.5 (2020), e2019WR026731.

[78] Pingchuan Ma, Peter Yichen Chen, Bolei Deng, Joshua B Tenenbaum, Tao Du, Chuang Gan, and Wojciech Matusik. "Learning neural constitutive laws from motion observations for generalizable pde dynamics". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 23279–23300.

[79] Ehsan Haghighat, Sahar Abouali, and Reza Vaziri. "Constitutive model characterization and discovery using physics-informed deep learning". In: *Engineering Applications of Artificial Intelligence* 120 (2023), p. 105828.

[80] Kevin Linka, Markus Hillgärtner, Kian P Abdolazizi, Roland C Aydin, Mikhail Itskov, and Christian J Cyron. "Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning". In: *Journal of Computational Physics* 429 (2021), p. 110010.

[81] Kailai Xu, Daniel Z Huang, and Eric Darve. "Learning constitutive relations using symmetric positive definite neural networks". In: *Journal of Computational Physics* 428 (2021), p. 110072.

[82] Daniel Z Huang, Kailai Xu, Charbel Farhat, and Eric Darve. "Learning constitutive relations from indirect observations using deep neural networks". In: *Journal of Computational Physics* 416 (2020), p. 109491.

[83] Jihun Han and Yoonsang Lee. "A neural network approach for homogenization of multiscale problems". In: *Multiscale Modeling & Simulation* 21.2 (2023), pp. 716–734.

[84] Hernan J Logarzo, German Capuano, and Julian J Rimoli. "Smart constitutive laws: Inelastic homogenization through machine learning". In: *Computer methods in applied mechanics and engineering* 373 (2021), p. 113482.

[85] Huaiqian You, Yue Yu, Stewart Silling, and Marta D'Elia. "Data-driven learning of nonlocal models: from high-fidelity simulations to constitutive laws". In: *CEUR-WS* 2964.177 (2021).

[86] Xin Liu, Fei Tao, and Wenbin Yu. "A neural network enhanced system for learning nonlinear constitutive law and failure initiation criterion of composites using indirectly measurable data". In: *Composite Structures* 252 (2020), p. 112658.

[87] Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. "Error estimates for DeepONets: A deep learning framework in infinite dimensions". In: *Transactions of Mathematics and Its Applications* 6.1 (2022), tnac001. eprint: 2102.09618.

[88] Carlo Marcati, Joost AA Opschoor, Philipp C Petersen, and Christoph Schwab. "Exponential relu neural network approximation rates for point and edge singularities". In: *Foundations of Computational Mathematics* (2022), pp. 1–85.

[89] Lukas Herrmann, Christoph Schwab, and Jakob Zech. *Neural and GPC operator surrogates: construction and expression rate bounds*. 2022. arXiv: 2207.04950 [math.NA].

[90] Joost AA Opschoor, Ch Schwab, and Jakob Zech. "Exponential ReLU DNN expression of holomorphic maps in high dimension". In: *Constructive Approximation* 55.1 (2022), pp. 537–582.

[91] R.A. DeVore and G.G. Lorentz. *Constructive Approximation*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 1993.

[92] Richard Hill. "The elastic behaviour of a crystalline aggregate". In: *Proceedings of the Physical Society. Section A* 65.5 (1952), p. 349.

[93] Maarten V de Hoop, Daniel Zhengyu Huang, Elizabeth Qian, and Andrew M. Stuart. "The cost-accuracy trade-off in operator learning with neural networks". In: *Journal of Machine Learning* 1.3 (2022), pp. 299–341.

[94] Samuel Lanthaler, Zongyi Li, and Andrew M. Stuart. *The nonlocal neural operator: universal approximation*. 2023. arXiv: 2304.13221 [cs.LG].

[95] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[96] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. *Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators*. 2022. arXiv: 2202.11214 [cs.LG].

[97] Tingtao Zhou, Xuan Wan, Daniel Zhengyu Huang, Zongyi Li, Zhiwei Peng, Anima Anandkumar, John F. Brady, Paul W. Sternberg, and Chiara Daraio. "AI-aided geometric design of anti-infection catheters". In: *Science Advances* 10.1, eadj1741 (2024).

[98] Jürgen Moser. "A rapidly convergent iteration method and non-linear partial differential equations-I". In: *Annali della Scuola Normale Superiore di Pisa-Scienze Fisiche e Matematiche* 20.2 (1966), pp. 265–315.

[99] Dan Hendrycks and Kevin Gimpel. *Gaussian error linear units (gelus)*. 2016. arXiv: 1606.08415 [cs.LG].

[100] Lukas Herrmann, Joost A. A. Opschoor, and Christoph Schwab. "Constructive deep ReLU neural network approximation". In: *Journal of Scientific Computing* 90.2, 75 (2022).

[101] Lukas Herrmann, Christoph Schwab, and Jakob Zech. "Deep neural network expression of posterior expectations in Bayesian PDE inversion". In: *Inverse Problems* 36.12 (2020), p. 125011.

[102] Lukas Herrmann, Christoph Schwab, and Jakob Zech. "Neural and spectral operator surrogates: unified construction and expression rate bounds". In: *Advances in Computational Mathematics* 50.4 (2024), p. 72.

[103] Marcello Longo, Joost AA Opschoor, Nico Disch, Christoph Schwab, and Jakob Zech. "De Rham compatible deep neural network FEM". In: *Neural Networks* 165 (2023), pp. 721–739.

[104] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. "Fourier neural operator with learned deformations for pdes on general geometries". In: *Journal of Machine Learning Research* 24.388 (2023), pp. 1–26.

[105] Nikola B Kovachki, Samuel Lanthaler, and Andrew M. Stuart. *Operator Learning: Algorithms and Analysis*. 2024. arXiv: 2402.15715 [cs.LG].

[106] Francesca Bartolucci, Emmanuel de Bezenac, Bogdan Raonic, Roberto Molinaro, Siddhartha Mishra, and Rima Alaifari. "Representation equivalent neural operators: a framework for alias-free operator learning". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 69661–69672.

[107] Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. "Convolutional neural operators for robust and accurate learning of PDEs". In: *Advances in Neural Information Processing Systems* 36 (2024).

[108] Robert Joseph George, Jiawei Zhao, Jean Kossaifi, Zongyi Li, and Anima Anandkumar. *Incremental Spatial and Spectral Learning of Neural Operators for Solving Large-Scale PDEs*. 2022. arXiv: 2211.15188 [cs.LG].

[109] Ying Shi Teh, Swarnava Ghosh, and Kaushik Bhattacharya. "Machine-learned prediction of the electronic fields in a crystal". In: *Mechanics of Materials* 163, 104070 (2021).

[110] Jan-Hendrik Bastek and Dennis M Kochmann. "Inverse-design of nonlinear mechanical metamaterials via video denoising diffusion models". In: *Nature Machine Intelligence* 5 (2023), pp. 1466–1475.

[111] Anima Anandkumar, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Nikola Kovachki, Zongyi Li, Burigede Liu, and Andrew Stuart. "Neural Operator: Graph Kernel Network for Partial Differential Equations". In: *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*. 2019.

[112] Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew M. Stuart, Kaushik Bhattacharya, and Anima Anandkumar. "Multipole graph neural operator for parametric partial differential equations". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 6755–6766.

[113] Ravi G Patel, Nathaniel A Trask, Mitchell A Wood, and Eric C Cyr. "A physics-informed operator regression framework for extracting data-driven continuum models". In: *Computer Methods in Applied Mechanics and Engineering* 373, 113500 (2021).

[114] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. "Fourier neural operator with learned deformations for PDEs on general geometries". In: *Journal of Machine Learning Research* 24.388 (2023), pp. 1–26.

[115] Levi Lingsch, Mike Y. Michelis, Emmanuel De Bézenac, Sirani M. Perera, Robert K. Katzschmann, and Siddhartha Mishra. "Beyond regular grids: Fourier-based neural operators on arbitrary domains". In: *Proceedings of the 41st International Conference on Machine Learning*. ICML'24. JMLR.org, 2024.

[116] Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar. "Fourcastnet: Accelerating global high-resolution

weather forecasting using adaptive fourier neural operators". In: *Proceedings of the platform for advanced scientific computing conference*. 2023, pp. 1–11.

[117] Somdatta Goswami, Minglang Yin, Yue Yu, and George Em Karniadakis. "A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials". In: *Computer Methods in Applied Mechanics and Engineering* 391, 114587 (2022).

[118] Tianping Chen and Hong Chen. "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems". In: *IEEE transactions on neural networks* 6.4 (1995), pp. 911–917.

[119] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2 (1991), pp. 251–257.

[120] Weinan E, Chao Ma, and Lei Wu. "The Barron space and the flow-induced function spaces for neural network models". In: *Constructive Approximation* 55.1 (2022), pp. 369–406.

[121] Nicolas Boullé, Diana Halikias, and Alex Townsend. "Elliptic PDE learning is provably data-efficient". In: *Proceedings of the National Academy of Sciences* 120.39, e2303904120 (2023).

[122] Beichuan Deng, Yeonjong Shin, Lu Lu, Zhongqiang Zhang, and George Em Karniadakis. "Approximation rates of DeepONets for learning operators arising from advection–diffusion equations". In: *Neural Networks* 153 (2022), pp. 411–426.

[123] Samuel Lanthaler. "Operator learning with PCA-Net: upper and lower complexity bounds". In: *Journal of Machine Learning Research* 24.318 (2023), pp. 1–67.

[124] Samuel Lanthaler, Roberto Molinaro, Patrik Hadorn, and Siddhartha Mishra. "Nonlinear reconstruction for operator learning of PDEs with discontinuities". In: *The Eleventh International Conference on Learning Representations*. 2022.

[125] Samuel Lanthaler and Andrew M. Stuart. *The parametric complexity of operator learning*. 2023. arXiv: 2306.15924 [cs.LG].

[126] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Deep network approximation for smooth functions". In: *SIAM Journal on Mathematical Analysis* 53.5 (2021), pp. 5465–5506.

[127] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Deep network approximation characterized by number of neurons". In: *Communications in Computational Physics* 28.5 (2020), pp. 1768–1811.

[128] Andrea Caponnetto and Ernesto De Vito. "Optimal rates for the regularized least-squares algorithm". In: *Foundations of Computational Mathematics* 7 (2007), pp. 331–368.

[129] Maarten V de Hoop, Nikola B Kovachki, Nicholas H Nelsen, and Andrew M. Stuart. "Convergence rates for learning linear operators from noisy data". In: *SIAM/ASA Journal on Uncertainty Quantification* 11.2 (2023), pp. 480–513.

[130] Jikai Jin, Yiping Lu, Jose Blanchet, and Lexing Ying. "Minimax optimal kernel operator learning via multilevel training". In: *The Eleventh International Conference on Learning Representations*. 2022.

[131] Samuel Lanthaler and Nicholas H. Nelsen. "Error bounds for learning with vector-valued random features". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 71834–71861.

[132] Zhu Li, Dimitri Meunier, Mattes Mollenhauer, and Arthur Gretton. "Towards optimal Sobolev norm rates for the vector-valued regularized least-squares algorithm". In: *Journal of Machine Learning Research* 25 (2024), pp. 1–51.

[133] Hao Liu, Haizhao Yang, Minshuo Chen, Tuo Zhao, and Wenjing Liao. "Deep nonparametric estimation of operators between infinite dimensional spaces". In: *Journal of Machine Learning Research* 25.24 (2024), pp. 1–67.

[134] Mattes Mollenhauer, Nicole Mücke, and T. J. Sullivan. *Learning linear operators: Infinite-dimensional regression as a well-behaved non-compact inverse problem*. 2022. arXiv: 2211.08875 [math.ST].

[135] Florian Schäfer and Houman Owhadi. "Sparse recovery of elliptic solvers from matrix-vector products". In: *SIAM Journal on Scientific Computing* 46.2 (2024), A998–A1025.

[136] George Stepaniants. "Learning partial differential equations in reproducing kernel Hilbert spaces". In: *Journal of Machine Learning Research* 24.86 (2023), pp. 1–72.

[137] Thomas Laurent and James Brecht. "Deep linear networks with arbitrary loss: All local minima are global". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2902–2907.

[138] Nicholas H. Nelsen and Andrew M. Stuart. "Operator learning using random features: a tool for scientific computing". In: *SIAM Review* 66.3 (2024).

[139] Zhongjie Shi, Jun Fan, Linhao Song, Ding-Xuan Zhou, and Johan A. K. Suykens. *Nonlinear functional regression by functional deep neural network with kernel embedding*. 2024. arXiv: 2401.02890 [stat.ML].

[140] Md Ashiqur Rahman, Manuel A. Florez, Anima Anandkumar, Zachary E. Ross, and Kamyar Azizzadenesheli. "Generative adversarial neural operators". In: *Transactions on Machine Learning Research* (2022).

[141] Zezhong Zhang, Feng Bao, and Guannan Zhang. "Improving the expressive power of deep neural networks through integral activation transform". In: *International Journal of Numerical Analysis and Modeling* 21.5 (2024), pp. 739–763.

[142] Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. "PDEBench: An extensive benchmark for scientific machine learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 1596–1611.

[143] Khemraj Shukla, Vivek Oommen, Ahmad Peyvan, Michael Penwarden, Nicholas Plewacki, Luis Bravo, Anindya Ghoshal, Robert M. Kirby, and George Em Karniadakis. "Deep neural operators as accurate surrogates for shape optimization". In: *Engineering Applications of Artificial Intelligence* 129, 107615 (2024).

[144] Thomas O'Leary-Roseberry, Xiaosong Du, Anirban Chaudhuri, Joaquim R. R. A. Martins, Karen Willcox, and Omar Ghattas. "Learning high-dimensional parametric maps via reduced basis adaptive residual networks". In: *Computer Methods in Applied Mechanics and Engineering* 402, 115730 (2022).

[145] Kjetil O Lye, Siddhartha Mishra, and Roberto Molinaro. "A multi-level procedure for enhancing accuracy of machine learning algorithms". In: *European Journal of Applied Mathematics* 32.3 (2021), pp. 436–469.

[146] Kjetil O. Lye, Siddhartha Mishra, and Deep Ray. "Deep learning observables in computational fluid dynamics". In: *Journal of Computational Physics* 410, 109339 (2020).

[147] Kjetil O. Lye, Siddhartha Mishra, Deep Ray, and Praveen Chandrashekar. "Iterative surrogate model optimization (ISMO): An active learning algorithm for PDE constrained optimization with deep neural networks". In: *Computer Methods in Applied Mechanics and Engineering* 374, 113575 (2021).

[148] Siddhartha Mishra and T. Konstantin Rusch. "Enhancing accuracy of deep learning algorithms by training with low-discrepancy sequences". In: *SIAM Journal on Numerical Analysis* 59.3 (2021), pp. 1811–1834.

[149] Huaiqian You, Quinn Zhang, Colton J. Ross, Chung-Hao Lee, and Yue Yu. "Learning deep implicit Fourier neural operators (IFNOs) with applications to heterogeneous material modeling". In: *Computer Methods in Applied Mechanics and Engineering* 398, 115296 (2022).

[150] Samuel Lanthaler, Andrew M. Stuart, and Margaret Trautner. *Discretization error of Fourier neural operators*. 2024. arXiv: 2405.02221 [math.NA].

[151] Erisa Hasani and Rachel A Ward. *Generating synthetic data for neural operators*. 2024. arXiv: 2401.02398 [cs.LG].

[152] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Vol. 1. Springer, 2006.

[153] Gerald Farin. *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier, 2014.

[154] Joseph L. Steger and Denny S. Chaussee. "Generation of body-fitted coordinates using hyperbolic partial differential equations". In: *SIAM Journal on Scientific and Statistical Computing* 1.4 (1980), pp. 431–437.

[155] Michael B Giles and Endre Süli. "Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality". In: *Acta Numerica* 11 (2002), pp. 145–236.

[156] Ian Goodfellow. *Deep learning*. 2016.

[157] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. "Highly accurate protein structure prediction with AlphaFold". In: *nature* 596.7873 (2021), pp. 583–589.

[158] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. "Magnetic control of tokamak plasmas through deep reinforcement learning". In: *Nature* 602.7897 (2022), pp. 414–419.

[159] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. "Probabilistic weather forecasting with machine learning". In: *Nature* (2024), pp. 1–7.

[160] Cristian Bodnar, Wessel P Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A Weyn, Haiyu Dong, et al. "A foundation model for the earth system". In: *Nature* (2025), pp. 1–8.

[161] Felipe Cucker and Steve Smale. "On the mathematical foundations of learning". In: *Bulletin of the American mathematical society* 39.1 (2002), pp. 1–49.

[162] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.

[163] Helmut Bolcskei, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. "Optimal approximation with sparsely connected deep neural networks". In: *SIAM Journal on Mathematics of Data Science* 1.1 (2019), pp. 8–45.

[164] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114. ISSN: 0893-6080.

[165] Ronald DeVore, Boris Hanin, and Guergana Petrova. "Neural network approximation". In: *Acta Numerica* 30 (2021), pp. 327–444.

[166] Dennis Elbrächter, Dmytro Perekrestenko, Philipp Grohs, and Helmut Bölcskei. "Deep neural network approximation theory". In: *IEEE Transactions on Information Theory* 67.5 (2021), pp. 2581–2623.

[167] Julius Berner, Philipp Grohs, and Felix Voigtlaender. "Learning ReLU networks to high uniform accuracy is intractable". English. In: The Eleventh International Conference on Learning Representations: ICLR 2023 ; Conference date: 01-05-2023 Through 05-05-2023. 2023.

[168] Ahmed Abdeljawad and Philipp Grohs. *Sampling Complexity of Deep Approximation Spaces*. 2023. arXiv: 2312.13379 [cs.LG].

[169] Kateryna Pozharska and Tino Ullrich. "A Note on Sampling Recovery of Multivariate Functions in the Uniform Norm". In: *SIAM Journal on Numerical Analysis* 60.3 (2022), pp. 1363–1384. DOI: 10.1137/21M1410580.

[170] Nikola B. Kovachki, Samuel Lanthaler, and Andrew M. Stuart. "Operator learning: Algorithms and analysis". In: Handbook of Numerical Analysis. Elsevier, 2024. eprint: 2402.15715.

[171] Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. "On Universal Approximation and Error Bounds for Fourier Neural Operators". In: *Journal of Machine Learning Research* 22.290 (2021), pp. 1–76. eprint: 2107.07562.

[172] Nicola Rares Franco, Stefania Fresca, Andrea Manzoni, and Paolo Zunino. "Approximation bounds for convolutional neural networks in operator learning". In: *Neural Networks* 161 (2023), pp. 129–141.

[173] Carlo Marcati and Christoph Schwab. "Exponential Convergence of Deep Operator Networks for Elliptic Partial Differential Equations". In: *SIAM Journal on Numerical Analysis* 61.3 (2023), pp. 1513–1545. DOI: 10.1137/21M1465718.

[174] Hrushikesh Narhar Mhaskar and Nahmwoo Hahm. "Neural networks for functional approximation and system identification". In: *Neural Computation* 9.1 (1997), pp. 143–159.

[175] Samuel Lanthaler. "Operator learning with PCA-Net: upper and lower complexity bounds". In: *Journal of Machine Learning Research* 24.318 (2023), pp. 1–67.

[176] Samuel Lanthaler and Andrew M. Stuart. *The parametric complexity of operator learning*. 2023. arXiv: 2306.15924 [cs.LG].

[177] Samuel Lanthaler. *Operator Learning of Lipschitz Operators: An Information-Theoretic Perspective*. 2024. arXiv: 2406.18794 [cs.LG].

[178] Ben Adcock, Nick Dexter, and Sebastian Moraga. "Optimal approximation of infinite-dimensional holomorphic functions". In: *Calcolo* 61.1 (2024), p. 12.

[179] Ben Adcock, Michael Griebel, and Gregor Maier. *Learning Lipschitz Operators with respect to Gaussian Measures with Near-Optimal Sample Complexity*. 2024. arXiv: 2410.23440 [cs.LG].

[180] Nikola B. Kovachki, Samuel Lanthaler, and Hrushikesh Mhaskar. *Data Complexity Estimates for Operator Learning*. 2024. arXiv: 2405.15992 [cs.LG].

[181] Samuel Lanthaler, Zongyi Li, and Andrew M. Stuart. *Nonlocality and Nonlinearity Implies Universality in Operator Learning*. 2024. arXiv: 2304.13221 [math.NA].

[182] James Dugundji. "An extension of Tietze's theorem". In: *Pacific Journal of Mathematics* 1.3 (1951), pp. 353–367.

[183] Hans-J urgen Schmeisser and Hans Triebel. *Topics in Fourier Analysis and Function Spaces*. Vol. 1. Wiley, 1987.

[184] Elias M Stein and Guido Weiss. *Introduction to Fourier analysis on Euclidean spaces*. Vol. 1. Princeton University Press, 1971.

[185] Robert A Adams and John JF Fournier. *Sobolev Spaces*. Elsevier, 2003.

[186] J-L Guermond. "The LBB condition in fractional Sobolev spaces and applications". In: *IMA journal of numerical analysis* 29.3 (2009), pp. 790–805.

[187] Jöran Bergh and Jörgen Löfström. *Interpolation Spaces: An Introduction*. Vol. 223. Springer Science & Business Media, 2012.

[188] Kösaku Yosida. *Functional analysis*. Springer Science & Business Media, 2012.

[189] Allan Pinkus. "Approximation theory of the MLP model in neural networks". In: *Acta Numerica* 8 (1999), pp. 143–195. DOI: 10.1017/S0962492900002919.

[190] Terence Tao. *Analysis ii*. Third. Vol. 1. Springer, 2016.

Appendix A

APPENDIX TO CHAPTER 2

Links code used to produce the numerical results and figures in this chapter are available at

https://github.com/mtrautner/ViscoML

except for the material-dependence experiment, which is produced from code written by Lianghao Cao at

https://github.com/lcao11/multiscale_viscoelastic.

A.1 Proofs

Proof of Theorem 2.3.4

The proof of Lemma 2.3.5, which underlies the proof of Theorem 2.3.4, uses the following two propositions

***Proposition* A.1.1.** Under Assumptions 2.3.1, for all solutions u of equation (2.2.2) the following bounds hold for some constant C_1:

1. $\sup_{t\in\mathcal{T}} \|u\|^2_{H^1_0,\nu} \leq \|u|_{t=0}\|^2_{H^1_0,\nu} + \left(\frac{\nu^+}{E^-}\right)^2 \frac{1}{\nu^-} C_1^2 \|f\|^2_{\mathcal{Z}}$
2. $\sup_{t\in\mathcal{T}} \|u\|^2_{H^1_0,E} \leq \frac{E_+}{\nu_-}\|u|_{t=0}\|^2_{H^1_0,\nu} + \left(\frac{\nu^+}{E^-}\right)^2 \frac{E^+}{(\nu^-)^2} C_1^2 \|f\|^2_{\mathcal{Z}}$
3. $\|\partial_t u\|_{H^1_0,\nu} \leq \frac{C_1\|f\|_{\mathcal{Z}}}{\nu^-} + \frac{E^+}{\nu^-}\|u\|_{H^1_0,E}$, for all $t \in \mathcal{T}$.

◊

Proof. To show the first bound, let $\varphi = u$ in equation (2.2.2). We have

$$q_\nu(\partial_t u, u) + q_E(u, u) = \langle f, u\rangle$$

so that

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\|u\|^2_{H^1_0,\nu} + \|u\|^2_{H^1_0,E} &\leq \|f\|_{H^{-1}}\|u\|_{H^1_0} \\ &\leq C_1\|f\|\|u\|_{H^1_0}\end{aligned}$$

for some constant C_1, by compact embedding. Then, using Lemma 2.1.1,

$$\frac{1}{2}\frac{d}{dt}\|u\|^2_{H^1_0,\nu} + \frac{E^-}{\nu^+}\|u\|^2_{H^1_0,\nu} \le \frac{C_1}{2\delta^2}\|f\|^2 + \frac{\delta^2}{2\nu^-}\|u\|^2_{H^1_0,\nu}$$

for any $\delta > 0$ by Young's Inequality. Letting $\delta^2 = \frac{E^-\nu^-}{\nu^+}$, we have

$$\frac{d}{dt}\|u\|^2_{H^1_0,\nu} + \frac{E^-}{\nu^+}\|u\|^2_{H^1_0,\nu} \le \frac{C_1^2\nu^+}{E^-\nu^-}\|f\|^2_{\mathcal{Z}}.$$

Finally, Gronwall's inequality yields

$$\sup_{t\in\mathcal{T}}\|u\|^2_{H^1_0,\nu} \le \|u|_{t=0}\|^2_{H^1_0,\nu} + \left(\frac{\nu^+}{E^-}\right)^2\frac{C_1^2}{\nu^-}\|f\|^2.$$

The second bound follows from Lemma 2.1.1. For the third bound, let $\varphi = \partial_t u$ in equation (2.2.2). Then

$$q_\nu(\partial_t u, \partial_t u) + q_E(u, \partial_t u) = \langle f, \partial_t u\rangle$$

so that, again using Lemma 2.1.1, and using the Poincaré inequality,

$$\|\partial_t u\|^2_{H^1_0,\nu} \le \frac{C_1}{\nu^-}\|f\|\|\partial_t u\|_{H^1_0,\nu} + \frac{E^+}{\nu^-}\|u\|_{H^1_0,E}\|\partial_t u\|_{H^1_0,\nu}$$

and

$$\|\partial_t u\|_{H^1_0,\nu} \le \frac{C_1}{\nu^-}\|f\|_{\mathcal{Z}} + \frac{E^+}{\nu^-}\|u\|_{H^1_0,E}.$$

□

Additionally, we need to bound the difference between two solutions u_1 and u_2 of the PDE in Lemma 2.3.5 with different material properties. Notice that u_1 and u_2 satisfy

$$\frac{\partial}{\partial x}\left(E_1\left(\frac{\partial}{\partial x}u_1\right) + \nu_1\left(\frac{\partial^2}{\partial t\partial x}u_1\right)\right) = -f$$
$$\frac{\partial}{\partial x}\left(E_1\left(\frac{\partial}{\partial x}u_2\right) + \nu_1\left(\frac{\partial^2}{\partial t\partial x}u_2\right)\right) = -f + \frac{\partial}{\partial x}\left[(E_1 - E_2)\frac{\partial}{\partial x}u_2 + (\nu_1 - \nu_2)\frac{\partial^2}{\partial t\partial x}u_2\right],$$

Subtracting yields

$$\partial_x\left[E_1\partial_x\gamma + \nu_1\partial^2_{xt}\gamma\right] = -\partial_x\left[(\Delta E)\partial_x u_2 + (\Delta\nu)\partial^2_{xt}u\right],$$

where $\gamma = u_1 - u_2$, $\Delta E = E_1 - E_2$, and $\Delta\nu = \nu_1 - \nu_2$. We can rewrite this as an equation for γ in weak form: for all test functions $\varphi \in V := H^1_0$

$$q_{\nu_1}(\partial_t\gamma, \varphi) + q_{E_1}(\gamma, \varphi) = \langle g, \partial_x\varphi\rangle, \quad \gamma|_{t=0} = 0, \tag{A.1.1}$$

where $g = \Delta E\partial_x u_2 + \Delta\nu\partial^2_{xt}u_2$. For the following discussion of bounds including both u_1 and u_2, let $E^+ = \max\{E_1^+, E_2^+\}$, $\nu^+ = \max\{\nu_1^+, \nu_2^+\}$, $E^- = \min\{E_1^-, E_2^-\}$, and $\nu^- = \min\{\nu_1^-, \nu_2^-\}$.

***Proposition* A.1.2.** Under Assumptions 2.3.1, for all solutions γ of equation (A.1.1), the following bounds hold:

1. $\sup_{t\in\mathcal{T}} \|\gamma\|^2_{H^1_0,\nu_1} \leq \left(\frac{\nu^+}{E^-}\right)^2 \frac{1}{\nu^-}\|g\|^2_{\mathcal{Z}}$
2. $\sup_{t\in\mathcal{T}} \|\gamma\|^2_{H^1_0,E_1} \leq \left(\frac{\nu^+}{E^-}\right)^2 \frac{E^+}{(\nu^-)^2}\|g\|^2_{\mathcal{Z}}$
3. $\sup_{t\in\mathcal{T}} \|\partial_t\gamma\|_{H^1_0,\nu_1} \leq \frac{\|g\|_{\mathcal{Z}}}{\nu^-} + \frac{E^+}{\nu^-}\|\gamma\|_{H^1_0,E}$

◊

Proof. To show the first bound, let $\varphi = \gamma$ in equation (A.1.1). We have

$$q_\nu(\partial_t\gamma,\gamma) + q_E(\gamma,\gamma) = \langle g, \partial_x\gamma\rangle$$

so that

$$\frac{1}{2}\frac{d}{dt}\|\gamma\|^2_{H^1_0,\nu} + \|u\|^2_{H^1_0,E} \leq \|g\|\|\gamma\|_{H^1_0}.$$

Then

$$\frac{1}{2}\frac{d}{dt}\|\gamma\|^2_{H^1_0,\nu} + \frac{E^-}{\nu^+}\|\gamma\|^2_{H^1_0,\nu} \leq \frac{1}{2\delta^2}\|g\|^2 + \frac{\delta^2}{2\nu^-}\|\gamma\|^2_{H^1_0,\nu}$$

for any $\delta > 0$ by Young's Inequality. Letting $\delta^2 = \frac{E^-\nu^-}{\nu^+}$, we have

$$\frac{d}{dt}\|\gamma\|^2_{H^1_0,\nu} + \frac{E^-}{\nu^+}\|\gamma\|^2_{H^1_0,\nu} \leq \frac{\nu^+}{E^-\nu^-}\|g\|^2_{\mathcal{Z}}.$$

Finally, since $\gamma(0) = 0$, Gronwall's inequality yields

$$\sup_{t\in\mathcal{T}} \|\gamma\|^2_{H^1_0,\nu} \leq \left(\frac{\nu^+}{E^-}\right)^2 \frac{1}{\nu^-}\|g\|^2.$$

The second bound follows from Lemma 2.1.1. For the third bound, let $\varphi = \partial_t\gamma$ in equation (A.1.1). Then

$$q_\nu(\partial_t\gamma,\partial_t\gamma) + q_E(\gamma,\partial_t\gamma) = \langle g, \partial^2_{xt}\gamma\rangle$$

so that, again using Lemma 2.1.1,

$$\|\partial_t\gamma\|^2_{H^1_0,\nu} \leq \frac{1}{\nu^-}\|g\|\|\partial_t\gamma\|_{H^1_0,\nu} + \frac{E^+}{\nu^-}\|\gamma\|_{H^1_0,E}\|\partial_t\gamma\|_{H^1_0,\nu}$$

and

$$\|\partial_t\gamma\|_{H^1_0,\nu} \leq \frac{1}{\nu^-}\|g\|_{\mathcal{Z}} + \frac{E^+}{\nu^-}\|\gamma\|_{H^1_0,E}.$$

□

To prove the Lipschitz property of the solution in Theorem 2.3.4, we will need the following lemma.

***Lemma* 2.3.5** (Lipschitz Solution)**.** Let u_i be the solution to

$$-\partial_x\big(E_i(x)\partial_x u_i(x,t) + \nu_i(x)\partial^2_{xt}u_i(x,t)\big) = f(x,t), \qquad x\in\mathcal{D}, t\in\mathcal{T}, \tag{2.3.1}$$

$$u_i(x,0) = \partial_t u_i(x,0) = 0, \qquad x\in\mathcal{D}, \tag{2.3.2}$$

$$u_i(0,t) = u_i(D,t) = 0, \qquad t\in\mathcal{T}, \tag{2.3.3}$$

associated with material properties E_i, ν_i, for $i\in\{1,2\}$, and forcing f, all satisfying the Assumptions 2.3.1. Then

$$\|u_1 - u_2\|_{\mathcal{Z}} \le C(\|\nu_1-\nu_2\|_\infty + \|E_1-E_2\|_\infty)$$

for some constant $C\in\mathbb{R}^+$ dependent on $f, E_i^+, E_i^-, \nu_i^+, \nu_i^-$, and L and independent of ε. ◊

Proof. Let γ and g be as defined before and after equation A.1.1. Then, by the result of Proposition A.1.2,

$$\sup_{t\in\mathcal{T}}\|\gamma\|^2_{H^1_0} \le \frac{1}{\nu^-}\sup_{t\in\mathcal{T}}\|\gamma\|^2_{H^1_0,\nu_1} \le \left(\frac{\nu^+}{E^-\nu^-}\right)^2\|g\|^2_{\mathcal{Z}}.$$

To bound the RHS:

$$\begin{aligned}
\|g\|_{\mathcal{Z}} &= \|(\Delta E)\partial_x u_2 + (\Delta\nu)\partial^2_{xt}u_2\|_{\mathcal{Z}}\\
&\le \|(\Delta E)\partial_x u_2\|_{\mathcal{Z}} + \|(\Delta\nu)\partial^2_{xt}u_2\|_{\mathcal{Z}}\\
&\le \|\Delta E\|_\infty\|\partial_x u_2\|_{\mathcal{Z}} + \|\Delta\nu\|_\infty\|\partial^2_{xt}u_2\|_{\mathcal{Z}}\\
&\le \sup_{t\in\mathcal{T}}\|u_2\|_{H^1_0}\|\Delta E\|_\infty + \sup_{t\in\mathcal{T}}\|\partial_t u_2\|_{H^1_0}\|\Delta\nu\|_\infty\\
&\le \frac{1}{(\nu^-)^{\frac12}}\Big(\sup_{t\in\mathcal{T}}\|u_2\|_{H^1_0,\nu_2}\|\Delta E\|_\infty + \sup_{t\in\mathcal{T}}\|\partial_t u_2\|_{H^1_0,\nu_2}\|\Delta\nu\|_\infty\Big).
\end{aligned}$$

To bound $\|\partial_x u_2\|_{\mathcal{Z}}$ and $\|\partial^2_{xt}u_2\|$, note that any solution u_2 will satisfy equation (2.2.2) for $(u,E,\nu)\mapsto(u_2,E_2,\nu_2)$. The analysis of Proposition A.1.1 yields

$$\sup_{t\in\mathcal{T}}\|u_2\|_{H^1_0,\nu_2} \le \|u|_{t=0}\|_{H^1_0,\nu} + \left(\frac{\nu^+}{E^-}\right)\frac{C_1}{(\nu^-)^{1/2}}\|f\|_{\mathcal{Z}}$$

and

$$\sup_{t\in\mathcal{T}}\|\partial_t u_2\|_{H_0^1,\nu_2} \le \frac{C_1\|f\|_{\mathcal{Z}}}{\nu^-} + \frac{E^+}{\nu^-}\|u_2\|_{H_0^1,E}$$
$$\le \frac{C_1}{\nu^-}\|f\|_{\mathcal{Z}} + \left(\frac{E^+}{\nu^-}\right)^{3/2}\|u|_{t=0}\|_{H_0^1,\nu} + \frac{(E^+)^{3/2}}{(\nu^-)^2}\frac{\nu^+}{E^-}C_1\|f\|_{\mathcal{Z}}.$$

By the Poincaré inequality, $\|\gamma\|_{\mathcal{Z}} \le C_p \sup_{t\in\mathcal{T}}\|\gamma\|_{H_0^1}$ for some constant C_p and setting

$$C = C_p\left(\frac{\nu^+}{E^-(\nu^-)^{\frac{3}{2}}}\right)\max\Big\{\|u|_{t=0}\|_{H_0^1,\nu} + \left(\frac{\nu^+}{E^-}\right)\frac{C_1}{(\nu^-)^{1/2}}\|f\|_{\mathcal{Z}},$$
$$\frac{C_1}{\nu^-}\|f\|_{\mathcal{Z}} + \left(\frac{E^+}{\nu^-}\right)^{3/2}\|u|_{t=0}\|_{H_0^1,\nu} + \frac{(E^+)^{3/2}}{(\nu^-)^2}\frac{\nu^+}{E^-}C_1\|f\|_{\mathcal{Z}}\Big\}$$

gives the result.

□

Now we can prove the piecewise constant approximation theorem.

***Theorem* 2.3.4** (Piecewise-Constant Approximation)**.** Let E and ν be piecewise-continuous functions, with a finite number of discontinuities, satisfying Assumptions 2.3.1; let u_ε be the corresponding solution to (2.2.1). Then, for any $\delta > 0$, there exist piecewise-constant E^{PC} and ν^{PC} such that solution $u_\varepsilon^{\mathrm{PC}}$ of equations (2.2.1) with these material properties satisfies

$$\|u_\varepsilon^{\mathrm{PC}} - u_\varepsilon\|_{\mathcal{Z}} < \delta.$$

◊

Proof. Let $\mathcal{A}_E$ and $\mathcal{A}_\nu$ be the finite sets of discontinuities of E_ϵ and ν_ϵ respectively, and let $\mathcal{A} = \mathcal{A}_E \cup \mathcal{A}_\nu$ with elements $a_1, a_2, \ldots, a_K$. Partition the interval Ω into intervals $D_1 = (a_0, a_1), D_2 = [a_1, a_2), \ldots D_K = [a_{K-1}, a_K)$ such that $\bigcup_{k=1}^K D_k = \Omega$ and $\bigcap_{k=1}^K D_k = 0$. Let $B_{k,\delta} = \{b_0^k, b_1^k, \ldots, b_{N(\delta)}^k\}$ be a uniform partition of D_k such that $b_i^k - b_{i-1}^k = \delta$. Furthermore, define E_ϵ^{PC} and $\nu_\epsilon^{\mathrm{PC}}$ via

$$E_\epsilon^{\mathrm{PC}}(x) = \sum_{k=1}^K\sum_{n=1}^N \mathbf{1}_{x\in(b_{n-1}^k, b_n^k]}E\left(\frac{1}{2}b_{n-1}^k + \frac{1}{2}b_n^k\right)$$
$$\nu_\epsilon^{\mathrm{PC}}(x) = \sum_{k=1}^K\sum_{n=1}^N \mathbf{1}_{x\in(b_{n-1}^k, b_n^k]}\nu\left(\frac{1}{2}b_{n-1}^k + \frac{1}{2}b_n^k\right)$$

for $x \in \Omega$, noting that E_ϵ^{PC} and $\nu_\epsilon^{\mathrm{PC}}$ are piecewise constant with $KN(\delta)$ pieces.

E_ϵ and ν_ϵ are continuous on each interval D_k, so for each $\delta' > 0$, there exists a mesh width δ such that with partitions $\{B_{k,\delta}\}_{k=1}^K$

$$\sup_{x \in (b_{n-1}^k, b_n^k]} \|E_\epsilon\left(\frac{1}{2}b_{n-1}^k + \frac{1}{2}b_n^k\right) - E_\epsilon(x)\| < \delta'$$
$$\sup_{x \in (b_{n-1}^k, b_n^k]} \|\nu_\epsilon\left(\frac{1}{2}b_{n-1}^k + \frac{1}{2}b_n^k\right) - \nu_\epsilon(x)\| < \delta'$$

for all $n \in \{1, \ldots, N(\delta)\}$. Thus, $\|E^{\mathrm{PC}} - E\|_\infty < \delta'$ and $\|\nu^{\mathrm{PC}} - \nu\|_\infty < \delta'$. Since δ' was arbitrary, we can pick $\delta' < \frac{\eta}{C_1}$ where C_1 is as in Lemma 2.3.5, and the theorem follows by use of the same lemma. □

Proof of Theorem 2.3.6

We will need the following lemma:

***Lemma* A.1.3** (Existence of Exact Parametrization)**.** For a piecewise constant material with $L'+1$ pieces and under Assumptions 2.3.1, a_0 in equation (2.2.5) can be written exactly as

$$\widehat{a}_0(s) = E' + \nu' s - \sum_{\ell=1}^{L'} \frac{\beta_\ell}{s + \alpha_\ell},$$

where $E', \nu', \beta_\ell \in \mathbb{R}$ and $\alpha_\ell \in \mathbb{R}_+$ for all $\ell \in [L']$. ◊

Proof. Let $E(y)$ and $\nu(y)$ have $L'+1$ constant pieces of lengths $\{d_\ell\}_{\ell=1}^{L'+1}$, each associated to values $\{E_\ell\}_{\ell=1}^{L'+1}$ and $\{\nu_\ell\}_{\ell=1}^{L'+1}$ of E and ν. Then equation (2.2.5), rewritten here for convenience

$$\widehat{a}_0(s) = \left(\int_0^1 \frac{dy}{s\nu(y) + E(y)}\right)^{-1},$$

becomes

$$\widehat{a}_0(s) = \left[\sum_{\ell=1}^{L'+1} \frac{d_\ell}{E_\ell + \nu_\ell s}\right]^{-1} \tag{A.1.2}$$
$$= \frac{\prod_{\ell=1}^{L'+1}(E_\ell + \nu_\ell s)}{\sum_{\ell=1}^{L'+1} d_\ell \prod_{j \neq \ell}(E_j + \nu_j s)} \tag{A.1.3}$$
$$= \frac{P(s)}{Q(s)}, \tag{A.1.4}$$

where $P(s)$ is a polynomial of degree $L'+1$ and $Q(s)$ a polynomial of degree L'. Therefore, there exists a decomposition

$$\frac{P(s)}{Q(s)} = E' + \nu' s - \frac{C(s)}{Q(s)} \tag{A.1.5}$$

for some constants E' and ν' and polynomial $C(s)$ of degree $L'-1$.

Let $-\alpha_1, \ldots, -\alpha_{L'}$ be the roots of $Q(s)$. Then $\frac{C(s)}{Q(s)} = \sum_{\ell=1}^{L'} \frac{\beta_\ell}{s+\alpha_\ell}$ for some constants $\beta_\ell \in \mathbb{C}$ by partial fraction decomposition. We wish to show that $\mathrm{Re}(\alpha_\ell) > 0$ for all roots $-\alpha_\ell$ of $Q(s)$ so that we can take the inverse Laplace transform. Furthermore, we wish to show that, in fact, $-\alpha_\ell \in \mathbb{R}$ for all roots $-\alpha_\ell$ so that $\beta_\ell \in \mathbb{R}$ as well. Since E_j and v_j are positive for all $j \in [L'+1]$, it is clear that if a root $-\alpha_\ell$ is real, then it cannot be positive since $Q(s) = \sum_{\ell=1}^{L'+1} d_\ell \prod_{j\neq\ell}(E_j + \nu_j s)$ has all positive coefficients. We now show that all roots of $Q(s)$ are real. Suppose $a+bi$ is a root of $Q(s)$. Then

$$\begin{aligned}
Q(a+bi) &= \sum_{\ell=1}^{L'+1} d_\ell \prod_{j\neq\ell}(E_j + \nu_j(a+bi)) \\
&= \left[\prod_{j=1}^{L'+1}(E_j + \nu_j(a+bi))\right] \cdot \sum_{\ell=1}^{L'+1} \frac{d_\ell}{E_\ell + \nu_\ell(a+bi)} \\
&= \left[\prod_{j=1}^{L'+1}(E_j + \nu_j(a+bi))\right] \cdot \sum_{\ell=1}^{L'+1} \left(\frac{d_\ell(E_\ell + \nu_\ell a)}{(E_\ell+\nu_\ell a)^2 + (\nu_\ell b)^2} - \frac{d_\ell(\nu_\ell b)}{(E_\ell+\nu_\ell a)^2 + (\nu_\ell b)^2} i \right).
\end{aligned}$$

The term $\prod_{j=1}^{L'+1}(E_j + \nu_j(a+bi))$ is a nonzero constant for $b \neq 0$ since $E_j, \nu_j \in \mathbb{R}_+$. Therefore, for $Q(a+bi) = 0$, both the real and imaginary components of the sum on the right must total 0. However, since d_ℓ, ν_ℓ, and the denominator term $(E_\ell + \nu_\ell a)^2 + (\nu_\ell b)^2$ are all positive as well, b must equal 0 to make $\mathrm{Im}[Q(a+bi)] = 0$. Therefore, all roots of $Q(s)$ are in $\mathbb{R}_-$. Returning to the decomposition, we now have

$$\widehat{a}_0(s) = E' + \nu' s - \sum_{\ell=1}^{L'} \frac{\beta_\ell}{s+\alpha_\ell}, \tag{A.1.6}$$

where $\beta_\ell \in \mathbb{R}$ and $\alpha_\ell \in \mathbb{R}_+$ for all $\ell \in [L']$. □

Now we may prove the theorem.

***Theorem* 2.3.6** (Existence of Exact Parametrization)**.** Let $\Psi_0^\dagger$ be the map from strain history to stress in the homogenized model, as defined by equation

(2.2.7), in a piecewise-constant material with $L+1$ pieces. Define Ψ_0^{PC} : $\mathbb{R}^2 \times C^1(\mathcal{T};\mathbb{R}) \times \mathcal{T} \times \Theta \to \mathbb{R}$ by

$$\Psi_0^{\mathrm{PC}}(\overline{\epsilon}(t), \dot{\overline{\epsilon}}(t), \{\overline{\epsilon}(\tau)\}_{\tau\in\mathcal{T}}, t; \theta) = E'\overline{\epsilon}(t) + \nu'\dot{\overline{\epsilon}}(t) - \sum_{\ell=1}^{L} \xi_\ell(t), \tag{2.3.4a}$$

$$\partial_t \xi_\ell(t) = \beta_\ell \overline{\epsilon}(t) - \alpha_\ell \xi_\ell(t),\ \xi_\ell(0) = 0, \quad \ell \in \{1,\dots,L\}. \tag{2.3.4b}$$

Then, under Assumptions 2.3.1, there exists a choice of parameters $\theta^* = (E', \nu', \alpha, \beta, L)$ such that

$$\Psi_0^{\dagger}(\overline{\epsilon}(t), \dot{\overline{\epsilon}}(t), \{\overline{\epsilon}(\tau)\}_{\tau\in\mathcal{T}}, t) = \Psi_0^{\mathrm{PC}}(\overline{\epsilon}(t), \dot{\overline{\epsilon}}(t), \{\overline{\epsilon}(\tau)\}_{\tau\in\mathcal{T}}, t; \theta^*)$$

for all $u_0 \in \mathcal{C}^2(\mathcal{D}\times\mathcal{T};\mathbb{R})$ and $t \in \mathcal{T}$. ◊

Proof. By Lemma A.1.3, we have that

$$\begin{aligned}\widehat{\sigma_0} &= \widehat{a_0}(s)\partial_x \widehat{u_0} \\ &= \left(E' + \nu' s - \sum_{\ell=1}^{L'} \frac{\beta_\ell}{s+\alpha_\ell}\right)\partial_x \widehat{u_0},\end{aligned}$$

where $\beta_\ell \in \mathbb{R}$ and $\alpha_\ell \in \mathbb{R}_+$ for all $\ell \in [L]$. Taking an inverse Laplace transform, we get

$$\sigma_0(t) = E'\partial_x u_0(t) + \nu'\partial_t\partial_x u_0(t) - \sum_{\ell=1}^{L} \beta_\ell \int_0^t \partial_x u_0(\tau)\exp[-\alpha_\ell(t-\tau)]\ \mathsf{d}\tau. \tag{A.1.7}$$

The above may be reexpressed as equations (2.3.4) with a choice of parameters $\theta = (E', \nu', L', \alpha, \beta)$, auxiliary variable ξ, and $\overline{\epsilon} := \partial_x u_0$. □

RNO Approximation Theorem 2.3.10 Proof

In this subsection we use $|\cdot|, \|\cdot\|$ to denote modulus and Euclidean norm, respectively, and $\langle\cdot,\cdot\rangle$ to denote Euclidean inner product. This overlap with the notation from the main body of the work should not lead to any confusion as it is confined to this subsection.

To prove Theorem 2.3.10 we first study the simple case where $\mathcal{F}^{\mathrm{PC}}, \mathcal{G}^{\mathrm{PC}}$ are uniformly approximated across all inputs; subsequently we will use this to establish Theorem 2.3.10 as stated.

***Assumption* 1.1.4.** For any $\delta > 0$, there exist $\mathcal{F}^{\text{RNO}}$ and $\mathcal{G}^{\text{RNO}}$ such that

$$\sup_{z \in \mathbb{R}^{2+L}} \left| \mathcal{F}^{\text{PC}}(z) - \mathcal{F}^{\text{RNO}}(z) \right| \leq \delta$$
$$\sup_{z \in \mathbb{R}^{1+L}} \left\| \mathcal{G}^{\text{PC}}(z) - \mathcal{G}^{\text{RNO}}(z) \right\| \leq \delta.$$

$\Diamond$

***Proposition* A.1.5.** Under Assumptions 1.1.4, if $\{\alpha_\ell\}$ in equations (2.3.9) are bounded such that $0 < a_0 < \alpha_\ell$ for some a_0 for all $\ell \in [L]$, then for any $\eta > 0$, there exists a map Ψ_0^{RNO} defined as in equations (2.3.11) such that for Ψ_0^{PC} defined in equations (2.3.10), for any $t \in \mathbb{R}^+$ and functions $b, c : \mathbb{R}^+ \to \mathbb{R}$,

$$\left| \Psi_0^{\text{PC}}(b(t), c(t), \{b(\tau)\}_{\tau \in \overline{\mathcal{T}}}, t; \theta^*) - \Psi_0^{\text{RNO}}(b(t), c(t), \{b(\tau)\}_{\tau \in \overline{\mathcal{T}}}, t) \right| \leq \eta.$$

$\Diamond$

Proof. Note that the main difficulty in this proof results from the fact that $\mathcal{F}^{\text{RNO}}$ and $\mathcal{F}^{\text{PC}}$ act on different hidden variables ξ, which we will denote ξ^{RNO} and ξ^{PC}, and whose first order time derivatives are given by $\mathcal{G}^{\text{RNO}}$ and $\mathcal{G}^{\text{PC}}$ respectively. We write

$$\begin{aligned}
&\left| \Psi_0^{\text{PC}}(b(t), c(t), \{b(\tau)\}_{\tau \in \overline{\mathcal{T}}}, t; \theta^*) - \Psi_0^{\text{RNO}}(b(t), c(t), \{b(\tau)\}_{\tau \in \overline{\mathcal{T}}}, t) \right| \\
&= |\mathcal{F}^{\text{PC}}(b(t), c(t), \xi^{\text{PC}}(t)) - \mathcal{F}^{\text{RNO}}(b(t), c(t), \xi^{\text{RNO}}(t))| \\
&\leq |\mathcal{F}^{\text{PC}}(b(t), c(t), \xi^{\text{RNO}}(t)) - \mathcal{F}^{\text{RNO}}(b(t), c(t), \xi^{\text{RNO}}(t))| \\
&+ |\mathcal{F}^{\text{PC}}(b(t), c(t), \xi^{\text{PC}}(t)) - \mathcal{F}^{\text{PC}}(b(t), c(t), \xi^{\text{RNO}}(t))| \\
&\leq \delta + |\mathcal{F}^{\text{PC}}(b(t), c(t), \xi^{\text{PC}}(t)) - \mathcal{F}^{\text{PC}}(b(t), c(t), \xi^{\text{RNO}}(t))|
\end{aligned}$$

by Assumptions 1.1.4 since $\mathcal{F}^{\text{PC}}$ and $\mathcal{F}^{\text{RNO}}$ share arguments in the first term. To bound the second term,

$$|\mathcal{F}^{\text{PC}}(b(t), c(t), \xi^{\text{PC}}(t)) - \mathcal{F}^{\text{PC}}(b(t), c(t), \xi^{\text{RNO}}(t))| = |\langle \mathbf{1}, \xi^{\text{PC}}(t) - \xi^{\text{RNO}}(t) \rangle| \leq \sqrt{L} \|\xi^{\text{PC}}(t) - \xi^{\text{RNO}}(t)\|$$

using the known form of $\mathcal{F}^{\text{PC}}$ where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^L$.

Let $e_\xi(t) = \xi^{\text{PC}}(t) - \xi^{\text{RNO}}(t)$. Note that $\xi^{\text{PC}}(0) = \xi^{\text{RNO}}(0) = 0$, so $e_\xi(0) = 0$. We wish to bound $\|e_\xi(t)\|$. To do so, we first bound $\|\dot{e}_\xi(t)\|$, where $\dot{e}_\xi(t) = \frac{d}{dt} e_\xi(t)$:

$$\begin{aligned}
\dot{e}_\xi(t) &= \dot{\xi}^{\text{PC}}(t) - \dot{\xi}^{\text{RNO}}(t) \\
&= \mathcal{G}^{\text{PC}}(\xi^{\text{PC}}(t), b(t)) - \mathcal{G}^{\text{PC}}(\xi^{\text{RNO}}(t), b(t)) - \mathcal{G}^{\text{RNO}}(\xi^{\text{RNO}}(t), b(t)) + \mathcal{G}^{\text{PC}}(\xi^{\text{RNO}}(t), b(t)) \\
&= \mathcal{G}^{\text{PC}}(\xi^{\text{PC}}(t), b(t)) - \mathcal{G}^{\text{PC}}(\xi^{\text{RNO}}(t), b(t)) + q(t),
\end{aligned}$$

where we have defined $q(t) = \mathcal{G}^{\text{PC}}(\xi^{\text{RNO}}(t), b(t)) - \mathcal{G}^{\text{RNO}}(\xi^{\text{RNO}}(t), b(t))$ and $\|q(t)\| \leq \delta$ by Assumptions 1.1.4. Now note that $\dot{e}_\xi(t) = -Ae_\xi(t) + q(t)$ by the form of $\mathcal{G}^{\text{PC}}$, so we can bound

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}\|e_\xi(t)\|^2 = \langle e_\xi(t), \dot{e}_\xi(t)\rangle &= -\langle e_\xi(t), Ae_\xi(t)\rangle + \langle q(t), e_\xi(t)\rangle \\
&\leq -\alpha_{\min}\|e_\xi(t)\|^2 + \left\langle \frac{1}{\alpha_{\min}^{1/2}} q(t), \alpha_{\min}^{1/2} e_\xi(t) \right\rangle \\
&\leq -\alpha_{\min}\|e_\xi(t)\|^2 + \frac{1}{2\alpha_{\min}}\|q(t)\|^2 + \frac{\alpha_{\min}}{2}\|e_\xi(t)\|^2 \\
\frac{d}{dt}\|e_\xi(t)\|^2 &\leq -\alpha_{\min}\|e_\xi(t)\|^2 + \frac{\delta^2}{\alpha_{\min}}
\end{aligned}$$

by Young's inequality. Then by Gronwall's inequality

$$\|e_\xi(t)\|^2 \leq \frac{\delta^2}{\alpha_{\min}^2}\left(1 - e^{-\alpha_{\min} t}\right) \tag{A.1.8}$$

so $\|e_\xi(t)\| < \frac{\delta}{a_0}$ for all time. Returning to the main proof narrative,

$$\begin{aligned}
&\left|\Psi_0^{\text{PC}}(b(t), c(t), \{b(\tau)\}_{\tau\in\overline{\mathcal{T}}}, t; \theta^*) - \Psi_0^{\text{RNO}}(b(t), c(t), \{b(\tau)\}_{\tau\in\overline{\mathcal{T}}}, t)\right| \\
&\leq \delta + \sqrt{L}\|\xi^{\text{PC}}(t) - \xi^{\text{RNO}}(t)\| \leq \delta + \frac{\sqrt{L}\delta}{a_0}.
\end{aligned}$$

Since by Assumptions 1.1.4, δ is arbitrarily small, the theorem result is shown. □

Although we did not need to restrict the inputs t, b, and c in Proposition A.1.5 to compact sets in order to prove it, we will argue that the statement holds under weaker assumptions if the inputs are also bounded. The following weaker assumptions follow from the Universal Approximation Theorem for RNNs[4].

***Assumption* 1.1.6.** If $D_1 \in \mathbb{R}^{2+L}$ and $D_2 \in \mathbb{R}^{1+L}$ are compact sets, then for any $\delta > 0$, there exist $\mathcal{F}^{\text{RNO}}$ and $\mathcal{G}^{\text{RNO}}$ such that

$$\begin{aligned}
\sup_{z\in D_1}\left|\mathcal{F}^{\text{PC}}(z) - \mathcal{F}^{\text{RNO}}(z)\right| &\leq \delta \\
\sup_{z\in D_2}\left\|\mathcal{G}^{\text{PC}}(z) - \mathcal{G}^{\text{RNO}}(z)\right\| &\leq \delta.
\end{aligned}$$

◊

***Theorem* 2.3.10** (RNO Approximation)**.** Consider Ψ_0^{PC} as defined by equations (2.3.9), (2.3.10). Assume that there exist $a_0 > 0$ and $0 \leq B < \infty$ such

that $a_0 < \min_\ell |\alpha_\ell|$ and $\max_\ell |\beta_\ell| \leq B$. Then, under Assumptions 2.3.1, for every $\eta > 0$ there exists Ψ_0^{RNO} of the form (2.3.11) such that

$$\sup_{t\in\mathcal{T}, b,c\in\mathsf{Z}_R} \left|\Psi_0^{\mathrm{PC}}\big(b(t), c(t), \{b(\tau)\}_{\tau\in\mathcal{T}}, t; \theta^*\big) - \Psi_0^{\mathrm{RNO}}\big(b(t), c(t), \{b(\tau)\}_{\tau\in\mathcal{T}}, t\big)\right| < \eta.$$

$\Diamond$

Proof. Notice first that Assumptions 1.1.6 are a weaker version of Assumptions 1.1.4. We will prove the theorem by showing that, for inputs bounded via $t \in \mathcal{T}$ and $b, c \in \mathsf{Z}_R$, we never need the stronger assumption in the proof of Proposition A.1.5 because the function arguments of $\mathcal{F}^{\mathrm{PC}}, \mathcal{F}^{\mathrm{RNO}}, \mathcal{G}^{\mathrm{PC}}$, and $\mathcal{G}^{\mathrm{RNO}}$ never leave a compact set. First we show that $\sup_{t\in\mathcal{T}} \|\xi^{\mathrm{PC}}(t)\| \leq R_3$ for some $R_3 > 0$. For any $\ell \in \{1, \ldots, L\}$, we have

$$\begin{aligned}
\dot{\xi}_\ell^{\mathrm{PC}}(t) &= \beta_\ell b(t) - \alpha_\ell \xi_\ell^{\mathrm{PC}}(t) \\
|\xi_\ell^{\mathrm{PC}}(t)| &\leq e^{-\alpha_\ell t}\beta_\ell \left(\sup_{t\in\mathcal{T}} |b(t)|\right) \int_0^t e^{\alpha_\ell t'} dt' \\
&\leq e^{-\alpha_\ell t}\beta_\ell \left(\sup_{t\in\mathcal{T}} |b(t)|\right) \frac{1}{\alpha_\ell} e^{\alpha_\ell t} \\
\sup_{t\in\mathcal{T}} |\xi_\ell^{\mathrm{PC}}(t)| &\leq \frac{B}{\rho} R,
\end{aligned}$$

so that $\sup_{t\in\mathcal{T}} \|\xi^{\mathrm{PC}}(t)\| \leq \frac{\sqrt{L}BR}{\rho}$. Let $R_3 = \frac{\sqrt{L}BR}{\rho}$. Define $R_4 = \max\{R_3 + \frac{\delta}{\rho}, R\}$ for δ in Assumptions 1.1.6. We will show that $\sup_{t\in\mathcal{T}} \|\xi^{\mathrm{RNO}}(t)\| \leq R_4$ for ξ^{RNO} defined by ξ^{RNO} in equations (2.3.11). Then the proof of Proposition A.1.5 will apply for bounded t, b and c with the weaker assumptions since all inputs to $\mathcal{F}^{\mathrm{PC}}, \mathcal{F}^{\mathrm{RNO}}, \mathcal{G}^{\mathrm{PC}}$, and $\mathcal{G}^{\mathrm{RNO}}$: $b(t)$, $c(t)$, $\xi^{\mathrm{PC}}(t)$, and $\xi^{\mathrm{RNO}}(t)$, will remain in a compact set for $t \in \mathcal{T}$.

Suppose for the sake of contradiction that there exists a time $t' \in \mathcal{T}$ at which $\|\xi^{\mathrm{RNO}}(t')\| > R_4$. Then there exists a time $T' < t' < T$ and $\epsilon > 0$ such that for $t \in [0, T']$, $\|\xi^{\mathrm{RNO}}(t)\| \leq R_4$, for $t \in (T', T' + \epsilon)$, $\|\xi^{\mathrm{RNO}}(t)\| > R_4$, and $\|\xi^{\mathrm{RNO}}(T')\| = R_4$ by continuity. In other words, T' is the time at which ξ^{RNO} first crosses the R_4 radius. Then

$$\|e_\xi(T')\| := \|\xi^{\mathrm{RNO}}(T') - \xi^{\mathrm{PC}}(T')\| \geq R_4 - R_3 \geq \frac{\delta}{\rho}$$

by triangle inequality. Since $\|\xi^{\mathrm{RNO}}(t)\| \leq R_4$ for $t \in [0, T']$, the bound on $\|e_\xi(t)\|$ in equation (A.1.8) in the proof of Proposition A.1.5 applies on the

interval $t \in [0, T']$ under the weaker assumptions 1.1.6, and $\|e_\xi(T')\| < \frac{\delta}{\rho}$. This is a contradiction. Therefore, $\sup_{t\in\mathcal{T}} \|\xi^{\text{RNO}}(t)\| \leq R_4$, and the proof of Proposition A.1.5 holds with the weaker Assumptions 1.1.6 for bounded inputs $t \in \mathcal{T}$ and $b, c \in \mathsf{Z}_R$. □

The bounds on α and β required in Theorem 2.3.10 are justified because for known material properties E and ν, α and β are determined and finite-dimensional, so they have maximum and minimum values.

A.2 Special Case Solutions

Laplace Transform Limit

Here we derive the form of $\Psi_0^\dagger$ in equation (2.2.7) via a power series expansion of the Laplace transform at $s = \infty$. Starting from the definition in equation (2.2.6):

$$\Psi_0^\dagger = \mathcal{L}^{-1}\left(\left(\int_0^1 \frac{dy}{s\nu(y) + E(y)}\right)^{-1} \partial_x \widehat{u}_0\right).$$

For $s \gg 1$, we have that

$$\left(\int_0^1 \frac{dy}{s\nu(y) + E(y)}\right)^{-1} \approx \left(\int_0^1 \frac{dy}{s\nu(y)}\right)^{-1} = s\left(\frac{dy}{\nu(y)}\right)^{-1}.$$

Setting $\nu' = \left(\frac{dy}{\nu(y)}\right)^{-1}$, we now subtract out the linear dependence on s and let $z = \frac{1}{s}$. We define

$$F(z) = \widehat{a}_0(s) - \nu' s\Big|_{s=z^{-1}}$$

to obtain

$$\begin{aligned}
F(z) &= \widehat{a}_0\big(z^{-1}\big) - \nu' z^{-1} \\
&= \left(\int_0^1 \frac{z\, dy}{\nu(y) + zE(y)}\right)^{-1} - \left(\int_0^1 \frac{z\, dy}{\nu(y)}\right)^{-1} \\
&= \frac{\int_0^1 \left(\frac{z}{\nu(y)} - \frac{z}{\nu(y)+zE(y)}\right) dy}{\left(\int_0^1 \frac{z\, dy}{\nu(y)+zE(y)}\right)\left(\int_0^1 \frac{z\, dy}{\nu(y)}\right)} \\
&= \frac{z^2 \int_0^1 \frac{E(y)}{\nu(y)(\nu(y)+zE(y))}\, dy}{z^2\left(\int_0^1 \frac{dy}{\nu(y)+zE(y)}\right)\left(\int_0^1 \frac{dy}{\nu(y)}\right)} \\
&= \frac{\int_0^1 \frac{E(y)}{\nu(y)(\nu(y)+zE(y))}\, dy}{\left(\int_0^1 \frac{dy}{\nu(y)+zE(y)}\right)\left(\int_0^1 \frac{dy}{\nu(y)}\right)}.
\end{aligned}$$

Since $\inf_{y\in(0,1)} \nu(y) > 0$,

$$\lim_{z\to 0} F(z) = \frac{\int_0^1 \frac{E(y)}{\nu^2(y)}\, dy}{\left(\int_0^1 \frac{dy}{\nu(y)}\right)^2} =: E'.$$

From this same computation, we see that for $\widehat{a}_0(s) = s\nu' + E' + \kappa(s)$, the contribution $\kappa(s)$ consists of lower order terms in s and is such that $\lim_{s\to\infty} \kappa(s) = 0$. Using the fact that the inverse Laplace transform of a product (if it exists) is a convolution, we justify the form of the integral term in equation (2.2.7).

Forced Boundary Problem

***Lemma* 2.5.1.** Let $\Omega = (0,1)$, and let σ be determined by the following equations, where E, ν, and b are given:

$$\partial_y \sigma(y,t) = 0, \qquad y \in \Omega, t \in \mathcal{T}, \tag{2.5.2a}$$

$$\sigma(y,t) = E(y)\partial_y u(y,t) + \nu(y)\partial^2_{yt} u(y,t), \qquad y \in \Omega, t \in \mathcal{T}, \tag{2.5.2b}$$

$$u(0,t) = 0, \quad u(1,t) = b(t), \qquad t \in \mathcal{T}, \tag{2.5.2c}$$

$$u(y,0) = 0, \qquad y \in \Omega. \tag{2.5.2d}$$

Then

$$\{\sigma(t)\}_{t\in\mathcal{T}} = \Psi_0^\dagger(b(t), \partial_t b(t), \{b(t)\}_{t\in\mathcal{T}}, t),$$

where $\Psi_0^\dagger$ is the map defined in (2.2.6). ◊

Proof. Taking the Laplace transform of (2.5.2) yields

$$\widehat{\sigma}(s) = (E(y) + \nu(y)s)\partial_y \widehat{u}(y,s).$$

Spatially averaging and noting that $b(t) = \langle \partial_y u(y,t) \rangle$, we have

$$\widehat{b}(s) = \int_0^1 \frac{dy}{(E + s\nu)(y)} \widehat{\sigma}(s). \tag{A.2.1}$$

Then $\widehat{\sigma}(s) = \left(\int_0^1 \frac{dy}{(E+s\nu)(y)}\right)^{-1} \widehat{b}(s)$, which is equivalent to $\widehat{\sigma}(s) = \widehat{a_0}(s)\widehat{b}(s)$ using equation (2.2.5). The definition of $\Psi_0^\dagger$ in equation (2.2.6) completes the proof.

□

Lemma 2.5.1 justifies the use of data arising from the system (2.5.2) to train the map Ψ_0.

A.3 Surrogate Model Experiments in Viscoelasticity

RNO Training and Testing: Piecewise-Constant Case

We trained three RNOs using the same dataset for the setting of a 2-piecewise constant material with material parameters $E_1 = 1$, $E_2 = 3$, $\nu_1 = 0.1$, and $\nu_2 = 0.2$. The data was generated using a forward Euler method with time discretization $dt = 0.001$ up to time $T = 4$ on the known analytic solution for the 2-piecewise-constant cell problem. Denote the data by $\{(\partial_x u_0)_n, (\sigma_0)_n\}_{n=1}^N$ as discussed in Section 2.5. We repeat the two loss functions here.

Accessible Loss Function:

$$L_1(\{\sigma_0\}_{n=1}^N, \{\widehat{\sigma}_0\}_{n=1}^N) = \frac{1}{N}\sum_{n=1}^N \frac{\|(\sigma_0)_n - (\widehat{\sigma}_0)_n\|}{\|(\sigma_0)_n\|}$$

Inaccessible Loss Function:

$$L_2(\{\sigma_0\}_{n=1}^N, \{\widehat{\sigma}_0\}_{n=1}^N, \{\xi\}_{n=1}^N, \{\widehat{\xi}_0\}_{n=1}^N) = \frac{1}{N}\sum_{n=1}^N \left(\frac{\|(\sigma_0)_n - (\widehat{\sigma}_0)_n\|}{\|(\sigma_0)_n\|} + \frac{\|(\xi)_n - (\widehat{\xi})_n\|}{\|(\xi)_n\|_{L^2(\Omega,\mathbb{R})}} \right)$$

For each of the following RNOs, the architecture for $\mathcal{F}_{\text{RNO}}$ and $\mathcal{G}_{\text{RNO}}$ consists of three internal layers of SeLU units of 100 nodes separated by linear layers, all followed by a final linear layer. The SELU function is applied element-wise as

$$\text{SELU}(x) = s(\max(0, x) + \min(0, \alpha(\exp(x) - 1)))$$

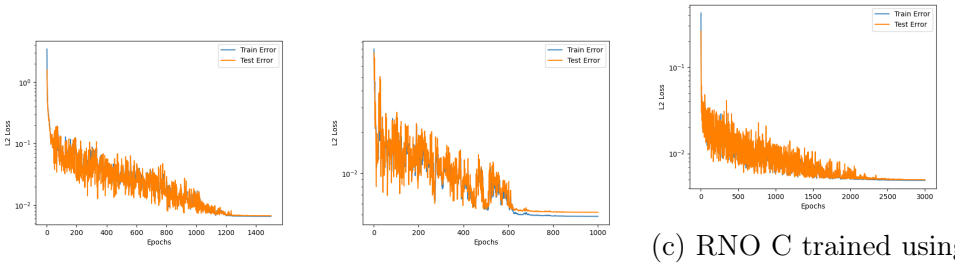
where $\alpha = 1.67326$ and $s = 1.05070$[1]. We trained three different RNOs on the same piecewise-constant dataset in the following manner:

- **RNO A**: Using only the inaccessible loss function L_2, we trained on $N = 400$ data points with subsampled time discretization of $dt = 0.004$ up to $T = 4$ for 1500 epochs with a batch size of 50.
- **RNO B**: first we used the inaccessible loss function L_2 to train on $N = 200$ data points with subsampled time discretization of $dt = 0.004$ up to $T = 2$ for 1500 epochs with a batch size of 40. Then we initialized a new RNO at the parameters of this RNO and trained with the accessible loss function L_1 for 1000 epochs on 200 data with batch size of 40.

[1] https://pytorch.org/docs/stable/generated/torch.nn.SELU.html

- **RNO C**: Using only the accessible loss function L_1, we trained on $N = 500$ data points with subsampled time discretization of $dt = 0.004$ up to $T = 4$ for 3000 epochs with a batch size of 50.

The train and test errors are shown for the three RNOs in Figure A.1.



(a) RNO A trained using inaccessible loss function

(b) RNO B initialized at inaccessible loss solution

(c) RNO C trained using only standard loss function

Figure A.1: Train and test error for the three RNOs.

The piecewise constant data was generated by solving the cell problem using a finite difference method with 300 spatial nodes and $dt = 0.005$ over a time length of $T = 10$ for the trajectories. In the training, we sliced the data trajectories by a slice of 2. For the 3-piecewise constant model, we trained on 500 data points for 3000 epochs with the squared relative loss function. For the 5 and 10-piecewise constant models, we trained on 600 data points for 4000 epochs with the squared relative loss function. The piecewise constant values were: $E_1 = 2$, $E_2 = 8$, $E_3 = 1$, $E_4 = 3$, $E_5 = 7$, $E_6 = 3$, $E_7 = 5$, $E_8 = 6$, $E_9 = 9$, $E_{10} = 4$, $\nu_1 = 0.1$, $\nu_2 = 0.9$, $\nu_3 = 0.7$, $\nu_4 = 0.4$, $\nu_5 = 1.5$, $\nu_6 = 1.2$, $\nu_7 = 0.5$, $\nu_8 = 1.4$, $\nu_9 = 0.5$, and $\nu_{10} = 0.3$. (first three for 3-piecewise, first five for 5-piecewise constant, all ten for 10-piecewise constant).

RNO Training and Testing: Continuous Case

We trained several RNOs on data $\{(\partial_x u_0)_n, (\sigma_0)_n\}_{n=1}^N$ for continuous material parameters given by $E(y) = 2 + \tanh\left(\frac{y-0.5}{0.2}\right)$ and $\nu(y) = 0.5 + 0.1\tanh\left(\frac{y-0.5}{0.2}\right)$. Each of the four RNOs had 1, 2, 5, and 10 hidden variables (L, or the dimension of ξ) respectively. The data was generated by solving the cell problem using a finite difference method with 200 spatial nodes and $dt = h^2$ where h is the spatial discretization. The RNO was trained for 3000 epochs on 500 data. The macroscale simulations were performed with a spacial discretization of $h_{cell} = 0.05$ and a time discretization of $dt = 0.4h_{cell}^2$. They were compared to

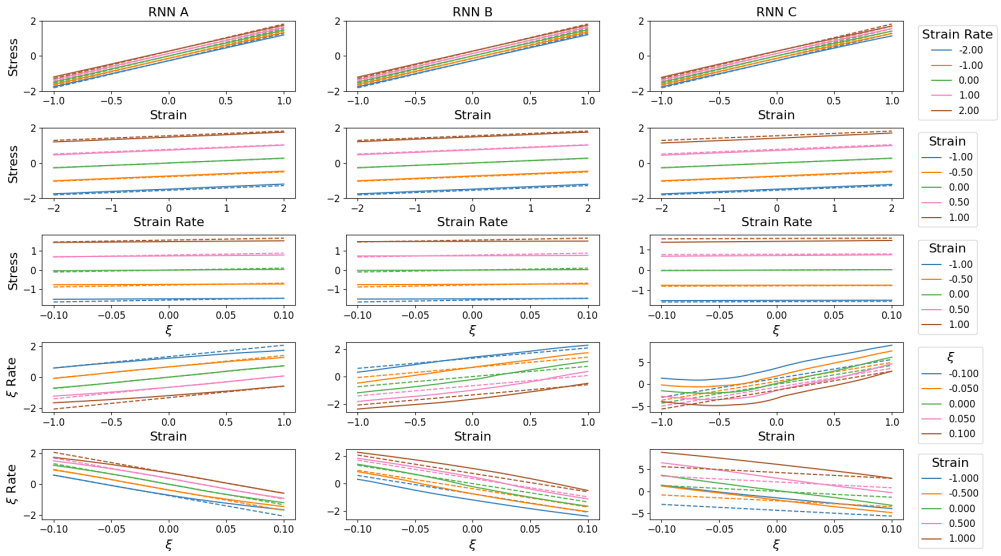


Figure A.2: RNO outputs versus the truth (dashed) for each of the three candidate RNOs. The columns correspond to RNOS A, B, and C respectively. The first row shows the strain-stress dependence for five fixed strain rate inputs. The second row shows the strain rate-stress dependence for five fixed strain inputs. The third row shows the ξ, stress relationships for hidden variable ξ for five fixed strain inputs. The fourth row shows the strain, $\dot{\xi}$ relationship for five different fixed values of ξ. Finally, the fifth row shows the $\xi, \dot{\xi}$ relationship for five fixed strain inputs.

an FEM solution computed with a spacial discretization of $h = 0.004$ with a material period of 0.04 and time discretization of $dt = 0.1h^2$.

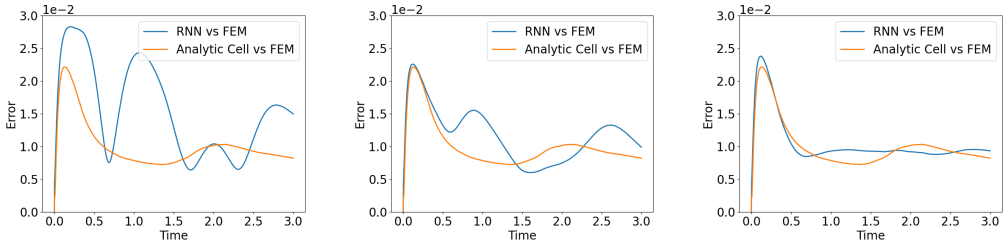
Additional numerical experiments for viscoelasticity

In Figure A.2 we investigate whether the learned models exhibit linearity in the inputs, as is the case for the true equations. The first two columns are models trained with the inaccessible loss function, and the rightmost column is a model trained only on the accessible loss function. All three models exhibit linearity in $\mathcal{F}^{\text{RNO}}$, but only the model trained on the inaccessible loss function exhibits linearity in the learned model for $\mathcal{G}^{\text{RNO}}$. Despite this lack of adherence to the true behavior, the third model achieves comparable test error and performance as a surrogate model.

In Figure A.3 we display the result of using RNOs A, B, and C as surrogate models with integrated Brownian motion forcing. Both the analytic homogenized equations and the RNO predicted stress are compared to the stress computed via an FEM simulation as the ground truth. Once more, the mag-

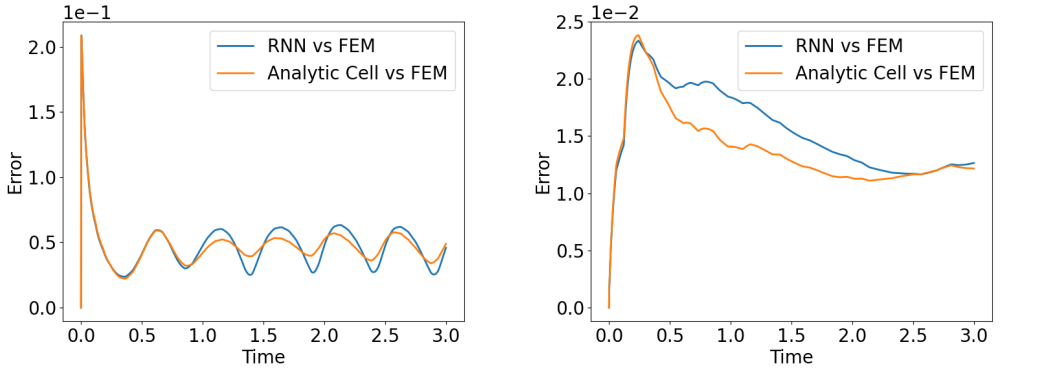
nitudes of the error are similar, implying that the main contributor of the error is homogenization itself.

Finally, in Figure A.4 we train another RNO on piecewise-constant material parameters exhibiting higher inertial effects and test them in macroscale simulations with the same sinusoidal and Brownian motion forcing as before.



(a) RNO A (b) RNO B (c) RNO C

Figure A.3: Analytic cell and RNO relative error versus FEM solution using integrated Brownian motion forcing; this supports Numerical Experiments, conclusion I.



(a) Sinusoidal forcing (b) Integrated Brownian motion forcing

Figure A.4: Relative error of RNO trained on material parameters with higher inertial effects in response to sinusoidal and integrated Brownian motion forcing; this demonstrates Numerical Experiments, conclusion I.

RNO Training and Testing: Elasto-viscoplasticity Experiments

The data for the elasto-viscoplasticity case was generated using a fixed-point iteration scheme with $dt = 0.0001$, 100 spatial elements, and a termination threshold of 0.001. The constant values used were $n = 10$, $E_1 = 5$, $E_2 = 1$, $E_3 = 3$, $E_4 = 2$, $E_5 = 4$, $E_6 = 6$, $E_7 = 1$, $E_8 = 3$, $E_9 = 4$, $\dot{\epsilon}_{p0,1} = 0.05$, $\dot{\epsilon}_{p0,2} = 0.1$, $\dot{\epsilon}_{p0,3} = 0.15$, $\dot{\epsilon}_{p0,4} = 0.07$, $\dot{\epsilon}_{p0,5} = 0.02$, $\dot{\epsilon}_{p0,6} = 0.08$, $\dot{\epsilon}_{p0,7} = 0.04$,

$\dot{\epsilon}_{p0,0.12}$, $\dot{\epsilon}_{p0,0.03}$, and each $\sigma_{0,i} = E_i\dot{\epsilon}_{p0,i}$. The RNO was trained without the strain rate variable on 400 data trajectories with a time slice of 8. The model trained for 3000 epochs with absolute error. The samples shown in Figure 2.6 were chosen using a random number generator. Figure A.5 shows the error for the train and test data points for this elasto-viscoplastic experiment.

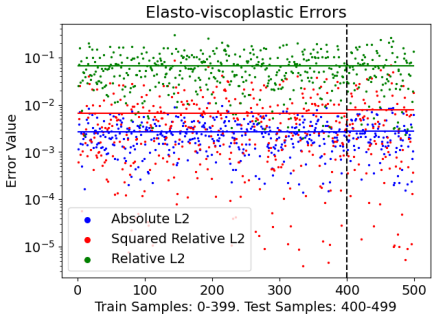


Figure A.5: Error evaluations of all train and test data points for the elasto-viscoplastic experiments. Solid lines indicate mean error values, which are computed separately for the train and test sets.

A.4 One-Dimensional Standard Linear Solid

In this section we address the model of the one-dimensional Maxwell version of the SLS, whose constitutive law depends only on the strain and strain history. The analysis for the SLS model demonstrates that the ideas presented for the KV model extend beyond that particular setting. In Section A.4, we present the governing equations, and in Section A.4 we homogenize the system.

Governing Equations

In the setting without inertia, the displacement u_ε, strain ϵ_ε, and inelastic strain ϵ_ε^p are related by

$$-\partial_x \sigma_\varepsilon = f, \tag{A.4.1a}$$

$$\epsilon_\varepsilon = \partial_x u, \tag{A.4.1b}$$

$$\sigma_\varepsilon = E_{1,\varepsilon}\epsilon_\varepsilon + E_{2,\varepsilon}(\epsilon_\varepsilon - \epsilon_\varepsilon^p), \tag{A.4.1c}$$

$$\partial_t \epsilon_\varepsilon^p = \frac{E_{2,\varepsilon}}{\nu_\varepsilon}(\epsilon_\varepsilon - \epsilon_\varepsilon^p), \tag{A.4.1d}$$

where $f : \Omega \times \mathcal{T} \mapsto \mathbb{R}$ is a known forcing, and we impose initial condition $u(x,0) = 0$ and boundary conditions $u(x,t) = 0$ for $x \in \partial\Omega$. We seek a solution $u_\varepsilon : \Omega \times \mathcal{T} \mapsto \mathbb{R}$. Once more we have small scale dependence in the material properties through ε: we have $E_{i,\varepsilon}(x) = E_i\left(\frac{x}{\varepsilon}\right)$ for $i = 1, 2$ and $\nu_\varepsilon(x) = \nu\left(\frac{x}{\varepsilon}\right)$ for $0 < \varepsilon \ll 1$.

Homogenization

First, we take the Laplace transform of equation (A.4.1) and combine the transformed expressions of equations (A.4.1c) and (A.4.1d) to arrive at

$$\widehat{\sigma}_\varepsilon = E_{1,\varepsilon}\widehat{\epsilon}_\varepsilon + E_{2,\varepsilon} s\left(s + \frac{E_{2,\varepsilon}}{\nu_\varepsilon}\right)^{-1}\widehat{\epsilon}_\varepsilon. \tag{A.4.2}$$

Letting $\widehat{a}(s) = E_{1,\varepsilon} + E_{2,\varepsilon} s\left(s + \frac{E_{2,\varepsilon}}{\nu_\varepsilon}\right)^{-1}$, the homogenization theory of Section 2.3 applies, and we can use the harmonic averaging expression in equation (2.2.5) to write

$$\widehat{a}_0(s) = \left\langle (a(s))^{-1}\right\rangle^{-1} = \left(\int_0^1 \frac{s + \frac{E_2}{\nu}}{(E_1+E_2)s + \frac{E_1E_2}{\nu}}\, \mathsf{d}y\right)^{-1}, \tag{A.4.3}$$

where the homogenized solution u_0 solves system 2.1.6 with $\Psi_0^\dagger$ is defined as

$$\Psi_0^\dagger = \mathcal{L}^{-1}[\widehat{a}_0(s)\partial_x \widehat{u}_0], \tag{A.4.4}$$

analogous to the KV case. However, in the case of piecewise-constant E_1, E_2 and ν the inverse Laplace transform yields a different form in the SLS case:

$$\Psi_0^{\mathrm{PC}}(\partial_x u_0, t; \theta) = E'\partial_x u_0(t) - \sum_{\ell=1}^{L} \xi_\ell(t) \tag{A.4.5a}$$

$$\partial_t \xi_\ell(t) = \beta_\ell \partial_x u_0(t) - \alpha_\ell \xi_\ell(t), \quad \ell \in \{1, \dots, L\} \tag{A.4.5b}$$

for a material with L pieces. Note that this model does not have dependence on the strain rate, but it has one more hidden variable than the piecewise-constant case for the KV model. The value of E' follows from taking the limit $s \to \infty$ and is given by

$$E' = \left(\int_0^1 \frac{1}{(E_1+E_2)}\, \mathsf{d}y\right)^{-1}.$$

SLS Derivation

Here we show that the SLS model has one more hidden variable in the piecewise-constant case than the KV model does. This is the analog of Theorem 2.3.6 for the SLS model. Starting from equation (A.4.3) for $\widehat{a}_0(s)$:

$$\begin{aligned}
\widehat{a}_0(s) &= \langle \widehat{a}(s)^{-1}\rangle^{-1} \\
&= \left(\int_0^1 \frac{s + \frac{E_2(y)}{\nu}}{(E_1(y)+E_2(y))s + \frac{E_1(y)E_2(y)}{\nu(y)}}\, dy\right)^{-1} \\
&\quad \left(\sum_{i=1}^{L} \frac{(s + \frac{E_{2,i}}{\nu_i})d_i}{(E_{1,i}+E_{2,i})s + \frac{E_{1,i}E_{2,i}}{\nu_i}}\right)^{-1}
\end{aligned}$$

for L-piecewise-constant E_1, E_2, and ν with pieces of length d_i. Let $c_i = \frac{E_{2,i}}{\nu_i}$, $k_i = E_{1,i} + E_{2,i}$ and $p_i = \frac{E_{1,i}E_{2,i}}{\nu_i}$. Continuing,

$$= \left(\sum_{i=1}^{L} \frac{(s+c_i)d_i}{k_i s + p_i} \right)^{-1}$$
$$= \frac{\prod_{i=1}^{L}(k_i s + p_i)}{\sum_{i=1}^{L} d_i(s+c_i)\prod_{j\neq i}(k_j s + p_j)} := \frac{P(s)}{Q(s)}.$$

Note that both $P(s)$ and $Q(s)$ have degree L. There is a unique constant E' such that

$$\frac{P(s)}{Q(s)} = E' + \frac{C(s)}{Q(s)},$$

where $C(s)$ has degree L. Then $\frac{C(s)}{Q(s)}$ decomposes uniquely as $\sum_{\ell=1}^{L} \frac{\beta_\ell}{s+\alpha_\ell}$. Note that this is one more pole than the decomposition for the KV model in Theorem 2.3.6 has. We will now show that roots of Q are real and negative, which will lead to the expression in equation (A.4.5). First notice that if the roots of $Q(s)$ are real, then they must be negative since k_i, c_i, d_i, and p_i are strictly positive for all $i \in [L]$. Suppose for the sake of contradiction that $Q(s)$ has a root with a nonzero imaginary component: $s = a + bi$ where $b \neq 0$. Then

$$\begin{aligned}
Q(a+bi) &= \sum_{\ell=1}^{L} d_i(a+bi+c_\ell)\prod_{j\neq\ell}(k_j(a+bi)+p_j) \\
&= \left(\prod_j (k_j(a+bi)+p_j)\right)\sum_{\ell=1}^{L}\frac{d_\ell(a+bi+c_\ell)}{k_\ell(a+bi)+p_\ell} \\
&= \left(\prod_j (k_j(a+bi)+p_j)\right)\sum_{\ell=1}^{L}\left(\frac{d_\ell a + d_\ell c_\ell + d_\ell bi}{k_\ell a + p_\ell + k_\ell bi}\right)\left(\frac{k_\ell a + p_\ell - k_\ell bi}{k_\ell a + p_\ell - k_\ell bi}\right) \\
&= \left(\prod_j (k_j(a+bi)+p_j)\right)\times \\
&\sum_{\ell=1}^{L}\frac{d_\ell}{(k_\ell a+p_\ell)^2+(k_\ell b)^2}\left[\left((a+c_\ell)(k_\ell a+p_\ell)+k_\ell b^2\right)+(-k_\ell b c_\ell + b p_\ell)i\right].
\end{aligned}$$

If $a+bi$ is a root of Q, then we need $b\sum_{\ell=1}^{L}\frac{d_\ell}{(k_\ell a+p_\ell)^2+(k_\ell b)^2}(-k_\ell c_\ell + p_\ell) = 0$. Notice that $-k_\ell c_\ell + p_\ell = -\frac{E_{2,\ell}^2}{\nu_\ell}$, which is strictly negative, so for $b \neq 0$, $\mathrm{Im}(Q(a+bi)) < 0$, which is a contradiction. Therefore, $b = 0$, and all the roots of Q are real and negative. Inverting the Laplace transform, we arrive at equation (A.4.5).

A.5 Fourier Neural Mapping Definition

Here we include the definition of the FNM used in the extension RNO-FNM to include material dependence in the model. This architecture is fully developed and explored in Chapter 5.

***Definition* A.5.1** (Fourier Neural Mapping (FNM))**.** Let the function input $M \in L^2(\mathbb{T}^d; \mathbb{R}^{d_M})$. Define the vector input $v_{\text{in}} \in \mathbb{R}^{d_{\text{in}}^v}$ and vector output $v_{\text{out}} \in \mathbb{R}^{d_{\text{out}}^v}$. Let $x \in \mathbb{T}^d$. Now we define the following layers:

$$
\begin{aligned}
\text{(Vector Lifting)} \quad & S_v : \mathbb{R}^{d_{\text{in}}^v} \to \mathbb{R}^{d_{\text{lift}}^v} \\
\text{(Vector to Function)} \quad & \mathscr{D} : \mathbb{R}^{d_{\text{lift}}^v} \to L^2(\mathbb{T}^d; \mathbb{R}^{d_{\text{lift}}^{vf}}) \\
& z \mapsto \mathscr{D}z = \kappa_v(\cdot) z \\
& z \mapsto \mathscr{D}z = \left\{ \textstyle\sum_{k \in \mathbb{Z}^d} \left(P_v^{(k)} z \right)_j \psi_k \right\}_{j \in [d_{\text{lift}}^{vf}]} \\
\text{(Function Lifting)} \quad & S_f : L^2(\mathbb{T}^d; \mathbb{R}^{d_M + d_{\text{lift}}^{vf}}) \to L^2(\mathbb{T}^d; \mathbb{R}^{d_0}) \\
\text{(Fourier)} \quad & \mathscr{L}_t : L^2(\mathbb{T}^d; \mathbb{R}^{d_{t-1}}) \to L^2(\mathbb{T}^d; \mathbb{R}^{d_t}),\ t \in [T], \\
& \big(\mathscr{L}_t(u)\big)(x) = \sigma\big(W_t u(x) + (\mathcal{K}_t u)(x) + b_t\big), \\
\text{(Function to Vector)} \quad & \mathscr{G} : L^2(\mathbb{T}^d; \mathbb{R}^{d_T}) \to \mathbb{R}^{d_{\text{proj}}^{fv}} \\
& h \mapsto \mathscr{G}h = \textstyle\int_{\mathbb{T}^d} \kappa_f(x) h(x)\, \mathsf{d}x \\
& h \mapsto \mathscr{G}h = \left\{ \textstyle\sum_{k \in \mathbb{Z}^d} \left(\sum_{j=1}^{d_T} (P_f^{(k)})_{\ell j} \langle \psi_k, h_j \rangle_{L^2(\mathbb{T}^d; \mathbb{C})} \right) \right\}_{\ell \in [d_{\text{proj}}^{fv}]} \\
\text{(Vector Projection)} \quad & Q_v : \mathbb{R}^{d_{\text{proj}}^{fv}} \to \mathbb{R}^{d_{\text{out}}^v}.
\end{aligned}
$$

The convolution operator is given, for $u : \mathbb{T}^d \to \mathbb{R}^{d_{t-1}}$ and $x \in \mathbb{T}^d$, by

$$
(\mathcal{K}_t u)(x) = \left\{ \sum_{k \in \mathbb{Z}^d} \left(\sum_{j=1}^{d_{t-1}} (P_t^{(k)})_{\ell j} \langle \psi_k, u_j \rangle_{L^2(\mathbb{T}^d; \mathbb{C})} \right) \psi_k(x) \right\}_{\ell \in [d_t]} \in \mathbb{R}^{d_t}. \quad \text{(A.5.1)}
$$

For given layer index t and wave vector $k \in \mathbb{Z}^d$, the matrix $P_t^{(k)} \in \mathbb{C}^{d_t \times d_{t-1}}$ comprises learnable parameters of the integral operator $\mathcal{K}_t$; furthermore, $W_t \in \mathbb{R}^{d_t \times d_{t-1}}$ is a weights matrix, $b_t \in \mathbb{R}^{d_t}$ is a bias vector, both learnable. And, for given wave vector $k \in \mathbb{Z}^d$, $P_v^{(k)} \in \mathbb{C}^{d_{\text{lift}}^{vf} \times d_{\text{lift}}^v}$ are the learnable parameters of the vector to function map $\mathscr{D}$, and $P_f^{(k)} \in \mathbb{C}^{d_{\text{proj}}^{fv} \times d_T}$ are the learnable parameters of the function to vector map $\mathscr{G}$. The vector lifting and projection layers, S_v and Q_v, are either shallow neural networks or linear maps, and hence also contain learnable parameters. Finally the function lifting layer S_f is applied pointwise

in $\mathbb{T}^d-$a.e. and is also defined by either a shallow neural network or a linear map containing learnable parameters. $\Diamond$

Appendix B

APPENDIX TO CHAPTER 3

Links to datasets and all code used to produce the numerical results and figures in this chapter are available at

https://github.com/mtrautner/LearningHomogenization/ .

B.1 Proofs of Stability Estimates

In this section, we prove the stability estimates stated in Propositions 3.1.2 and 3.1.3. The following lemma is a modification of the standard estimate for parametric dependence of elliptic equations on their coefficient. We include it here for completeness.

***Proposition* 3.1.1.** Consider the cell problem defined by equation (3.1.4). The following hold:

1. If $A \in \mathsf{PD}_{\alpha,\beta}$, then (3.1.4) has a unique solution $\chi \in \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)$ and
$$\|\chi\|_{\dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)} \leq \frac{\sqrt{d}\beta}{\alpha}.$$
2. For $\chi^{(1)}$ and $\chi^{(2)}$ solutions to the cell problem in equation (3.1.4) associated with coefficients $A^{(1)}, A^{(2)} \in \mathsf{PD}_{\alpha,\beta}$, respectively, it follows that
$$\|\chi^{(2)} - \chi^{(1)}\|_{\dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)} \leq \frac{\sqrt{d}}{\alpha}\left(1 + \frac{\beta}{\alpha}\right)\|A^{(1)} - A^{(2)}\|_{L^\infty(\mathbb{T}^d;\mathbb{R}^{d\times d})}. \quad (3.1.8)$$

◊

Proof. For existence and uniqueness of the solution to the cell problem using Lax-Milgram, we refer to the texts [59, 27]; we simply derive the bounds and stability estimate. First, note that (3.1.4) decouples, in particular,

$$-\nabla\cdot(\nabla\chi_\ell A) = \nabla\cdot(Ae_\ell), \qquad y \in \mathbb{T}^d \quad (B.1.1)$$

for $l = 1, \dots, d$ where e_ℓ is the ℓ-th standard basis vector of $\mathbb{R}^d$ and each $\chi_\ell \in \dot{H}^1(\mathbb{T}^d; \mathbb{R})$. Multiplying by χ_ℓ and integrating by parts shows

$$\begin{aligned}
\alpha \|\nabla \chi_\ell\|_{L^2}^2 &\leq \int_{\mathbb{T}^d} \langle A \nabla \chi_\ell, \nabla \chi_\ell \rangle \, \mathsf{d}y \\
&= -\int_{\mathbb{T}^d} \langle A e_\ell, \nabla \chi_\ell \rangle \, \mathsf{d}y \\
&\leq \int_{\mathbb{T}^d} |A e_\ell| |\nabla \chi_\ell| \, \mathsf{d}y \\
&\leq \left(\int_{\mathbb{T}^d} |A e_\ell|^2 \, \mathsf{d}y \right)^{\frac{1}{2}} \left(\int_{\mathbb{T}^d} |\nabla \chi_\ell|^2 \, \mathsf{d}y \right)^{\frac{1}{2}} \\
&\leq \|A\|_{L^\infty} \|\nabla \chi_\ell\|_{L^2}.
\end{aligned}$$

Therefore

$$\|\nabla \chi\|_{L^2}^2 = \sum_{l=1}^{d} \|\nabla \chi_\ell\|_{L^2}^2 \leq \frac{d \|A\|_{L^\infty}^2}{\alpha^2} \leq \frac{d\beta^2}{\alpha^2},$$

which implies the first result.

To prove the second result, we denote the right hand side of B.1.1 by $f_\ell^{(i)} = \nabla \cdot A^{(i)} e_\ell$ in what follows. For any $v \in \dot{H}^1(\mathbb{T}^d; \mathbb{R})$, we have that

$$\begin{aligned}
-\int_{\mathbb{T}^d} \nabla \cdot (A^{(1)} \nabla \chi_\ell^{(1)}) v \;\; \mathsf{d}y &= \int_{\mathbb{T}^d} f_\ell^{(1)} v \;\; \mathsf{d}y \\
-\int_{\partial \mathbb{T}^d} v A^{(1)} \nabla \chi_\ell^{(1)} \cdot \widehat{n} \;\; \mathsf{d}y + \int_{\mathbb{T}^d} \nabla v \cdot A^{(1)} \nabla \chi_\ell^{(1)} \;\; \mathsf{d}y &= \int_{\mathbb{T}^d} f_\ell^{(1)} v \;\; \mathsf{d}y.
\end{aligned}$$

Since v, $A^{(1)}$, and the solution $\chi_\ell^{(1)}$ are all periodic on $\mathbb{T}^d$, the first term is 0. Combining with the equation for $\chi_\ell^{(2)}$, we get

$$\begin{aligned}
&\int_{\mathbb{T}^d} \nabla v \cdot \left(A^{(1)} - A^{(2)} \right) \nabla \chi_\ell^{(1)} \;\; \mathsf{d}y = \\
&\qquad = \int_{\mathbb{T}^d} (f_\ell^{(1)} - f_\ell^{(2)}) v + \nabla v \cdot \left(A^{(2)} \left(\nabla \chi_\ell^{(2)} - \nabla \chi_\ell^{(1)} \right) \right) \mathsf{d}y.
\end{aligned}$$

Setting $v = \chi_\ell^{(2)} - \chi_\ell^{(1)}$, we have

$$\begin{aligned}
\int_{\mathbb{T}^d} \left(\nabla \chi_\ell^{(2)} - \nabla \chi_\ell^{(1)} \right) \cdot \left(\left(A^{(1)} - A^{(2)} \right) \nabla \chi_\ell^{(1)} \right) \mathsf{d}y &= \int_{\mathbb{T}^d} (f_\ell^{(1)} - f_\ell^{(2)}) \left(\chi_\ell^{(2)} - \chi_\ell^{(1)} \right) \mathsf{d}y \\
&\quad + \int_{\mathbb{T}^d} \left(\nabla \chi_\ell^{(2)} - \nabla \chi_\ell^{(1)} \right) \cdot \left(A^{(2)} \left(\nabla \chi_\ell^{(2)} - \nabla \chi_\ell^{(1)} \right) \right) \mathsf{d}y, \\
\alpha \|\nabla \chi_\ell^{(2)} - \nabla \chi_\ell^{(1)}\|_{L^2}^2 &\leq \|A^{(1)} - A^{(2)}\|_{L^\infty} \|\nabla \chi_\ell^{(1)}\|_{L^2} \|\nabla \chi_\ell^{(2)} - \nabla \chi_\ell^{(1)}\|_{L^2} \\
&\quad + \|f_\ell^{(1)} - f_\ell^{(2)}\|_{\dot{H}^{-1}} \|\nabla \chi_\ell^{(2)} - \nabla \chi_\ell^{(1)}\|_{L^2},
\end{aligned}$$

$$\|\chi_\ell^{(2)} - \chi_\ell^{(1)}\|_{\dot{H}^1} \leq \frac{1}{\alpha}\Big(\|A^{(1)} - A^{(2)}\|_{L^\infty}\|\nabla\chi_\ell^{(1)}\|_{L^2} + \|f_\ell^{(1)} - f_\ell^{(2)}\|_{\dot{H}^{-1}}\Big). \quad \text{(B.1.2)}$$

Evaluating,

$$\begin{aligned}
\|f_\ell^{(1)} - f_\ell^{(2)}\|_{\dot{H}^{-1}} &= \|\nabla\cdot A^{(1)}e_\ell - \nabla\cdot A^{(2)}e_\ell\|_{\dot{H}^{-1}}, && \text{(B.1.3)}\\
&= \sup_{\|\xi\|_{\dot{H}^1}=1}\int_{\mathbb{T}^d}\nabla\cdot(A^{(1)} - A^{(2)})e_\ell\xi \;\; \mathsf{d}y, && \text{(B.1.4)}\\
&\leq \sup_{\|\xi\|_{\dot{H}^1}=1}\|(A^{(1)} - A^{(2)})e_\ell\|_{L^2}\|\nabla\xi\|_{L^2}, && \text{(B.1.5)}\\
&\leq \|A^{(1)} - A^{(2)}\|_{L^2} \leq \|A^{(1)} - A^{(2)}\|_{L^\infty} && \text{(B.1.6)}
\end{aligned}$$

since our domain is $\mathbb{T}^d$. Combining this with (B.1.2) and the bound of $\|\nabla\chi_\ell\|_{L^2} \leq \frac{\beta}{\alpha}$ obtained in the first part of this proposition, we have

$$\|\chi_\ell^{(2)} - \chi_\ell^{(1)}\|_{\dot{H}^1} \leq \frac{1}{\alpha}\left(1 + \frac{\beta}{\alpha}\right)\|A^{(1)} - A^{(2)}\|_{L^\infty}. \quad \text{(B.1.7)}$$

Returning to d vector components yields the result. □

The following result shows that the mapping $A \mapsto \overline{A}$ is continuous on separable subspaces of $L^\infty(\mathbb{T}^d;\mathbb{R}^{d\times d})$.

***Lemma* B.1.1.** Let $\mathbf{A} \subset L^\infty(\mathbb{T}^d;\mathbb{R}^{d\times d})$ be a separable subspace and $K \subset \mathbf{A}\cap\mathsf{PD}_{\alpha,\beta}$ a closed set in L^∞. Define the mapping $F : K \to \mathbb{R}^{d\times d}$ by $A \mapsto \overline{A}$ as given by (3.1.3). Then there exists a continuous mapping $\mathcal{F} \in C(\mathbf{A};\mathbb{R}^{d\times d})$ such that $\mathcal{F}(A) = F(A)$ for any $A \in K$. ◊

Proof. Let $A^{(1)}, A^{(2)} \in K$ then, by Proposition 3.1.1,

$$\begin{aligned}
\big|F\big(A^{(1)}\big) - F\big(A^{(2)}\big)\big|_F &\leq \int_{\mathbb{T}^d}|A^{(1)} - A^{(2)}|_F\big(1 + |\nabla\chi^{(1)}|_F\big)\;\mathsf{d}y\\
&\quad + \int_{\mathbb{T}^d}|A^{(2)}|_F|\nabla\chi^{(1)} - \nabla\chi^{(2)}|_F\;\mathsf{d}y\\
&\leq \|A^{(1)} - A^{(2)}\|_{L^\infty}\big(1 + \|\nabla\chi^{(1)}\|_{L^2}\big) + \|A^{(2)}\|_{L^\infty}\|\nabla\chi^{(1)} - \nabla\chi^{(2)}\|_{L^2}\\
&\leq \left(1 + \frac{\sqrt{d}}{\alpha}\left(\|A^{(1)}\|_{L^\infty} + \|A^{(2)}\|_{L^\infty}\left(\frac{\min\big(\|A^{(1)}\|_{L^\infty}, \|A^{(2)}\|_{L^\infty}\big)}{\alpha} + 1\right)\right)\right)\\
&\quad \cdot \|A^{(1)} - A^{(2)}\|_{L^\infty},
\end{aligned}$$

and hence $F \in C(K;\mathbb{R}^{d\times d})$. Applying the Tietze extension theorem [182] to F implies the existence of $\mathcal{F}$. □

The following lemma shows that, unfortunately, separable subspaces of $L^\infty(\mathbb{T}^d;\mathbb{R}^{d\times d})$ are not very useful. Indeed, in the desired area of application of continuum mechanics, we ought to be able to place a boundary of material discontinuity anywhere in the domain. The following result shows that doing so is impossible for a subset of $\mathsf{PD}_{\alpha,\beta}$ which lies only in a separable subspace of $L^\infty(\mathbb{T}^d;\mathbb{R}^{d\times d})$.

***Lemma* B.1.2.** For any $t \in [0,1]$ define $c_t : [0,1] \to \mathbb{R}$ by

$$c_t(x) = \begin{cases} 1, & x \leq t \\ 0, & x > t \end{cases}, \qquad \forall\, x \in [0,1].$$

Define $E = \{c_t : t \in [0,1]\} \subset L^\infty([0,1])$. There exists no separable subspace $\mathbf{A} \subset L^\infty([0,1])$ such that $E \subseteq \mathbf{A}$. ◊

Proof. Suppose otherwise. Since $(\mathbf{A}, \|\cdot\|_{L^\infty})$ is a separable metric space, $(E, \|\cdot\|_{L^\infty})$ must be separable since $E \subseteq \mathbf{A}$; this is a contradiction since $(E, \|\cdot\|_{L^\infty})$ is not separable. To see this, let $\{c_{t_j}\}_{j=1}^\infty$ be an arbitrary countable susbset of E. Then for any $t \notin \{t_j\}_{j=1}^\infty$, we have

$$\inf_{\{t_j\}_{j=1}^\infty} \|c_t - c_{t_j}\|_{L^\infty} = 1.$$

Hence no countable subset can be dense. □

Instead of working on a compact subset of a separable subspace of $L^\infty(\mathbb{T}^d;\mathbb{R}^{d\times d})$, we may instead try to find a suitable probability measure which contains the discontinous functions of interest. The following remarks makes clear why such an approch would still be problematic for the purposes of approximation.

***Remark* B.1.3** (Gaussian Threshholding)**.** Let μ be a Gaussian measure on $L^2([0,1])$. Define

$$T(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}, \qquad \forall\, x \in [0,1]$$

and consider the corresponding Nemytskii operator $N_T : L^2([0,1]) \to L^\infty([0,1])$. Then, working with the definitions in Lemma B.1.2, it is easy to see that $E \subset \text{supp} N_T{}^\sharp \mu$. Therefore there exists no separable subspace of $L^\infty([0,1])$ which contains $\text{supp} N_T{}^\sharp \mu$. ◊

We therefore abandon L^∞ and instead show continuity and Lipschitz continuity for some L^q with $q < \infty$ to $\dot{H}^1$. The following lemma is a general result for convergence of sequences in metric spaces which is used in a more specific context in the next lemma.

***Lemma* B.1.4.** Let (M, d) be a metric space and $(a_n) \subset M$ a sequence. If every subsequence $(a_{n_k}) \subset (a_n)$ contains a subsequence $(a_{n_{k_l}}) \subset (a_{n_k})$ such that $(a_{n_{k_l}}) \to a \in M$ then $(a_n) \to a$. ◊

Proof. Suppose otherwise. Then, there exists some $\epsilon > 0$ such that, for every $N \in \mathbb{Z}^+$, there exists some $n = n(N) > N$ such that

$$d(a_n, a) \geq \epsilon.$$

Then we can construct a subsequence $(a_{n_j}) \subset (a_n)$ such that $d(a_{n_j}, a) \geq \epsilon \; \forall n_j$. Therefore a_{n_j} does not have a subsequence converging to a, which is a contradiction. □

The following lemma proves existence of a limit in $L^2(D; \mathbb{R}^d)$ of a sequence of outputs of operators in $L^\infty(D; \mathbb{R}^{d\times d})$.

***Lemma* B.1.5.** Let $D \subseteq \mathbb{R}^d$ be an open set and $(A_n) \subset L^\infty(D; \mathbb{R}^{d\times d})$ a sequence satisfying the following.

1. $A_n \in \mathsf{PD}_{\alpha,\beta}$ for all n.
2. There exists $A \in L^\infty(D; \mathbb{R}^{d\times d})$ such that $(A_n) \to A$ in $L^2(D; \mathbb{R}^{d\times d})$.

Then, for any $g \in L^2(D; \mathbb{R}^d)$, we have that $(A_n g) \to Ag$ in $L^2(D; \mathbb{R}^d)$. ◊

Proof. We have

$$\|A_n g\|_{L^2} \leq \beta \|g\|_{L^2},$$

and hence $(A_n g) \subset L^2(D; \mathbb{R}^d)$ and, similarly, by finite-dimensional norm equivalence, there is a constant $C_1 > 0$ such that

$$\|Ag\|_{L^2} \leq C_1 \|A\|_{L^\infty} \|g\|_{L^2},$$

and hence $Ag \in L^2(D;\mathbb{R}^d)$. Again, by finite-dimensional norm equivalence, we have that there exists a constant $C_2 > 0$ such that, for $j \in \{1,\dots,d\}$ and almost every $y \in D$, we have

$$(A_n g)_j(y)^2 \le |A_n^{(j)}(y)|^2 |g(y)|^2 \le C_2 \beta^2 |g(y)|^2,$$

where $A_n^{(j)}(y)$ denotes the j-th row of $A_n^{(j)}(y)$. In particular,

$$|(A_n g)_j(y)| \le \sqrt{C_2}\beta |g(y)|.$$

Let $(A_{n_k}) \subset (A_n)$ be an arbitrary subsequence. Since $(A_n) \to A$, we have that $(A_{n_k}) \to A$ in $L^2(D;\mathbb{R}^{d\times d})$. Therefore, there exists a subsequence $(A_{n_{k_l}}) \subset (A_{n_k})$ such that $A_{n_{k_l}}(y) \to A(y)$ for almost every $y \in D$. Then $A_{n_{k_l}}(y)g(y) \to A(y)g(y)$ for almost every $y \in D$. Since $|g| \in L^2(\mathbb{R}^d)$, we have, by the dominated convergence theorem, that $(A_{n_{k_l}} g)_j \to (Ag)_j$ in $L^2(D)$ for every $j \in \{1,\dots,d\}$. Therefore $(A_{n_{k_l}} g) \to Ag$ in $L^2(D;\mathbb{R}^d)$. Since the subsequence (A_{n_k}) was arbitrary, Lemma B.1.4 implies the result. □

Finally, we may prove Proposition 3.1.2.

***Proposition* 3.1.2.** Endow $\mathsf{PD}_{\alpha,\beta}$ with the $L^2(\mathbb{T}^d;\mathbb{R}^{d\times d})$ induced topology and let $K \subset \mathsf{PD}_{\alpha,\beta}$ be a closed set. Define the mapping $G : K \to \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)$ by $A \mapsto \chi$ as given by (3.1.4). Then there exists a bounded continuous mapping

$$\mathcal{G} \in C(L^2(\mathbb{T}^d;\ \mathbb{R}^{d\times d}); \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d))$$

such that $\mathcal{G}(A) = G(A)$ for any $A \in K$. ◊

Proof. Consider the PDE

$$-\nabla\cdot(A\nabla u) = \nabla\cdot Ae, \qquad y \in \mathbb{T}^d, \tag{B.1.8}$$

where e is some standard basis vector of $\mathbb{R}^d$. Let $(A_n) \subset K$ be a sequence such that $(A_n) \to A \in K$ in $L^2(\mathbb{T}^d;\mathbb{R}^{d\times d})$. Denote by $u_n \in \dot{H}^1(\mathbb{T}^d)$ the solution to (B.1.8) corresponding to each A_n and by $u \in \dot{H}^1(\mathbb{T}^d)$ the solution corresponding to the limiting A. A similar calculation as in the proof of Proposition 3.1.1 shows

$$\begin{aligned}\alpha\|u_n - u\|_{\dot{H}^1}^2 &\le \int_{\mathbb{T}^d} \langle (A - A_n)(\nabla u + e), \nabla u_n - \nabla u\rangle \,\mathsf{d}y \\ &\le \|u_n - u\|_{\dot{H}^1} \|(A_n - A)(\nabla u + e)\|_{L^2}.\end{aligned}$$

Since $\nabla u + e \in L^2(\mathbb{T}^d;\mathbb{R}^d)$, by Lemma B.1.5, $\big(A_n(\nabla u + e)\big) \to A(\nabla u + e)$ in $L^2(\mathbb{T}^d;\mathbb{R}^d)$ hence $(u_n) \to u$ in $\dot{H}^1(\mathbb{T}^d)$. In particular, the mapping $A \mapsto u$ defined by (B.1.8) is continuous. Since the problem (3.1.4) decouples as shown by (B.1.1), we have that each component mapping $G_l : K \to \dot{H}^1(\mathbb{T}^d)$ defined by $A \mapsto \chi_\ell$ is continuous thus G is continuous. Applying the Tietze extension theorem [182] to G implies the existence of $\mathcal{G}$. □

The following is a straightforward consequence of Proposition 3.1.2 that establishes continuity of the map $A \mapsto \overline{A}$ defined in (3.1.3) as well.

***Lemma* B.1.6.** Endow $\mathsf{PD}_{\alpha,\beta}$ with the $L^2(\mathbb{T}^d;\mathbb{R}^{d\times d})$ induced topology and let $K \subset \mathsf{PD}_{\alpha,\beta}$ be a closed set. Define the mapping $F : K \to \mathbb{R}^{d\times d}$ by $A \mapsto \overline{A}$ as given by (3.1.3). Then there exists a bounded continuous mapping $\mathcal{F} \in C\big(L^2(\mathbb{T}^d;\mathbb{R}^{d\times d});\mathbb{R}^{d\times d}\big)$ such that $\mathcal{F}(A) = F(A)$ for any $A \in K$. ◊

Proof. Since $\nabla : \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d) \to L^2(\mathbb{T}^d;\mathbb{R}^{d\times d})$ is a bounded operator, Lemma 3.1.2 implies that the mapping $A \mapsto A + A\nabla\chi^T$ is continuous as compositions, sums, and products of continuous functions are continuous. Now let $A \in \mathsf{PD}_{\alpha,\beta}$ then $A \in L^1(\mathbb{T}^d;\mathbb{R}^{d\times d})$ since $A \in L^\infty(\mathbb{T}^d;\mathbb{R}^{d\times d})$. Thus

$$\left|\int_{\mathbb{T}^d} A \,\mathsf{d}y\right|_F \leq \int_{\mathbb{T}^d} |A|_F \,\mathsf{d}y \leq \|A\|_{L^2}$$

by Hölder's inequality and the fact that $\int_{\mathbb{T}^d} \mathsf{d}y = 1$. Hence $F \in C(K;\mathbb{R}^{d\times d})$ as a composition of continuous maps. Again applying the Tietze extension theorem [182] to F implies the existence of $\mathcal{F}$. □

To prove Proposition 3.1.3, we need to establish Lipschitz continuity. We first establish the following result, which is similar to the one proved in [62] in Theorem 2.1. We show it again here both for completeness and because we specialize to the case of the cell problem (3.1.4) with periodic boundary conditions rather than the system (3.1.1) with Dirichlet boundary conditions.

***Lemma* B.1.7.** Let $A^{(1)}, A^{(2)} \in \mathsf{PD}_{\alpha,\beta}$ and let $\chi^{(1)}, \chi^{(2)}$ be the corresponding solutions to (3.1.4). Then

$$\|\chi^{(1)} - \chi^{(2)}\|_{\dot{H}^1} \leq \frac{\sqrt{d}}{\alpha}\big(\|A^{(2)} - A^{(1)}\|_{L^2} + \|\nabla\chi^{(2)}\|_{L^p}\|A^{(2)} - A^{(1)}\|_{L^q}\big) \quad \text{(B.1.9)}$$

for $p \geq 2$ and $q = \frac{2p}{p-2}$. ◊

Proof. As in the proof of Proposition 3.1.1, we denote $f^{(i)} = \nabla \cdot A^{(i)}$ for $i \in \{1, 2\}$ for simplicity of notation and to be easily comparable to the proof of Theorem 2.1 in [62]. Since both sides of the cell problem equation (3.1.4) depend on $A^{(i)}$, we introduce $\widetilde{\chi}$ as the solution of

$$-\nabla \cdot \left(\nabla \widetilde{\chi} A^{(2)}\right) = \nabla \cdot A^{(1)}, \quad \widetilde{\chi} \in \dot{H}^1(\mathbb{T}^d; \mathbb{R}^d) \tag{B.1.10}$$

as an intermediate function. We obtain bounds using $\widetilde{\chi}$ and apply the triangle inequality to

$$\|(\chi^{(1)} - \widetilde{\chi}) + (\widetilde{\chi} - \chi^{(2)})\|_{\dot{H}^1}$$

to obtain a bound on $\|\chi^{(1)} - \chi^{(2)}\|_{\dot{H}^1}$. From the naïve perturbation bound in (B.1.2) we have

$$\|\widetilde{\chi}_\ell - \chi_\ell^{(2)}\|_{\dot{H}^1} \leq \frac{1}{\alpha} \|f_\ell^{(1)} - f_\ell^{(2)}\|_{\dot{H}^{-1}},$$

so we are left to bound $\|\chi_\ell^{(1)} - \widetilde{\chi}_\ell\|_{\dot{H}^1}$. We note that

$$\nabla \cdot \left(A^{(2)} \nabla \widetilde{\chi}_\ell\right) = \nabla \cdot \left(A^{(1)} \nabla \chi_\ell^{(1)}\right)$$

$$\int_{\mathbb{T}^d} A^{(2)} \nabla \widetilde{\chi}_\ell \cdot \nabla v \; \mathsf{d}y = \int_{\mathbb{T}^d} A^{(1)} \nabla \chi_\ell^{(1)} \cdot \nabla v \; \mathsf{d}y \quad \forall v \in \dot{H}^1(\mathbb{T}^d; \mathbb{R}).$$

Letting $v = \chi_\ell^{(1)} - \widetilde{\chi}_\ell$,

$$\int_{\mathbb{T}^d} A^{(2)} \nabla \widetilde{\chi}_\ell \cdot \left(\nabla \chi_\ell^{(1)} - \nabla \widetilde{\chi}_\ell\right) \mathsf{d}y = \int_{\mathbb{T}^d} A^{(1)} \nabla \chi_\ell^{(1)} \cdot \left(\nabla \chi_\ell^{(1)} - \nabla \widetilde{\chi}_\ell\right) \mathsf{d}y$$

$$\int_{\mathbb{T}^d} A^{(2)} \left(\nabla \widetilde{\chi}_\ell - \nabla \chi_\ell^{(1)}\right) \cdot \left(\nabla \widetilde{\chi}_\ell - \nabla \chi_\ell^{(1)}\right) \mathsf{d}y$$

$$= \int_{\mathbb{T}^d} (A^{(2)} - A^{(1)}) \nabla \chi_\ell^{(1)} \cdot \left(\nabla \chi_\ell^{(1)} - \nabla \widetilde{\chi}_\ell\right) \mathsf{d}y$$

$$\alpha \|\widetilde{\chi}_\ell - \chi_\ell^{(1)}\|_{\dot{H}^1} \leq \|(A^{(2)} - A^{(1)})(\nabla \chi_\ell^{(1)})\|_{L^2}.$$

Applying Hölder for L^2, we get

$$\|\widetilde{\chi}_\ell - \chi_\ell^{(1)}\|_{\dot{H}^1} \leq \frac{1}{\alpha} \|\nabla \chi_\ell^{(1)}\|_{L^p} \|A^{(2)} - A^{(1)}\|_{L^q} \tag{B.1.11}$$

for $q = \frac{2p}{p-2}$ where $p \in [2, \infty]$. Putting the two parts together, we have that

$$\|\chi_\ell^{(2)} - \chi_\ell^{(1)}\|_{\dot{H}^1} \leq \frac{1}{\alpha} \|\nabla \cdot A^{(2)} e_\ell - \nabla \cdot A^{(1)} e_\ell\|_{\dot{H}^{-1}} + \frac{1}{\alpha} \|\nabla \chi_\ell^{(1)}\|_{L^p} \|A^{(2)} - A^{(1)}\|_{L^q}$$

$$\leq \frac{1}{\alpha} \|A^{(2)} - A^{(1)}\|_{L^2} + \frac{1}{\alpha} \|\nabla \chi_\ell^{(1)}\|_{L^p} \|A^{(2)} - A^{(1)}\|_{L^q}.$$

Combining bounds for all d dimensions yields the result. □

***Remark* B.1.8.** Since $L^q(\Omega) \hookrightarrow L^2(\Omega)$ for bounded $\Omega \subset \mathbb{R}^d$ and $q \geq 2$, we could also write the bound of Lemma B.1.7 as

$$\|\chi_\ell^{(2)} - \chi_\ell^{(1)}\|_{\dot{H}^1} \leq \frac{1}{\alpha}\Big(C + \|\nabla\chi_\ell^{(1)}\|_{L^p}\Big)\|A^{(2)} - A^{(1)}\|_{L^q}$$

for some C dependent only on q and Ω. ◊

The result of Lemma B.1.7 is unhelpful if $\|\nabla\chi\|_{L^p}$ is unbounded. In this setting, it is not possible for Lemma B.1.7 to result in Lipschitz continuity as a map from L^2 to $\dot{H}^1$. Instead, we seek to bound $\|\nabla\chi\|_{L^p}$ for some p satisfying $2 < p < \infty$.

Before continuing, we establish a bound on the gradient of the solution to the Poisson equation on the torus. This follows the strategy of [62] for the Dirichlet problem. In order to avoid extra factors of 2π in all formulae, we work on the rescaled torus denoted $\mathbb{Y}^d = [0, 2\pi]^d$ with opposite faces identified for the following result of Lemma B.1.9. As we work on the torus, it is useful to first set up notation for the function spaces of interest. Let

$$\mathcal{D}(\mathbb{Y}^d) = C_c^\infty(\mathbb{Y}^d) = C^\infty(\mathbb{Y}^d)$$

be the space of test functions where the last equality follows from compactness of the torus. Functions can be either $\mathbb{R}$ or $\mathbb{C}$ valued hence we do not explicitly specify the range. We equip $\mathcal{D}(\mathbb{Y}^d)$ with a locally convex topology generated by an appropriate family of semi-norms, see, for example, [183, Section 3.2.1]. Any function $g \in \mathcal{D}(\mathbb{Y}^d)$ can be represented by its Fourier series

$$g(x) = \sum_{k\in\mathbb{Z}^d} \widehat{g}(k)\mathrm{e}^{ix\cdot k},$$

where $\widehat{g}$ denotes the Fourier transform of g and convergence of the right-hand side sum is with respect to the topology of $\mathcal{D}(\mathbb{Y}^d)$, and i denotes the imaginary unit. It holds that $\widehat{g} \in \mathcal{S}(\mathbb{Z}^d)$, the Schwartz space of rapidly decreasing functions on the integer lattice, so we have

$$|\widehat{g}(k)| \leq c_m(1 + |k|)^{-m}, \quad m = 0, 1, \ldots$$

for some constants c_m. We may then define the topological (continuous) dual space of $\mathcal{D}(\mathbb{Y}^d)$, the space of distributions, denoted $\mathcal{D}'(\mathbb{Y}^d)$, which can be described as follows: the condition that $f \in \mathcal{D}'(\mathbb{Y}^d)$ is characterized by the property

$$|\widehat{f}(k)| \leq b_m(1 + |k|)^m, \quad m = 0, 1, \ldots$$

for some constants b_m. We take the weak-* topology on $\mathcal{D}'(\mathbb{Y}^d)$ and generally use the prime notation for any such dual space. For any $-\infty < s < \infty$, we define the fractional Laplacian as

$$(-\Delta)^s f = \sum_{k\in\mathbb{Z}^d\setminus\{0\}} |k|^{2s}\widehat{f}(k)\mathrm{e}^{ik\cdot x}, \tag{B.1.12}$$

where the right-hand side sum converges in the topology of $\mathcal{D}'(\mathbb{Y}^d)$. It is easy to see that $(-\Delta)^s : \mathcal{D}'(\mathbb{Y}^d) \to \mathcal{D}'(\mathbb{Y}^d)$ is continuous. Furthermore, for any $j \in \{1,\dots,d\}$, we define the family of operators $\widetilde{R}_j : \mathcal{D}'(\mathbb{Y}^d) \to \mathcal{D}'(\mathbb{Y}^d)$, defining periodic Riesz transforms, by

$$\widetilde{R}_j f = \sum_{k\in\mathbb{Z}^d} -\frac{ik_j}{|k|}\widehat{f}(k)\mathrm{e}^{ik\cdot x}, \tag{B.1.13}$$

where we identify $\frac{k_j}{|k|}|_{k=0} = \lim_{|k|\to 0}\frac{k_j}{|k|} = 0$. Again, we stress that convergence of the right-hand side sum is in the topology of $\mathcal{D}'(\mathbb{Y}^d)$. Lastly, we denote by $\mathcal{S}(\mathbb{R}^d)$ and $\mathcal{S}'(\mathbb{R}^d)$ the Schwartz space and the space of tempered distributions on $\mathbb{R}^d$ respectively; see, for example, [184, Chapter 1] for the precise definitions.

The following lemma establishes boundedness of the periodic Riesz transform on $L^p(\mathbb{Y}^d)$. It is essential in proving boundedness of the gradient to the solution of the Poisson equation on the torus. The result is essentially proven in [184]. We include it here, in our specific torus setting, giving the full argument for completeness.

***Lemma* B.1.9.** There exists a constant $c = c(d,p) > 0$ such that, for any $j \in \{1,\dots,d\}$ and any $f \in L^p(\mathbb{Y}^d)$ for some $2 \le p < \infty$, we have

$$\|\widetilde{R}_j f\|_{L^p(\mathbb{Y}^d)} \le c\|f\|_{L^p(\mathbb{Y}^d)}.$$

$\Diamond$

Proof. Let $g \in L^2(\mathbb{R}^d) \cap L^p(\mathbb{R}^d)$ for some $1 < p < \infty$. For any $j \in \{1,\dots,d\}$, define the family of operators R_j by

$$(R_j g)(x) = \lim_{\delta^{-1},\epsilon\to 0^+} \int_{\delta\ge|t|\ge\epsilon} g(x-t)K_j(t)\,dt,$$

where

$$K_j(t) = \frac{\Gamma\big((d+1)/2\big)t_j}{\pi^{(d+1)/2}|t|^{d+1}}$$

and Γ denotes the Euler-Gamma function. By [184, Chapter 4, Theorem 4.5], $K_j \in \mathcal{S}'(\mathbb{R}^d)$ and its Fourier transform satisfies

$$\widehat{K}_j(t) = -\frac{it_j}{|t|}.$$

Therefore, for any $\phi \in \mathcal{S}(\mathbb{R}^d)$, we have

$$(K_j * \phi)\widehat{\ }(t) = -\frac{it_j}{|t|}\widehat{\phi}(t),$$

where $*$ denotes convolution, see, for example, [184, Chapter 1, Theorem 3.18]. Since $g \in L^2(\mathbb{R}^d)$, we therefore find that, by [184, Chapter 6, Theorem 2.6],

$$(R_j g)\widehat{\ }(x) = -\frac{ix_j}{|x|}\widehat{g}(x) \tag{B.1.14}$$

for Lebesgue almost every $x \in \mathbb{R}^d$. The result [184, Chapter 6, Theorem 2.6] further shows that there exists a constant $c = c(d,p) > 0$ such that

$$\|R_j g\|_{L^p(\mathbb{R}^d)} \leq c\|g\|_{L^p(\mathbb{R}^d)}.$$

We note from (B.1.14) and the definition (B.1.13) that $\widetilde{R}_j$ may be viewed as R_j with the restriction of the Fourier multiplier $-\frac{ix_j}{|x|}$ to the lattice $\mathbb{Z}^d$. We can therefore use the transference theory of [184] to establish boundedness of $\widetilde{R}_j$ from the boundedness of R_j. In particular, note that the mapping $x \mapsto -\frac{ix_j}{|x|}$ is continuous at all $x \in \mathbb{R}^d$ except $x = 0$. However, by symmetry, we have that, for all $\epsilon > 0$

$$\int_{|x|\leq\epsilon} -\frac{ix_j}{|x|}\, dx = 0.$$

Therefore we can apply [184, Chapter 7, Theorem 3.8, Corollary 3.16] to conclude that, since R_j is bounded from $L^p(\mathbb{R}^d)$ to $L^p(\mathbb{R}^d)$, $\widetilde{R}_j$ is bounded from $L^p(\mathbb{Y}^d)$ to $L^p(\mathbb{Y}^d)$ with

$$\|\widetilde{R}_j\|_{L^p(\mathbb{Y}^d)\to L^p(\mathbb{Y}^d)} \leq \|R_j\|_{L^p(\mathbb{R}^d)\to L^p(\mathbb{R}^d)}.$$

This implies the desired result. □

We define the Bessel potential spaces by

$$L^{s,p}(\mathbb{Y}^d) = \{u \in \mathcal{D}'(\mathbb{Y}^d) \mid \|u\|_{L^{s,p}(\mathbb{Y}^d)} := \|(I-\Delta)^{s/2}u\|_{L^p(\mathbb{Y}^d)} < \infty\}$$

for any $-\infty < s < \infty$ and $1 < p < \infty$. We also define the homogeneous version of these spaces, sometimes called the Riesz potential spaces, by

$$\dot{L}^{s,p}(\mathbb{Y}^d) = \{u \in \mathcal{D}'(\mathbb{Y}^d) \mid \|u\|_{\dot{L}^{s,p}(\mathbb{Y}^d)} \coloneqq \|(-\Delta)^{s/2}u\|_{L^p(\mathbb{Y}^d)} < \infty, \int_{\mathbb{Y}^d} u \, \mathsf{d}y = 0\}.$$

It is clear that $\dot{L}^{s,p}(\mathbb{Y}^d) \subset L^{s,p}(\mathbb{Y}^d)$ is closed subspace. We then have the following result for the Poisson equation.

***Lemma* B.1.10.** For each $f \in L^{s,p}(\mathbb{Y}^d)$, for $-\infty < s < \infty$ and $2 \leq p < \infty$, the solution u of the equation

$$-\Delta u = f, \quad u \text{ 1-periodic}, \int_{\mathbb{Y}^d} u \, \mathsf{d}y = 0 \tag{B.1.15}$$

satisfies

$$\|\nabla u\|_{\dot{L}^{s+1,p}(\mathbb{Y}^d)} \leq K\|f\|_{\dot{L}^{s,p}(\mathbb{Y}^d)} \tag{B.1.16}$$

for some finite $K > 0$ depending only on p and d. ◊

Proof. From the definitions (B.1.12) and (B.1.13), it is easy to see that the Riesz transform can be written as

$$\widetilde{R}_j = -\partial_{x_j}(-\Delta)^{-1/2}$$

in the sense of distributions. Consider now equation (B.1.15) with $f \in L^{s,p}(\mathbb{Y}^d)$ for $2 \leq p < \infty$. We have that

$$\begin{aligned}\|\partial_{x_j}u\|_{\dot{L}^{s+1,p}(\mathbb{Y}^d)} &= \|\partial_{x_j}(-\Delta)^{-1}f\|_{\dot{L}^{s+1,p}(\mathbb{Y}^d)} \\ &= \|\partial_{x_j}(-\Delta)^{-1/2}(-\Delta)^{s/2}f\|_{L^p(\mathbb{Y}^d)} \\ &= \|\widetilde{R}_j(-\Delta)^{s/2}f\|_{L^p(\mathbb{Y}^d)}.\end{aligned}$$

It is clear that

$$\|(-\Delta)^{s/2}f\|_{L^p(\mathbb{Y}^d)} = \|f\|_{\dot{L}^{s,p}(\mathbb{Y}^d)} < \infty,$$

and hence $(-\Delta)^{s/2}f \in L^p(\mathbb{Y}^d)$. We can thus apply Lemma B.1.9 to find a constant $c = c(d,p) > 0$ such that

$$\|\partial_{x_j}u\|_{\dot{L}^{s+1,p}(\mathbb{Y}^d)} \leq c\|(-\Delta)^{s/2}f\|_{L^p(\mathbb{Y}^d)} = c\|f\|_{\dot{L}^{s,p}(\mathbb{Y}^d)}.$$

The result follows by finite-dimensional norm equivalence.

□

Next we define the homogeneous Sobolev spaces on the torus as

$$\dot{W}^{k,p}(\mathbb{T}^d) = \{u \in W^{k,p}(\mathbb{T}^d) \mid u \text{ is 1-periodic}, \int_{\mathbb{T}^d} u \, \mathsf{d}y = 0\} \tag{B.1.17}$$

for $k = 0, 1, \ldots,$ and $1 \leq p \leq \infty$ with the standard norm on $W^{k,p}$, see, for example [185].

***Remark* B.1.11.** By [183, Section 3.5.4], we have that, for any $k = 0, 1, \ldots$ and $1 < p < \infty$,

$$L^{k,p}(\mathbb{T}^d) = W^{k,p}(\mathbb{T}^d), \qquad \dot{L}^{k,p}(\mathbb{T}^d) = \dot{W}^{k,p}(\mathbb{T}^d).$$

Furthermore, by [183, Section 3.5.6],

$$\begin{aligned} W^{-k,p'}(\mathbb{T}^d) &= \big(W^{k,p}(\mathbb{T}^d)\big)' = \big(L^{k,p}(\mathbb{T}^d)\big)' = L^{-k,p'}(\mathbb{T}^d), \\ \dot{W}^{-k,p'}(\mathbb{T}^d) &= \big(\dot{W}^{k,p}(\mathbb{T}^d)\big)' = \big(\dot{L}^{k,p}(\mathbb{T}^d)\big)' = \dot{L}^{-k,p'}(\mathbb{T}^d), \end{aligned}$$

where p' is the Hölder conjugate of p i.e. $1/p + 1/p' = 1$. ◊

In the following, we use the notation

$$[K_0, K_1]_{\theta,q} \tag{B.1.18}$$

to denote the real interpolation between two Banach spaces continuously embedded in the same Hausdorff topological space, as described in [185]. We also need Lemma A1 from [186], which we have copied below as Lemma B.1.12 to ease readability. Although this lemma was written only for $q = 2$, the result still holds for our $q > 2$ with a very similar proof.

***Lemma* B.1.12.** Let $E_1 \subset E_0$ be two Banach spaces with E_1 continuously embedded in E_0. Let $T : E_j \to E_j$ be a bounded operator with closed range and assume that T is a projection, $j \in \{0, 1\}$. Denote by K_0 and K_1 the ranges of $T|_{E_0}$ and $T|_{E_1}$ respectively. Then the following two spaces coincide with equivalent norms:

$$[K_0, K_1]_{\theta,q} = [E_0, E_1]_{\theta,q} \cap K_0 \quad \forall \theta \in (0, 1).$$

◊

We now state the result for the bound on $\|\nabla\chi\|_{L^p}$ with a proof largely developed in [62].

***Lemma* B.1.13.** Let χ solve (3.1.4) for $A \in \mathsf{PD}_{\alpha,\beta}$. Then

$$\|\nabla\chi\|_{L^p} \leq \frac{K^{\eta(p)}}{1 - K^{\eta(p)}\left(1 - \frac{\alpha}{\beta}\right)} \tag{B.1.19}$$

for $2 \leq p < p^*\left(\frac{\alpha}{\beta}\right)$ where

$$p^*(t) := \max\left\{p \mid K^{-\eta(p)} \geq 1 - t,\ 2 < p < Q\right\} \tag{B.1.20}$$

for $\eta(p) = \frac{1/2-1/p}{1/2-1/Q}$and $K = K(d, Q)$ is the constant in Lemma B.1.10, for any choice of $Q > p$. ◊

Proof. The operator $T = -\Delta$ is invertible from H^{-1} to $\dot{H}^1$, and the inverse T^{-1} is bounded with norm 1 since the Poisson equation with periodic boundary conditions has a unique solution in $\dot{H}^1$ for $f \in H^{-1}$ with bound $\|u\|_{\dot{H}^1} \leq \|f\|_{H^{-1}}$. From Lemma B.1.10 it is also bounded with norm $K = K(d, Q)$ from $W^{-1,Q}$ to $\dot{W}^{1,Q}$ for any $Q > 2$. By the real method of interpolation [185], for $2 < p < Q$ we have that

$$W^{1,p} = \left[H^1, W^{1,Q}\right]_{\eta(p),p} \tag{B.1.21}$$

using the notation of [185] where $\eta(p) = \frac{1/2-1/p}{1/2-1/Q}$. From the duality theorem (Theorem 3.7.1. of [187]), we have that

$$\left[H^{-1}, W^{-1,Q}\right]_{\eta(p),p} = \left(\left[H^1, W^{1,Q'}\right]_{\eta(p),p'}\right)'. \tag{B.1.22}$$

From real interpolation, the right hand side equals $(W^{1,p'})' = W^{-1,p}$ in our notation. Therefore, we have the necessary dual statement that parallels (B.1.21):

$$W^{-1,p} = \left[H^{-1}, W^{-1,Q}\right]_{\eta(p),p}. \tag{B.1.23}$$

Next we restrict these spaces to functions with periodic boundary conditions. Using the projection onto the space of continuous, periodic functions on $\mathbb{T}^d$ and noticing that $W^{1,Q} \hookrightarrow H^1$, we apply Lemma B.1.12 with $K_0 = \dot{H}^1$ and have

$$\dot{W}^{1,p} = [\dot{H}^1, \dot{W}^{1,Q}]_{\eta(p),p}. \tag{B.1.24}$$

Using the exact interpolation theorem, Theorem 7.23 of [185], T^{-1} is also a bounded map from $W^{-1,p}$ to $\dot{W}^{1,p}$ with norm $K^{\eta(p)}$:

$$\|T^{-1}f\|_{\dot{W}^{1,p}} \leq K^{\eta(p)}\|f\|_{W^{-1,p}}. \tag{B.1.25}$$

The remainder of the proof is identical to that of the proof of Proposition 1 in [62], but we state it here in our notation for completeness. Define S: $\dot{W}^{1,p} \to W^{-1,p}$ as the operator $Su = -\nabla \cdot \left(\frac{1}{\beta} A \nabla u\right)$. Let V be the perturbation operator $V := T - S$. Since $A \in \mathsf{PD}_{\alpha,\beta}$, S and V are bounded operators from $\dot{W}^{1,p}$ to $W^{-1,p}$, with the operator norms $\|S\| \leq 1$ and $\|V\| \leq 1 - \frac{\alpha}{\beta}$. Therefore,

$$\|T^{-1}V\|_{\dot{W}^{1,p} \to \dot{W}^{1,p}} \leq \|T^{-1}\|_{W^{-1,p} \to \dot{W}^{1,p}} \|V\|_{\dot{W}^{1,p} \to W^{-1,p}} \leq K^{\eta(p)} \left(1 - \frac{\alpha}{\beta}\right), \tag{B.1.26}$$

where the input and output spaces defining the operator norms are included for clarity. Since T is invertible, $S = T(I - T^{-1}V)$ is invertible provided $K^{\eta(p)}\left(1 - \frac{\alpha}{\beta}\right) < 1$. Moreover, for S^{-1} as a mapping from $W^{-1,p}$ to $\dot{W}^{1,p}$,

$$\|S^{-1}\| \leq \|(I - T^{-1}V)^{-1}\|_{\dot{W}^{1,p} \to \dot{W}^{1,p}} \|T^{-1}\|_{W^{-1,p} \to \dot{W}^{1,p}} \leq \frac{K^{\eta(p)}}{1 - K^{\eta(p)}\left(1 - \frac{\alpha}{\beta}\right)}. \tag{B.1.27}$$

Therefore,

$$\|\nabla \chi\|_{L^p} = \|\chi\|_{\dot{W}^{1,p}} \leq \frac{1}{\beta} \|S^{-1}\| \|\nabla \cdot A\| \leq \frac{K^{\eta(p)}}{1 - K^{\eta(p)}\left(1 - \frac{\alpha}{\beta}\right)} \tag{B.1.28}$$

provided $K^{\eta(p)}\left(1 - \frac{\alpha}{\beta}\right) < 1$. The bound and specified range of p follow. □

Finally, we may prove Proposition 3.1.3

***Proposition* 3.1.3.** There exists $q_0 \in (2, \infty)$ such that, for all q satisfying $q \in (q_0, \infty]$, the following holds. Endow $\mathsf{PD}_{\alpha,\beta}$ with the $L^q(\mathbb{T}^d; \mathbb{R}^{d \times d})$ topology and let $K \subset \mathsf{PD}_{\alpha,\beta}$ be a closed set. Define the mapping $G : K \to \dot{H}^1(\mathbb{T}^d; \mathbb{R}^d)$ by $A \mapsto \chi$ as given by (3.1.4). Then there exists a bounded Lipschitz-continuous mapping

$$\mathcal{G} : \ L^q(\mathbb{T}^d; \mathbb{R}^{d \times d}) \to \dot{H}^1(\mathbb{T}^d; \mathbb{R}^d)$$

such that $\mathcal{G}(A) = G(A)$ for any $A \in K$. ◊

Proof. Lemma B.1.13 guarantees a $p_0 > 2$ such that $\|\nabla \chi^{(2)}\|_{L^p}$ in Lemma B.1.7 is bounded above by a constant for $2 < p < p_0$. Then Lemma B.1.7 gives Lipschitz continuity of the solution map from $L^q(\mathbb{T}^d) \mapsto \dot{H}^1(\mathbb{T}^d)$ for q satisfying $q_0 < q < \infty$ for some $q_0 > 2$. □

***Remark* B.1.14.** From the results of Lemma B.1.13 and Lemma B.1.7, we have that we can take $q_0 = \frac{2p_0}{p_0-2}$ where

$$p_0 = \max\{p \mid K^{-\eta(p)} \geq 1 - t,\ 2 < p < Q\}.$$

Therefore, bounds on p_0 may be inherited from bounds on K that appears in Lemma B.1.10. ◊

B.2 Proofs of Approximation Theorems

In this section we prove the approximation theorems stated in Section 3.3.

***Theorem* 3.3.2.** Let $K \subset \mathsf{PD}_{\alpha,\beta}$ and define the mapping $G : K \to \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)$ by $A \mapsto \chi$ as given by (3.1.4). Assume in addition that K is compact in $L^2(\mathbb{T}^d;\mathbb{R}^{d\times d})$. Then, for any $\epsilon > 0$, there exists an FNO $\Psi : K \to \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)$ such that

$$\sup_{A\in K} \|G(A) - \Psi(A)\|_{\dot{H}^1} < \epsilon.$$

◊

Proof. By Proposition 3.1.2, there exists a continuous map

$\mathcal{G} \in C(L^2(\mathbb{T}^d;\mathbb{R}^{d\times d}); \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d))$ such that $\mathcal{G}(A) = G(A)$ for any $A \in K$. By [20, Theorem 5], there exists a FNO $\Psi : L^2(\mathbb{T}^d;\mathbb{R}^{d\times d}) \to \dot{H}^1(\mathbb{T}^d;\mathbb{R}^d)$ such that

$$\sup_{A\in K} \|\mathcal{G}(A) - \Psi(A)\|_{\dot{H}^1} < \epsilon.$$

Therefore

$$\sup_{A\in K} \|G(A) - \Psi(A)\|_{\dot{H}^1} = \sup_{A\in K} \|\mathcal{G}(A) - \Psi(A)\|_{\dot{H}^1} < \epsilon$$

as desired. □

***Theorem* 3.3.3.** Let $K \subset \mathsf{PD}_{\alpha,\beta}$ and define the mapping $F : K \to \mathbb{R}^{d\times d}$ by $A \mapsto \overline{A}$ as given by (3.1.3), (3.1.4). Assume in addition that K is compact in $L^2(\mathbb{T}^d;\mathbb{R}^{d\times d})$. Then, for any $\epsilon > 0$, there exists an FNO $\Phi : K \to L^\infty(\mathbb{T}^d;\mathbb{R}^{d\times d})$ such that

$$\sup_{A\in K} \sup_{x\in\mathbb{T}^d} |F(A) - \Phi(A)(x)|_F < \epsilon.$$

◊

Proof. The result follows as in Theorem 3.3.2 by applying Lemma B.1.6 instead of Proposition 3.1.2. □

B.3 Proofs for Microstructure Examples

The following lemma establishes the compactness of subsets of $\mathsf{PD}_{\alpha,\beta}$ generated by the probability measures from Section 3.4. As we are unaware of a proof in the literature, we have provided one below. The proof uses the L^1-Lipschitz spaces, which are defined as

$$\mathrm{Lip}_\alpha(L^1) = \{u \in L^1 : \exists M(u) > 0 : \omega(u,t)_1 \leq Mt^\alpha\},$$

where $\omega(u,t)_1$ is the 1-modulus of continuity, defined via

$$\omega(u,t)_1 = \sup_{0\leq|h|\leq t} \|\tau_h u - u\|_{L^1(\mathbb{T}^d)}.$$

***Lemma* B.3.1.** $\mathrm{BV}(\mathbb{T}^d) \cap L^\infty(\mathbb{T}^d)$ is compactly embedded in $L^2(\mathbb{T}^d)$. ◊

Proof. Let $u \in B$, where B is a bounded subset of $\mathrm{BV}(\mathbb{T}^d) \cap L^\infty(\mathbb{T}^d)$ with L^∞ norm and BV seminorm bounded by M, and let $\tau_h f$ denote the translation of f by h, i.e. $\tau_h f(x) = f(x-h)$. Then

$$\|\tau_h u - u\|_{L^2} \leq \|\tau_h u - u\|_{L^1}^{1/2} \|\tau_h u - u\|_{L^\infty}^{1/2}. \tag{B.3.1}$$

Since $\mathrm{BV}(\mathbb{T}^d) \equiv \mathrm{Lip}_1(L^1(\mathbb{T}^d))$, $\|\tau_h u - u\|_{L^1} \leq \|u\|_{\mathrm{BV}}|h|$. We have then

$$\|\tau_h u - u\|_{L^2} \leq \|u\|_{\mathrm{BV}}^{1/2} |h|^{1/2} (2M)^{1/2}.$$

By the Fréchet-Kolmogorov theorem [188], this equicontinuity result is sufficient for compactness of B in $L^2(\mathbb{T}^d)$. □

Using the result of Lemma B.3.1, we see that any set of microstructure coefficients bounded in $L^\infty(\mathbb{T}^d) \cap BV(\mathbb{T}^d)$ satisfies the compactness assumption of the Approximation Theorems in Section 3.3. It is clear that the method of construction of the microstructure examples used in the main body of the work leads to such sets.

B.4 Numerical Implementation Details

All FNO models are implemented in pytorch using python 3.9.7. Unless otherwise specified, the models have 18 modes in each dimension, a width of 64, and 4 hidden layers. The lifting layer is a linear transformation with trainable parameters, and the projecting layer is a pointwise multilayer perceptron with

trainable parameters. The batch size is 20, the learning rate is 0.001, and the number of epochs is 400. These hyperparameters are chosen with a small grid search, but we emphasize that the FNO does not drastically change in performance unless these parameters are changed by an order of magnitude. For a model trained on 9500 data using these hyperparameters and accelerated with an Nvidia P100 GPU, the training time is approximately 7 hours. In Figures 3.4, 3.5, and 3.6, the error bars shown correspond to two standard deviations in each direction over the five samples.

Appendix C

APPENDIX TO CHAPTER 4

Links to datasets and all code used to produce the numerical results and figures in this chapter are available at

https://github.com/mtrautner/BoundFNO/ .

C.1 Trigonometric interpolation and aliasing

In this section, we present a self-contained analysis of aliasing errors for $v \in H^s(\mathbb{T}^d)$. These results are straightforward and well known in numerical analysis, but we give a clear exposition here as background as it is difficult to find a succinct and widely-available reference. The primary goal is to state and prove Proposition C.1.6, which controls the difference between a function defined over $\mathbb{T}^d$ and the trigonometric interpolation of a function defined on a grid. In the following, $N \in \mathbb{Z}_{>0}$. We recall that $X^{(N)}$ is a set of equidistant grid points on the torus $\mathbb{T}^d$,

$$X^{(N)} = \{x_n \in \mathbb{T}^d \,|\, x = n/N,\ n \in [N]^d\}.$$

We note that the discrete Fourier transform gives rise to a natural correspondence between grid values and Fourier modes,

$$\{v(x_n)\}_{n\in[N]^d} \ \leftrightarrow\ \{\widehat{v}_k\}_{k\in[[N]]^d}, \tag{C.1.1}$$

where

$$\widehat{v}_k = \frac{1}{N^d}\sum_{n\in[N]^d} v(x_n)e^{-2\pi i\langle k,x_n\rangle} =: \mathsf{DFT}(v)(k). \tag{C.1.2}$$

We begin with the following observation:

***Lemma* C.1.1.** Let N be given. Then,

$$\frac{1}{N^d}\sum_{k\in[[N]]^d} e^{2\pi i\langle k,x_m-x_n\rangle} = \delta_{mn}, \quad \forall\, m,n \in [N]^d, \tag{C.1.3}$$

$$\frac{1}{N^d}\sum_{n\in[N]^d} e^{2\pi i\langle k-k',x_n\rangle} = \delta_{kk'}, \quad \forall\, k,k' \in [[N]]^d. \tag{C.1.4}$$

$\Diamond$

Proof. This follows from an elementary calculation, which we briefly recall here. For $d = 1$, the claim follows by noting that $x_n = n/N$, and using the identity

$$\sum_{\ell=0}^{N-1} q^\ell = \begin{cases} \frac{q^N-1}{q-1}, & (q \neq 1), \\ N, & (q = 1), \end{cases} \tag{C.1.5}$$

with $q = e^{2\pi i(m-n)/N}$ and $q = e^{2\pi i(k-k')/N}$, respectively. Indeed, assuming $d = 1$ and denoting $-K := \min[[N]]$, then the above identity implies, for example,

$$\sum_{k\in[[N]]} e^{2\pi ik(x_m-x_n)} = \sum_{k\in[[N]]} \big[\underbrace{e^{2\pi i(m-n)/N}}_{=:q}\big]^k = \sum_{k\in[[N]]} q^k = q^{-K}\sum_{\ell=0}^{N-1} q^\ell.$$

If $q \neq 1$, then $q^N = e^{2\pi i(m-n)} = 1$. By (C.1.5), this implies that the last sum is 0. On the other hand, if $q = 1$, then the last sum is trivially $= N$. We finally note that, for $m, n \in [N]$, we have $q = 1$ if and only if $m = n$, implying that

$$q^{-K}\sum_{\ell=0}^{N} q^\ell = N\delta_{mn}.$$

Thus,

$$\sum_{k\in[[N]]} e^{2\pi ik(x_m-x_n)} = N\delta_{mn},$$

and (C.1.3) follows. The argument for (C.1.4) is analogous. For $d > 1$, the sum over $[[N]]^d = [[N]] \times \cdots \times [[N]]$ is split into sums along each dimension, and the same argument is applied for each of the d components, yielding the claim also for $d > 1$. □

A *trigonometric polynomial* $p : \mathbb{T}^d \mapsto \mathbb{R}^m$ is a function of the form

$$p(x) = \sum_{k\in[[N]]^d} c_k e^{2\pi i\langle k,x\rangle} \tag{C.1.6}$$

with $c_k \in \mathbb{C}^m$ chosen to make $p(x)$ $\mathbb{R}^m$-valued at each $x \in \mathbb{T}^d$. We note that the discrete and continuous L^2-norms are equivalent for trigonometric polynomials:

***Lemma* C.1.2.** Let N be a positive integer. If $p(x)$ is a trigonometric polynomial, then

$$\frac{1}{N^{d/2}}\|p\|_{\ell^2(n\in[N]^d)} = \|p\|_{L^2(\mathbb{T}^d)}.$$

◊

Proof. We have

$$\|p\|_{L^2(\mathbb{T}^d)}^2 = \int_{\mathbb{T}^d} |p(x)|^2\,dx = \sum_{k,k'\in[[N]]^d} c_k\bar{c}_{k'} \underbrace{\int_{\mathbb{T}^d} e^{2\pi i\langle k-k',x\rangle}\,dx}_{=\delta_{kk'}} = \sum_{k\in[[N]]^d} |c_k|^2,$$

and

$$\begin{aligned}\frac{1}{N^d}\|p\|_{\ell^2(n\in[N]^d)}^2 &= \frac{1}{N^d}\sum_{n\in[N]^d} |p(x_n)|^2 \\ &= \sum_{k,k'\in[[N]]^d} c_k\bar{c}_{k'} \underbrace{\frac{1}{N^d}\sum_{n\in[N]^d} e^{2\pi i\langle k-k',x_n\rangle}}_{=\delta_{kk'}} \\ &= \sum_{k\in[[N]]^d} |c_k|^2.\end{aligned}$$

This proves the claim. □

Let $v : \mathbb{T}^d \to \mathbb{R}$ be a function with grid values $\{v(x_n)\}_{n\in[N]^d}$. Let $\mathsf{DFT}(v)(k)$ denote the coefficients of the discrete Fourier transform defined by (C.1.1). Then

$$p(x) := \sum_{k\in[[N]]^d} \mathsf{DFT}(v)(k)e^{2\pi i\langle k,x\rangle}, \tag{C.1.7}$$

is the trigonometric polynomial associated to v. The next lemma shows that $p(x)$ interpolates $v(x)$.

***Lemma* C.1.3.** The trigonometric polynomial $p(x)$ defined by (C.1.7) interpolates $v(x)$ at the grid points, i.e., we have $p(x_n) = v(x_n)$ for all $n \in [N]^d$. ◊

Proof. Fix $n \in [N]^d$. Then

$$\begin{aligned}p(x_n) &= \sum_{k\in[[N]]^d} \mathsf{DFT}(v)(k)e^{2\pi i\langle k,x_n\rangle} \\ &= \sum_{k\in[[N]]^d} \left\{\frac{1}{N^d}\sum_{m\in[N]^d} v(x_m)e^{-2\pi i\langle k,x_m\rangle}\right\} e^{2\pi i\langle k,x_n\rangle} \\ &= \sum_{m\in[N]^d} v(x_m)\left\{\frac{1}{N^d}\sum_{k\in[[N]]^d} e^{2\pi i\langle k,x_n-x_m\rangle}\right\} \\ &= \sum_{m\in[N]^d} v(x_m)\delta_{mn} \\ &= v(x_n),\end{aligned}$$

where we have made use of (C.1.3) to pass to the fourth line. □

The following trigonometric polynomial interpolation estimate for functions in Sobolev spaces $H^s(\mathbb{T}^d)$ will be useful in stating our main proposition.

***Lemma* C.1.4.** Let $v \in H^s(\mathbb{T}^d)$ for $s > d/2$. Let p denote the interpolating trigonometric polynomial given by (C.1.7). Then

$$v(x) - p(x) = \sum_{k \in \mathbb{Z}^d \setminus [[N]]^d} \widehat{v}(k) e^{2\pi i \langle k, x \rangle} - \sum_{k \in [[N]]^d} \left\{ \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} \widehat{v}(k + \ell N) \right\} e^{2\pi i \langle k, x \rangle}. \tag{C.1.8}$$

Furthermore, there exists a constant $c_{s,d} > 0$, such that

$$\|v - p\|_{L^2(\mathbb{T}^d)} \leq c_{s,d} \|v\|_{H^s(\mathbb{T}^d)} N^{-s}. \tag{C.1.9}$$

◊

***Remark* C.1.5.** The first sum on the right-hand side of (C.1.8) is the L^2-orthogonal Fourier projection of v onto the complement of $\operatorname{span}\{e^{2\pi i \langle k, x \rangle} \mid k \in [[N]]^d\}$. The second sum in (C.1.8) is known as an "aliasing" error; it arises because two Fourier modes are indistinguishable on the discrete grid whenever $k - k' \in N\mathbb{Z}^d$, i.e. $e^{2\pi i \langle k, x_n \rangle} = e^{2\pi i \langle k', x_n \rangle}$ for all $n \in [N]^d$. ◊

Proof. Since $v \in H^s(\mathbb{T}^d)$ has Sobolev smoothness s for $s > d/2$, it can be shown that the Fourier series of v is uniformly convergent. In particular, we may write

$$v(x) = \sum_{k' \in \mathbb{Z}^d} \widehat{v}(k') e^{2\pi i \langle k', x \rangle}$$

for

$$\widehat{v}(k') = \int_{\mathbb{T}^d} v(x) e^{-2\pi i \langle k', x \rangle} \, \mathsf{d}x.$$

First, substitution of $v(x_n) = \sum_{k' \in \mathbb{Z}^d} \widehat{v}(k') e^{2\pi i \langle k', x_n \rangle}$ into $\mathsf{DFT}(v)(k)$ yields

$$\begin{aligned} \mathsf{DFT}(v)(k) &= \frac{1}{N^d} \sum_{n \in [N]^d} \left\{ \sum_{k' \in \mathbb{Z}^d} \widehat{v}(k') e^{2\pi i \langle k', x_n \rangle} \right\} e^{-2\pi i \langle k, x_n \rangle} \\ &= \sum_{k' \in \mathbb{Z}^d} \widehat{v}(k') \left\{ \frac{1}{N^d} \sum_{n \in [N]^d} e^{2\pi i \langle k' - k, x_n \rangle} \right\}. \end{aligned}$$

We now note that

$$\frac{1}{N^d}\sum_{n\in[N]^d} e^{2\pi i\langle k'-k,x_n\rangle} = \begin{cases} 0, & (k' \not\equiv k \mod N), \\ 1, & (k' \equiv k \mod N), \end{cases}$$

as a consequence of the trigonometric identity (C.1.4). Letting $k' = k + \ell N$, i.e. k' for which the sum inside the braces does not vanish, it follows that

$$\mathsf{DFT}(v)(k) = \sum_{\ell\in\mathbb{Z}^d} \widehat{v}(k+\ell N).$$

Thus,

$$\begin{aligned} v(x) - p(x) &= \sum_{k\in\mathbb{Z}^d} \widehat{v}(k)e^{2\pi i\langle k,x\rangle} - \sum_{k\in[[N]]^d} \mathsf{DFT}(v)(k)e^{2\pi i\langle k,x\rangle} \\ &= \sum_{k\in\mathbb{Z}^d\setminus[[N]]^d} \widehat{v}(k)e^{2\pi i\langle k,x\rangle} + \sum_{k\in[[N]]^d} \{\widehat{v}(k) - \mathsf{DFT}(v)(k)\}e^{2\pi i\langle k,x\rangle} \\ &= \sum_{k\in\mathbb{Z}^d\setminus[[N]]^d} \widehat{v}(k)e^{2\pi i\langle k,x\rangle} - \sum_{k\in[[N]]^d} \left\{ \sum_{\ell\in\mathbb{Z}^d\setminus\{0\}} \widehat{v}(k+\ell N) \right\} e^{2\pi i\langle k,x\rangle}. \end{aligned}$$

We proceed to bound the last two terms. For the first term, we have by Parseval's theorem,

$$\begin{aligned} \Big\| \sum_{k\in\mathbb{Z}^d\setminus[[N]]^d} \widehat{v}(k)e^{2\pi i\langle k,x\rangle} \Big\|^2_{L^2(\mathbb{T}^d)} &= \sum_{k\in\mathbb{Z}^d\setminus[[N]]^d} |\widehat{v}(k)|^2 \\ &\le \frac{1}{(1+(N/2)^{2s})} \sum_{k\in\mathbb{Z}^d} (1+|k|^{2s})|\widehat{v}(k)|^2 \\ &\le 4^s N^{-2s} \|v\|^2_{H^s(\mathbb{T}^d)}, \end{aligned}$$

where $\|v\|^2_{H^s(\mathbb{T}^d)} = \sum_{k\in\mathbb{Z}^d}(1+|k|^{2s})|\widehat{v}(k)|^2$, and for the second term

$$\begin{aligned} &\Big\| \sum_{k\in[[N]]^d} \Big\{ \sum_{\ell\in\mathbb{Z}^d\setminus\{0\}} \widehat{v}(k+\ell N) \Big\} e^{2\pi i\langle k,x\rangle} \Big\|^2_{L^2(\mathbb{T}^d)} \\ &\qquad = \sum_{k\in[[N]]^d} \Big| \sum_{\ell\in\mathbb{Z}^d\setminus\{0\}} \widehat{v}(k+\ell N) \Big|^2 \\ &\qquad \le \sum_{k\in[[N]]^d} \Big(\sum_{\ell\in\mathbb{Z}^d\setminus\{0\}} (1+|k+\ell N|^{2s})^{-1} \Big) \\ &\qquad\qquad \times \Big(\sum_{\ell\in\mathbb{Z}^d\setminus\{0\}} (1+|k+\ell N|^{2s})|\widehat{v}(k+\ell N)|^2 \Big). \end{aligned}$$

The final inequality is obtained via Cauchy-Schwarz. We note that for $k \in [[N]]^d$, we have $|k|_\infty \leq N/2$, and hence, for any integer vector $\ell \neq 0$, we obtain

$$|k + \ell N| \geq |k + \ell N|_\infty \geq |\ell|_\infty N - |k|_\infty \geq |\ell|_\infty N - \frac{N}{2} \geq \frac{N}{2}|\ell|_\infty \geq \frac{N}{2\sqrt{d}}|\ell|. \tag{C.1.10}$$

We can now bound

$$\sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} (1 + |k + \ell N|^{2s})^{-1} \leq \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} \left(\frac{N}{2\sqrt{d}}\right)^{-2s} |\ell|^{-2s} \tag{C.1.11a}$$

$$\leq c_{d,s} N^{-2s}, \tag{C.1.11b}$$

where $c_{d,s} := (4d)^s \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} |\ell|^{-2s} < \infty$ is finite, since $s > d/2$ implies that the last series converges. Substitution of this bound in the estimate above implies

$$\begin{aligned}
&\left\| \sum_{k \in [[N]]^d} \Big\{ \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} \widehat{v}(k + \ell N) \Big\} e^{2\pi i \langle k, x \rangle} \right\|^2_{L^2(\mathbb{T}^d)} \\
&\qquad \leq c_{d,s} N^{-2s} \sum_{k \in [[N]]^d} \left(\sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} (1 + |k + \ell N|^{2s}) |\widehat{v}(k + \ell N)|^2 \right) \\
&\qquad \leq c_{d,s} N^{-2s} \|v\|^2_{H^s(\mathbb{T}^d)}.
\end{aligned}$$

Combining the above estimates, we conclude that

$$\|v - p\|_{L^2} \leq c_{d,s} \|v\|_{H^s(\mathbb{T}^d)} N^{-s},$$

where we have re-defined $c_{d,s} := 2^s + (4d)^{s/2} \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} |\ell|^{-2s}$. □

We can now state the main outcome of this section.

***Proposition* C.1.6.** Let $v \in H^s(\mathbb{T}^d)$ be given for $s > d/2$ and let $\{v^N(x_n)\}_{n \in [N]^d}$ be any grid values. Let $p^N(x) = \sum_{k \in [[N]]^d} \mathsf{DFT}(v^N)(k) e^{2\pi i \langle k, x \rangle}$ be the interpolating trigonometric polynomial of v^N. Then,

$$\|v - p^N\|_{L^2(\mathbb{T}^d)} \leq \frac{1}{N^{d/2}} \|v - v^N\|_{\ell^2(n \in [N]^d)} + c_{d,s} \|v\|_{H^s(\mathbb{T}^d)} N^{-s}.$$

◊

Proof. Let $p(x) = \sum_{k \in [[N]]^d} \mathsf{DFT}(v)(k) e^{2\pi i \langle k, x \rangle}$ be the interpolating trigonometric polynomial given the point-values $\{v(x_n)\}_{n \in [N]^d}$. Then,

$$\|v - p^N\|_{L^2(\mathbb{T}^d)} \leq \|v - p\|_{L^2(\mathbb{T}^d)} + \|p - p^N\|_{L^2(\mathbb{T}^d)}. \tag{C.1.12}$$

By Lemma C.1.4, we have

$$\|v - p\|_{L^2(\mathbb{T}^d)} \leq c_{d,s}\|v\|_{H^s(\mathbb{T}^d)}N^{-s}.$$

By Lemma C.1.2, and since $p(x_n) = v(x_n)$, $p^N(x_n) = v^N(x_n)$ by Lemma C.1.3, we have

$$\begin{aligned}\|p - p^N\|_{L^2(\mathbb{T}^d)} &= \frac{1}{N^{d/2}}\|p(x_n) - p^N(x_n)\|_{\ell^2(n\in[N]^d)} \\ &= \frac{1}{N^{d/2}}\|v(x_n) - v^N(x_n)\|_{\ell^2(n\in[N]^d)}.\end{aligned}$$

Substitution in (C.1.12) gives the claimed bound. □

C.2 Discretization error derivation

In this section, we derive the error breakdown within each FNO layer. This error breakdown is used in the proofs of subsequent sections. Within a single layer, we define the following quantities to track the error origin and propagation, noting that, for values of m_t that will vary with layer t, $\mathcal{E}_t^{(j)} : X^{(N)} \to \mathbb{R}^{m_t}$, $j = 0, 3$ and $\mathcal{E}_t^{(j)} : [[K]]^d \to \mathbb{C}^{m_t}$, $j = 1, 2$.

0. $\mathcal{E}_t^{(0)}(x_n) = v_t^N(x_n) - v_t(x_n), \quad x_n \in X^{(N)}$.
1. $\mathcal{E}_t^{(1)}(k) = \frac{1}{N^d}\sum_{n\in[N]^d} v_t(x_n)e^{-2\pi i\langle k,x_n\rangle} - \int_{\mathbb{T}^d} v_t(x)e^{-2\pi i\langle k,x\rangle} \, \mathsf{d}x, \quad k \in [[K]]^d$.
2. $\mathcal{E}_t^{(2)}(k) = \frac{1}{N^d}\sum_{n\in[N]^d} \mathcal{E}_t^{(0)}(x_n)e^{-2\pi i\langle k,x_n\rangle}, \quad k \in [[K]]^d$.
3. $\mathcal{E}_t^{(3)}(x_n) = \sum_{k\in[[K]]^d} P_t^{(k)}\big(\mathcal{E}^{(1)}(k) + \mathcal{E}^{(2)}(k)\big)e^{2\pi i\langle k,x_n\rangle}, \quad x_n \in X^{(N)}$.
4. $\mathcal{E}_{t+1}^{(0)}(x_n) = \sigma\Big(W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t + W_t\mathcal{E}_t^{(0)}(x_n) + \mathcal{E}_t^{(3)}(x_n)\Big)$
   $- \sigma(W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t) = v_{t+1}^N(x_n) - v_{t+1}(x_n), \quad x_n \in X^{(N)}$.

Here, $\mathcal{E}_t^{(0)}$ is the initial error in the inputs to FNO layer t, $\mathcal{E}^{(1)}$ is the aliasing error, $\mathcal{E}_t^{(2)}$ is the initial error $\mathcal{E}_t^{(0)}$ after the discrete Fourier transform, and $\mathcal{E}_t^{(3)}$ is the error after the operation of the kernel $\mathcal{K}_t$. Finally, the initial error for the next layer is given by $\mathcal{E}_{t+1}^{(0)}$ in terms of the error quantities of the previous layer. Intuitively, the quantity $\mathcal{E}^{(1)}$ is the source of the error within each layer since it depends only on the ground truth v_t. All other error quantities are propagation of existing error from previous layers. We provide an exact derivation of these quantities in the following.

Let $\mathcal{E}_t^{(0)}$ be the error in the inputs to FNO layer t such that

$$\mathcal{E}_t^{(0)}(x_n) = v_t^N(x_n) - v_t(x_n), \quad x_n \in X^{(N)}.$$

Let $\mathcal{F}(v_t)(k) = \int_{\mathbb{T}^d} v_t(x) e^{-2\pi i \langle k, x \rangle} \, \mathsf{d}x$ denote the Fourier transform and DFT as in equation (C.1.2). Then for $k \in [[K]]^d$,

$$\begin{aligned}
\mathsf{DFT}(v_t^N)(k) &= \frac{1}{N^d} \sum_{n \in [N]^d} v_t(x_n) e^{-2\pi i \langle k, x_n \rangle} + \frac{1}{N^d} \sum_{n \in [N]^d} \mathcal{E}_t^{(0)}(x_n) e^{-2\pi i \langle k, x_n \rangle} \\
&= \mathcal{F}(v_t)(k) + \mathcal{E}_t^{(1)}(k) + \mathcal{E}_t^{(2)}(k),
\end{aligned}$$

where $\mathcal{E}_t^{(1)}$ is the error resulting from computing the Fourier transform of v_t on a discrete grid rather than all of $\mathbb{T}^d$, i.e.

$$\mathcal{E}_t^{(1)}(k) = \frac{1}{N^d} \sum_{n \in [N]^d} v_t(x_n) e^{-2\pi i \langle k, x_n \rangle} - \int_{\mathbb{T}^d} v_t(x) e^{-2\pi i \langle k, x \rangle} \, \mathsf{d}x$$

and $\mathcal{E}_t^{(2)}$ is the error $\mathcal{E}_t^{(0)}$ after the discrete Fourier transform, i.e.

$$\mathcal{E}_t^{(2)}(k) = \frac{1}{N^d} \sum_{n \in [N]^d} \mathcal{E}_t^{(0)}(x_n) e^{-2\pi i \langle k, x_n \rangle}.$$

For $x_n \in X^{(N)}$, the output of the discrete kernel integral operator acting on v_t^N is given by

$$\begin{aligned}
(\mathcal{K}_t^N v_t^N)(x_n) &= \sum_{k \in [[K]]^d} P_t^{(k)} \Big(\mathcal{F}(v_t)(k) + \mathcal{E}_t^{(1)}(k) + \mathcal{E}_t^{(2)}(k) \Big) e^{2\pi i \langle k, x_n \rangle} \\
&= (\mathcal{K}_t v_t)(x_n) + \mathcal{E}_t^{(3)}(x_n),
\end{aligned}$$

where

$$\mathcal{E}_t^{(3)}(x_n) = \sum_{k \in [[K]]^d} P_t^{(k)} \big(\mathcal{E}^{(1)}(k) + \mathcal{E}^{(2)}(k) \big) e^{2\pi i \langle k, x_n \rangle}.$$

Finally, the output of layer t is given by

$$\begin{aligned}
v_{t+1}^N(x_n) &= \sigma\Big(W_t \big(v_t(x_n) + \mathcal{E}_t^{(0)}(x_n) \big) + (\mathcal{K}_t^N v_t^N)(x_n) + b_t \Big) \\
&= \sigma\Big(W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t + W_t \mathcal{E}_t^{(0)}(x_n) + \mathcal{E}_t^{(3)}(x_n) \Big).
\end{aligned}$$

Therefore, the initial error for the next layer is given by

$$\begin{aligned}
\mathcal{E}_{t+1}^{(0)}(x_n) = \sigma\Big(& W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t + W_t \mathcal{E}_t^{(0)}(x_n) + \mathcal{E}_t^{(3)}(x_n) \Big) \\
& - \sigma(W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t).
\end{aligned}$$

C.3 Proofs of approximation theory lemmas

We bound the components described in Appendix C.2 in the following proposition.

***Proposition* C.3.1.** Under Assumptions 4.3.1, it holds that

1. $\|\mathcal{E}_t^{(1)}\|_{\ell^2(k\in[[K]]^d)} \le \alpha_{d,s} N^{-s} \|v_t\|_{H^s}$ where $\alpha_{d,s}$ is independent of N, v_t;
2. $\|\mathcal{E}_t^{(2)}\|_{\ell^2(k\in[[N]]^d)} = N^{-d/2} \|\mathcal{E}_t^{(0)}\|_{\ell^2(n\in[N]^d)}$;
3. $\|\mathcal{E}_t^{(3)}\|_{\ell^2(n\in[N]^d)} \le N^{d/2} \|P_t\|_F \Big(\|\mathcal{E}_t^{(1)}\|_{\ell^2(k\in[[K]]^d)} + \|\mathcal{E}_t^{(2)}\|_{\ell^2(k\in[[K]]^d)} \Big)$;
4. $\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2(n\in[N]^d)} \le B \Big(\|W_t\|_2 \|\mathcal{E}_t^{(0)}\|_{\ell^2(n\in[N]^d)} + \|\mathcal{E}_t^{(3)}\|_{\ell^2(n\in[N]^d)} \Big)$.

◊

Proof. Beginning with the definition of $\mathcal{E}_t^{(1)}(k)$, we have

$$\|\mathcal{E}_t^{(1)}\|^2_{\ell^2(k\in[[K]]^d)} = \Big\| \frac{1}{N^d} \sum_{n\in[N]^d} v_t(x_n) e^{-2\pi i \langle k, x_n\rangle} - \int_{\mathbb{T}^d} e^{-2\pi i \langle k,x\rangle} v_t(x)\, \mathsf{d}x \Big\|^2_{\ell^2(k\in[[K]]^d)}.$$

Denote the terms in the above expression $\widehat{v}_t^N(k)$ and $\widehat{v}_t(k)$, respectively. Since $s > \frac{d}{2}$,

$$v_t(x_n) = \sum_{k\in\mathbb{Z}^d} \widehat{v}_t(k) e^{2\pi i \langle k, x_n\rangle},$$

and it follows that

$$\begin{aligned}
\widehat{v}_t^N(k') &= \frac{1}{N^d} \sum_{n\in[N]^d} \left(\sum_{k\in\mathbb{Z}^d} \widehat{v}_t(k) e^{2\pi i \langle k, x_n\rangle} \right) e^{-2\pi i \langle k', x_n\rangle} \\
&= \sum_{k\in\mathbb{Z}^d} \widehat{v}_t(k) \frac{1}{N^d} \sum_{n\in[N]^d} e^{2\pi i \langle k-k', x_n\rangle} \\
&= \sum_{\ell\in\mathbb{Z}^d} \widehat{v}_t(k' + \ell N).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\mathcal{E}_t^{(1)}\|^2_{\ell^2(k\in[[K]]^d)} &= \|\widehat{v}_t^N - \widehat{v}_t\|^2_{\ell^2(k\in[[K]]^d)} \\
&= \sum_{k\in[[K]]^d} \left| \sum_{\ell\in\mathbb{Z}^d\setminus\{0\}} \widehat{v}_t(k+\ell N) \right|^2 \\
&\le \sum_{k\in[[K]]^d} \left(\sum_{\ell\in\mathbb{Z}^d\setminus\{0\}} \frac{1}{|k+\ell N|^{2s}} \right) \sum_{\ell\in\mathbb{Z}^d\setminus\{0\}} |k+\ell N|^{2s} |\widehat{v}_t(k+\ell N)|^2
\end{aligned}$$

by Cauchy-Schwarz. We bound each component separately. It is clear from Definition 4.2.1 that

$$\sum_{k\in[[K]]^d}\sum_{\ell\in\mathbb{Z}^d\setminus\{0\}}|k+\ell N|^{2s}|\widehat{v}_t(k+\ell N)|^2\le\|v_t\|_{H^s}^2. \tag{C.3.1}$$

To bound the first component independently of k, we note from $K\le\frac{N}{2}$ and equation (C.1.10) that

$$\begin{aligned}\sum_{\ell\in\mathbb{Z}^d\setminus\{0\}}\frac{1}{|k+\ell N|^{2s}}&\le\sum_{\ell\in\mathbb{Z}^d\setminus\{0\}}\left(\frac{N}{2\sqrt{d}}|\ell|\right)^{-2s}\\&\le\alpha_{d,s}^2N^{-2s}\end{aligned}$$

by equation (C.1.11), where $\alpha_{d,s}^2=(4d)^s\sum_{\ell\in\mathbb{Z}^d\setminus\{0\}}|\ell|^{-2s}$ is finite since $s\ge\frac{d}{2}$. We express the final bound as

$$\|\mathcal{E}_t^{(1)}\|_{k\in[[K]]^d}\le\alpha_{d,s}N^{-s}\|v_t\|_{H^s}.$$

For $\mathcal{E}_t^{(2)}(k)$ we have the definition

$$\mathcal{E}_t^{(2)}(k)=\frac{1}{N^d}\sum_{n\in[N]^d}\mathcal{E}_t^{(0)}(x_n)e^{-2\pi i\langle k,x_n\rangle}.$$

By Parseval's Theorem, we have

$$\|\mathcal{E}_t^{(2)}\|_{\ell^2(k\in[[N]]^d)}^2=\frac{1}{N^d}\|\mathcal{E}_t^{(0)}\|_{\ell^2(n\in[N]^d)}^2. \tag{C.3.2}$$

For $P_t\in\mathbb{R}^{d_{v_{t+1}}\times K^d\times d_{v_t}}$ we define the tensor Frobenius norm $\|P_t\|_F^2=\sum_{k\in[[K]]^d}\|P_t^{(k)}\|_F^2$.

$$\begin{aligned}\|\mathcal{E}_t^{(3)}\|_{\ell^2(n\in[N]^d)}^2&=\sum_{n\in[N]^d}\left|\sum_{k\in[[K]]^d}P_t^{(k)}\Big(\mathcal{E}_t^{(1)}(k)+\mathcal{E}_t^{(2)}(k)\Big)e^{2\pi i\langle k,x_n\rangle}\right|^2\\&\le N^d\left|\sum_{k\in[[K]]^2}|P_t^{(k)}(\mathcal{E}_t^{(1)}(k)+\mathcal{E}_t^{(2)}(k))|\right|^2\\&\le N^d\sum_{k\in[[K]]^d}\|P_t^{(k)}\|_F^2\sum_{k\in[[K]]^d}|\mathcal{E}_t^{(1)}(k)+\mathcal{E}_t^{(2)}(k)|^2\\&=N^d\|P_t\|_F^2\|\mathcal{E}_t^{(1)}+\mathcal{E}_t^{(2)}\|_{\ell^2(k\in[[K]]^d)}^2\\\|\mathcal{E}_t^{(3)}\|_{\ell^2(n\in[N]^d)}&\le N^{d/2}\|P_t\|_F\Big(\|\mathcal{E}_t^{(1)}\|_{\ell^2(k\in[[K]]^d)}+\|\mathcal{E}_t^{(2)}\|_{\ell^2(k\in[[K]]^d)}\Big).\end{aligned}$$

Finally, we have

$$\begin{aligned}\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2(n\in[N]^d)}^2 &= \sum_{n\in[N]^d}\left|\sigma(W_t v_t + \mathcal{K}_t v_t + b_t + W_t\mathcal{E}_t^{(0)}(x_n) + \mathcal{E}_t^{(3)}(x_n)) - \sigma(W_t v_t + \mathcal{K}_t v_t + b_t)\right|^2 \\ &\leq \sum_{n\in[N]^d} B^2\left|W_t\mathcal{E}_t^{(0)}(x_n) + \mathcal{E}_t^{(3)}(x_n)\right|^2 \\ \|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2(n\in[N]^d)} &\leq B\Big(\|W_t\|_2\|\mathcal{E}_t^{(0)}\|_{\ell^2(n\in[N]^d)} + \|\mathcal{E}_t^{(3)}\|_{\ell^2(n\in[N]^d)}\Big),\end{aligned}$$

where $\|\cdot\|_2$ is the matrix-2 norm. Recall B bounds derivatives of σ in Assumptions 4.3.1.

□

The results of Proposition C.3.1 allow us to easily prove the following lemma.

Under Assumptions 4.3.1, the following bound holds:

$$\frac{1}{N^{d/2}}\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2(n\in[N]^d)} \leq BM\left(\frac{2}{N^{d/2}}\|\mathcal{E}_t^{(0)}\|_{\ell^2(n\in[N]^d)} + \alpha_{d,s}N^{-s}\|v_t\|_{H^s}\right), \quad \text{(C.3.3)}$$

where $\alpha_{d,s}$ is a constant dependent only on d and s.

Proof. From Proposition C.3.1, and shortening the notation $\ell^2(n \in [N]^d)$ to ℓ^2,

$$\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2} \leq B\Big(\|W_t\|_2\|\mathcal{E}_t^{(0)}\|_{\ell^2} + N^{d/2}\|P_t\|_F\Big(\alpha_{d,s}N^{-s}\|v_t\|_{H^s} + N^{-d/2}\|\mathcal{E}_t^{(0)}\|_{\ell^2}\Big)\Big).$$

Combining terms gives

$$\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2} \leq B\Big(\big(\|W_t\|_2 + \|P_t\|_F\big)\|\mathcal{E}_t^{(0)}\|_{\ell^2} + \alpha_{d,s}N^{d/2-s}\|P_t\|_F\|v_t\|_{H^s}\Big). \quad \text{(C.3.4)}$$

Replacing $\|W_t\|_2$ and $\|P_t\|_F$ with M and rescaling gives

$$\frac{1}{N^{d/2}}\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2(n\in[N]^d)} \leq BM\left(\frac{2}{N^{d/2}}\|\mathcal{E}_t^{(0)}\|_{\ell^2(n\in[N]^d)} + \alpha_{d,s}N^{-s}\|v_t\|_{H^s}\right).$$

□

C.4 Proofs of regularity theory lemmas

The proof of Lemma 4.3 relies on another result for bounding the H^s norm of compositions of functions, which is largely taken from the lemma of Moser [98, sec. 2 , p. 273] without assuming an L^∞ norm of v less than 1. We state a proof here for completeness.

***Lemma* C.4.1.** Assume $\varphi : \mathbb{T}^d \to \mathbb{T}^d$ possesses continuous derivatives up to order r which are bounded by B. Then

$$|\varphi \circ v|_r \leq Bc\big(1 + \|v\|_\infty^{r-1}\big)\|v\|_{H^r}$$

provided $v \in H^r(\mathbb{T}^d)$, where c is a constant dependent on r and d. ◊

Proof. By Faà di Bruno's formula, we have

$$D_x^r(\varphi \circ v(x)) = \sum C_{\alpha,r} \frac{d^\rho \varphi}{dx^\rho}(v(x)) \prod_{j=1}^r (D_x^j v(x))^{\alpha_j}, \tag{C.4.1}$$

where the sum is over all nonnegative integers $\alpha_1, \ldots, \alpha_r$ such that $\alpha_1 + 2\alpha_2 + \cdots + r\alpha_r = r$, the constant $C_{\alpha,r} = \frac{r!}{\alpha_1!\alpha_2!2!^{\alpha_2}\ldots\alpha_r!r!^{\alpha_r}}$, and $\rho := \alpha_1 + \alpha_2 + \cdots + \alpha_r$.

We seek a bound on square integrals of (C.4.1). Setting $v_0 = \frac{d^\rho \varphi}{dx^\rho} v$, $v_\lambda = D_x^\lambda v$, $\alpha_0 = 1$, $p_0 = \infty$, and $p_\lambda = \frac{r}{\lambda \alpha_\lambda}$ and noting that $\sum_{\lambda=0}^r \frac{1}{2p_\lambda} = \frac{1}{2}$, we have by Hölder's inequality for multiple products that

$$\begin{aligned}\int_{\mathbb{T}^d} \left| \frac{d^\rho \varphi}{dx^\rho}(v(x)) \prod_{j=1}^r (D_x^j v(x))^{\alpha_j} \right|^2 \mathsf{d}x &\leq \int_{\mathbb{T}^d} \prod_{\lambda=0}^r |v_\lambda|^{2\alpha_\lambda} \, \mathsf{d}x \leq \prod_{\lambda=0}^r \left(\int_{\mathbb{T}^d} |v_\lambda|^{2\alpha_\lambda p_\lambda} \, \mathsf{d}x \right)^{1/p_\lambda} \\ &= \|v_0\|_\infty^2 \prod_{\lambda=1}^r \left(\int_{\mathbb{T}^d} |v_\lambda|^{2\alpha_\lambda p_\lambda} \, \mathsf{d}x \right)^{1/p_\lambda}.\end{aligned}$$

The first factor is bounded above by B^2 by assumption. By application of Gagliardo-Nirenberg, the second factor may be bounded by

$$\begin{aligned}\prod_{\lambda=1}^r \left(\int_{\mathbb{T}^d} |D_x^\lambda v|^{2r/\lambda} \, \mathsf{d}x \right)^{\lambda\alpha_\lambda/r} &\leq C^r \prod_{\lambda=1}^r \|v\|_\infty^{2\alpha_\lambda(1-\lambda/r)} \big(\|D_x^r v\|^2 + \|v\|^2 \big)^{\alpha_\lambda \lambda / r} \\ &\leq C^r \|v\|_\infty^{2\rho-2} \|v\|_{H^r}^2\end{aligned}$$

since $\sum_\lambda \lambda\alpha_\lambda = r$, and $\sum_\lambda \alpha_\lambda = \rho$. Combining the bounds,

$$\int_{\mathbb{T}^d} \prod_{\lambda=0}^r |v_\lambda|^{2\alpha_\lambda} \, \mathsf{d}x \leq B^2 C^r \|v\|_\infty^{2\rho-2} \|v\|_{H^r}^2.$$

If $\|v\|_\infty < 1$, we have the bound

$$\int_{\mathbb{T}^d} \prod_{\lambda=0}^r |v_\lambda|^{2\alpha_\lambda} \, \mathsf{d}x \leq B^2 C^r \|v\|_{H^r}^2, \tag{C.4.2}$$

and otherwise since $\rho \leq r$,

$$\int_{\mathbb{T}^d} \prod_{\lambda=0}^r |v_\lambda|^{2\alpha_\lambda} \, \mathsf{d}x \leq B^2 C^r \|v\|_\infty^{2r-2} \|v\|_{H^r}^2. \tag{C.4.3}$$

Since these bounds hold for any term in the sum C.4.1, we obtain

$$|\varphi \circ v|_r \le Bc\big(1 + \|v\|_\infty^{r-1}\big)\|v\|_{H^r} \tag{C.4.4}$$

for a different constant c depending on r and d. □

Now we may prove Lemma 4.3. Under Assumptions 4.3.1, the following bounds hold:

- $\|v_{t+1}\|_\infty \le \sigma_0 + BM(1 + \|v_t\|_\infty + K^{d/2}\|v_t\|_{L^2(\mathbb{T}^d)})$
- $|v_{t+1}|_s \le BcM^s K^{ds/2}(1 + \|v_t\|_\infty)^s(1 + |v_t|_s)$

for some constant c dependent on d and s, where $\sigma_0 := \max\{\max_{0\le t\le T}\sigma_t(0), 1\}$.

Proof. First we bound $\|\mathcal{K}_t v_t\|_\infty$. Recall $\widehat{v}_t(k) := \int_{\mathbb{T}^d} v_t(x)e^{-2\pi i\langle k,x\rangle}\,\mathsf{d}x$.

$$\begin{aligned}
\|\mathcal{K}_t v_t\|_\infty &= \|\sum_{k\in[[K]]^d} P_t^{(k)}\widehat{v}_t(k)e^{2\pi i\langle k,x\rangle}\|_\infty \\
&\le \sum_{k\in[[K]]^d} \|P_t^{(k)}\||\widehat{v}_t(k)| \\
&\le \left(\sum_{k\in[[K]]^d} \|P_t^{(k)}\|^2\right)^{1/2} \|\widehat{v}_t\|_{\ell^2(k\in[[K]]^d)} \\
&\le \|P_t\|_F K^{d/2}\|\widehat{v}_t\|_{\ell^2(k\in[[K]]^d)} \\
&\le \|P_t\|_F K^{d/2}\|v_t\|_{L^2(\mathbb{T}^d)}.
\end{aligned}$$

Then

$$\|W_t v_t + \mathcal{K}_t v_t + b_t\|_\infty \le \|W_t\|_2\|v_t\|_\infty + |b_t| + \|P_t\|_F K^{d/2}\|v_t\|_{L^2(\mathbb{T}^d)},$$

and by Lipschitzness of σ we have

$$\|v_{t+1}\|_\infty \le \sigma^* + BM\big(1 + \|v_t\|_\infty + K^{d/2}\|v_t\|_{L^2(\mathbb{T}^d)}\big).$$

Next we bound $|v_{t+1}|_s$. Letting $f_t = W_t v_t + \mathcal{K}_t v_t + b_t$, we see from Lemma C.4.1 that bounding $\|f_t\|_{H^s}$ will give the result.

$$D_x^s(f_t) = W_t(D_x^s v_t) + \mathcal{K}_t(D_x^s v_t).$$

$$\int_{\mathbb{T}^d} |D_x^s(f_t)|^2\,\mathsf{d}x \le 2\left(\int_{\mathbb{T}^d} |W_t(D_x^s v_t)|^2\,\mathsf{d}x + \int_{\mathbb{T}^d} |\mathcal{K}_t(D_x^s v_t)|^2\,\mathsf{d}x\right).$$

The first integral on the right may be bounded by $\|W_t\|_2^2 |v_t|_s^2$. To bound the second integral,

$$\int_{\mathbb{T}^d} |\mathcal{K}_t(D_x^s v_t)|^2 \, \mathrm{d}x = \int_{\mathbb{T}^d} \left| \sum_{k \in [[K]]^d} P_t^{(k)} \widehat{g}_t(k) e^{2\pi i \langle k, x \rangle} \right|^2 \mathrm{d}x,$$

where $\widehat{g}_t(k)$ are the Fourier coefficients of $D_x^s v_t$. Continuing,

$$\begin{aligned} \int_{\mathbb{T}^d} |\mathcal{K}_t(D_x^s v_t)|^2 \, \mathrm{d}x &\leq \int_{\mathbb{T}^d} \|P_t\|_F^2 \sum_{k \in [[K]]^d} |\widehat{g}_t(k)|^2 \, \mathrm{d}x \\ &\leq \|P_t\|_F^2 \|D_x^s v_t\|_{L^2}^2, \end{aligned}$$

giving a bound of

$$|f_t|_s \leq 2M |v_t|_s.$$

In the following, $\lesssim$ denotes inequality up to a constant multiple that does not depend on any of the variables involved. Combining Lemma C.4.1 and the above bounds, we have

$$\begin{aligned} |\sigma \circ f_t|_s &\leq Bc(1 + \|f_t\|_\infty^{s-1}) \|f_t\|_{H^s} \\ &\leq Bc(1 + (M(1 + \|v_t\|_\infty + K^{d/2}\|v_t\|_\infty))^{s-1})(M(1 + \|v_t\|_\infty + K^{d/2}\|v_t\|_\infty) + 2M|v_t|_s) \\ &\lesssim BcM^s K^{ds/2}(1 + (1 + \|v_t\|_\infty)^{s-1})(1 + \|v_t\|_\infty + |v_t|_s) \\ &\lesssim BcM^s K^{ds/2}(1 + \|v_t\|_\infty)^{s-1}(1 + \|v_t\|_\infty)(1 + |v_t|_s) \\ &\lesssim BcM^s K^{ds/2}(1 + \|v_t\|_\infty)^s (1 + |v_t|_s). \end{aligned}$$

□

C.5 Proof of Theorem 4.3.2

***Theorem* 4.3.2.** Let Assumptions 4.3.1 hold. Let $\mathcal{A}_c$ be a compact set in $\mathcal{A}$. Let $v_t(a) := \mathsf{L}_t \circ \mathsf{L}_{t-1} \cdots \circ \mathsf{L}_0 \circ \mathcal{P}(a)$ with $\mathcal{P}$ and each L as defined in Definition 1.3.1. Similarly, let $v_t^N(a) := \mathsf{L}_t^N \circ \mathsf{L}_{t-1}^N \cdots \circ \mathsf{L}_0^N \circ \mathcal{P}(a)$ where $\mathsf{L}_j^N v_j^N = \sigma_j(W_j v_j^N + \mathcal{K}_j^N v_j + b_j)$ for $\mathcal{K}_j^N$ defined in (4.3.1) for each $0 \leq j \leq t$. Then

$$\sup_{a \in \mathcal{A}_c} \frac{1}{N^{d/2}} \|v_t(a) - v_t^N(a)\|_{\ell^2(n \in [N]^d)} \leq CN^{-s}, \tag{4.3.2}$$

where the constant C depends on B, M, d, s, t, and $\mathcal{A}_c$. ◊

Proof. Temporarily dropping the notational dependence of v_t and v_0 on a, from Lemma 4.3 we have for $t \geq 1$,

$$\|v_t\|_\infty \lesssim \sigma^* \sum_{j=0}^{t-1} (BMK^{d/2})^j + \sum_{j=1}^{t} (BMK^{d/2})^j + (BMK^{d/2})^t \|v_0\|_\infty$$

$$|v_t|_s \lesssim \left(\sum_{j=1}^{t} (BcM^s K^{ds/2})^j \prod_{\ell=t-j}^{t-1} (1 + \|v_\ell\|_\infty)^s \right) + (BcM^s K^{ds/2})^t \left(\prod_{\ell=0}^{t-1} (1 + \|v_\ell\|_\infty)^s \right) |v_0|_s.$$

Denote $\max\{BMK^{d/2}, B^{1/s} c^{1/s} M K^{d/2}, 1\}$ by C_0. Since $\sigma^* \geq 1$, the bound on $\|v_t\|_\infty$ simplifies to

$$\begin{aligned} \|v_t\|_\infty &\lesssim \sigma^* \sum_{j=1}^{t} C_0^j + C_0^t \|v_0\|_\infty \\ &\leq \sigma^* t C_0^t + C_0^t \|v_0\|_\infty. \end{aligned}$$

Plugging in this bound to the product in the bound on $|v_t|_s$, we have

$$\begin{aligned} \prod_{\ell=t-j}^{t-1} (1 + \|v_\ell\|_\infty)^s &\lesssim \prod_{\ell=t-j}^{t-1} (1 + \ell \sigma^* C_0^\ell + C_0^\ell \|v_0\|_\infty)^s \\ &\lesssim C_0^{tsj} (t)^{sj} (\sigma^* + \|v_0\|_\infty)^{sj}. \end{aligned}$$

Combining these two bounds, we attain the following bound on $|v_t|_s$ for $t \geq 1$.

$$\begin{aligned} |v_t|_s &\lesssim \left(\sum_{j=1}^{t} (C_0)^{sj} C_0^{tsj} (t)^{sj} (\sigma^* + \|v_0\|_\infty)^{sj} \right) + C_0^{ts} \left(C_0^{t^2 s} (t)^{st} (\sigma^* + \|v_0\|_\infty)^{st} \right) |v_0|_s \\ &\lesssim \left(\sum_{j=1}^{t} C_0^{2tsj} (t)^{sj} (\sigma^* + \|v_0\|_\infty)^{sj} \right) + C_0^{2t^2 s} (t)^{st} (\sigma^* + \|v_0\|_\infty)^{st} |v_0|_s \\ &\lesssim (C_0^{2t^2 s} t^{st+1} + C_0^{2t^2 s} t^{st} |v_0|_s)(\sigma^* + \|v_0\|_\infty)^{st} \end{aligned}$$

and the following bound on $\|v_t\|_{H^s}$

$$\|v_t\|_{H^s} \lesssim (C_0^{2t^2 s} t^{st+1} |v_0|_s)(\sigma^* + \|v_0\|_\infty)^{st} + \sigma^* t C_0^t + C_0^t \|v_0\|_\infty. \tag{C.5.1}$$

Recall that $v_0 = \mathcal{P}(a)$, and $\mathcal{P}$ is a shallow neural network, which is a special case of a Fourier layer where the coefficients $P_t^{(k)}$ are set to 0. Assumptions 4.3.1 include boundedness of the coefficients of $\mathcal{P}$ by M. Thus we may increment t by 1 in the bound and write

$$\sup_{a \in \mathcal{A}_c} \|v_t(a)\|_{H^s} \tag{C.5.2a}$$

$$\lesssim \sup_{a \in \mathcal{A}_c} (C_0^{2(t+1)^2 s} (t+1)^{s(t+1)+1} |a|_s)(\sigma^* + \|a\|_\infty)^{s(t+1)} + \sigma^* (t+1) C_0^{t+1} + C_0^{t+1} \|a\|_\infty. \tag{C.5.2b}$$

Since $\mathcal{A}$ is a compact set in H^s, and $s > \frac{d}{2}$, both $\|a\|_\infty$ and $|a|_s$ are bounded uniformly over $\mathcal{A}$ by a constant depending on $\mathcal{A}$ since H^s is continuously embedded in L^∞. Thus, we may denote this upper bound by C_1, which does not depend on N. Let $\mathcal{E}_{t+1}^{(0)}(a) = v_t^N(a) - v_t(a)$. Then from Lemma C.3, we have

$$\sup_{a\in\mathcal{A}_c} \frac{1}{N^{d/2}}\|\mathcal{E}_{t+1}^{(0)}(a)\|_{\ell^2(n\in[N]^d)} \lesssim BM\left(\frac{2}{N^{d/2}}\sup_{a\in\mathcal{A}_c}\|\mathcal{E}_t^{(0)}(a)\|_{\ell^2(n\in[N]^d)} + \alpha_{d,s}N^{-s}C_1\right).$$

By the discrete Gronwall lemma,

$$\begin{aligned}&\sup_{a\in\mathcal{A}_c} \frac{1}{N^{d/2}}\|\mathcal{E}_t^{(0)}(a)\|_{\ell^2(n\in[N]^d)}\\ &\lesssim \frac{BM\alpha_{d,s}N^{-s}C_1}{1-2BM}(1-(2BM)^t) + \frac{1}{N^{d/2}}\sup_{a\in\mathcal{A}_c}\|\mathcal{E}_0^{(0)}(a)\|_{\ell^2(n\in[N]^d)}(2BM)^t.\end{aligned}$$

Since we assume we begin with no error, $\|\mathcal{E}_0^{(0)}(a)\|_{\ell^2(n\in[N]^d)} = 0$, this simplifies to

$$\sup_{a\in\mathcal{A}_c} \frac{1}{N^{d/2}}\|\mathcal{E}_t^{(0)}(a)\|_{\ell^2(n\in[N]^d)} \lesssim \frac{BM\alpha_{d,s}C_1}{1-2BM}(1-(2BM)^t)N^{-s}.$$

Denoting the factor in front of N^{-s} by C and absorbing the effects of $\lesssim$ into C, we have the result that

$$\sup_{a\in\mathcal{A}_c} \frac{1}{N^{d/2}}\|v_t(a) - v_t^N(a)\|_{\ell^2(n\in[N]^d)} \le CN^{-s}.$$

□

***Remark* C.5.1.** A trivial consequence of the above theorem is that under Assumptions 4.3.1,

$$\lim_{N\to\infty}\sup_{a\in\mathcal{A}_c} \frac{1}{N^{d/2}}\|v_t^N(a) - v_t(a)\|_{\ell^2(n\in[N]^d)} = 0.$$

Indeed, a stronger result holds that the discrete ℓ^∞ norm converges at a rate $N^{-s+d/2}$ by a straightforward inverse inequality. ◊

C.6 Proof of Theorem 4.3.3

***Theorem* 4.3.3.** Let $p_t^N(x) = \sum_{k\in[[N]]^d} \mathsf{DFT}(v_t^N)(k)e^{2\pi i\langle k,x\rangle}$ denote the interpolating trigonometric polynomial of $\{v_t^N(x_n)\}_{n\in[N]^d}$. Let the assumptions of Theorem 4.3.2 hold. Then,

$$\sup_{a\in\mathcal{A}_c}\|v_t(a) - p_t^N(a)\|_{L^2(\mathbb{T}^d)} \le C'N^{-s}. \tag{4.3.3}$$

Here, C' depends on B, M, d, s, t, and $\mathcal{A}$. ◊

Proof. We temporarily drop the dependence of p_t^N and v_t^N on a. Let $p_t^N(x)$ be the interpolating trigonometric polynomial associated with the data $\{v_t^N(x_n)\}_{n\in[N]^d}$. By Proposition C.1.6, we have

$$\|v_t - p_t^N\|_{L^2(\mathbb{T}^d)} \leq \frac{1}{N^{d/2}}\|v_t - v_t^N\|_{\ell^2(n\in[N]^d)} + c_{d,s}\|v\|_{H^s(\mathbb{T}^d)}N^{-s}.$$

By (C.5.2), we have $\sup_{a\in\mathcal{A}_c}\|v_t(a)\|_{H^s(\mathbb{T}^d)} \leq C_1$. Furthermore, it follows from Theorem 4.3.2, that

$$\sup_{a\in\mathcal{A}_c}\frac{1}{N^{d/2}}\|v_t(a) - v_t^N(a)\|_{\ell^2(n\in[N]^d)} \leq CN^{-s}.$$

We conclude that

$$\sup_{a\in\mathcal{A}_c}\|v_t(a) - p_t^N(a)\|_{L^2(\mathbb{T}^d)} \leq (C + c_{d,s}C_1)N^{-s}.$$

Thus, the claimed bound holds with $C' = C + c_{d,s}C_1$.

□

C.7 Additional numerical results

Figure C.1 addresses the question of error decreasing or increasing with layer count. The figure shows that when the FNO weights are randomly initialized with the default initialization and then multiplied by 10, the error increases with the number of layers instead of decreases. Additionally, in this model, the large weights mean that the GeLU activation acts like a ReLU activation for smaller discretizations. This phenomenon is apparent for inputs with regularity $s = 2$, where the first layer has the appropriate slope, but the other layers only begin to approach that rate at higher discretizations. Earlier layers achieve this rate first because of the smaller magnitude state norm in earlier layers for this model.

As an alternative setting of the weights, Figure C.2 shows the discretization error when all the weights are set to 1. In this case, the error is more erratic. The error decreases faster than expected and with less consistency than the Gaussian weight models, and the decay rate increases with each layer. In this sense, the all-ones model has a smoothing effect on the state at each layer. We note that this generally occurs with any initialization that sets the spectral weights on the same order of magnitude as the affine weights; for instance, the same super-convergence effect occurs when all weights are initialized $U(0, 1)$.

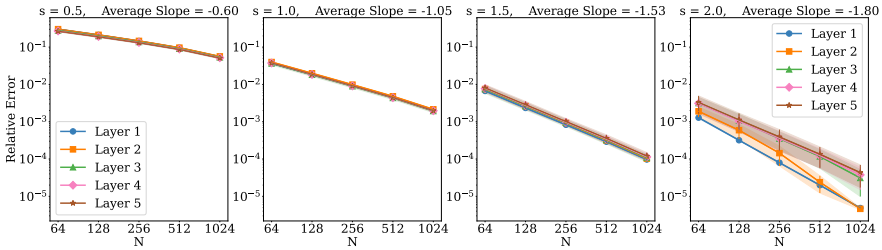


Figure C.1: Relative error versus N and s for an FNO with default $\times 10$ initial weights.

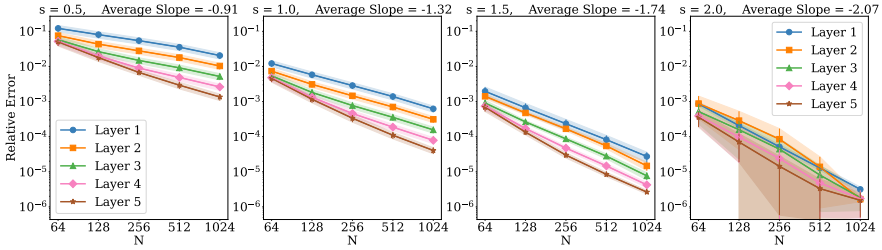


Figure C.2: Relative error versus N and s for an FNO with all weights equals to 1.

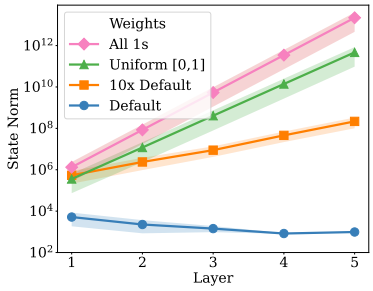


Figure C.3: State norm versus layer for various untrained model initializations.

We hypothesize that this is because when the spectral weights are of equal magnitude to the affine weights, the function is progressively smoothed as it passes through the model.

We can also observe the state norm as the state passes through the layers for various settings of the weights. Indeed, for all choices of initialization that we explored except the default setting, the state norm increases exponentially through the layers, while for the default initialization the magnitude stays roughly constant. This phenomenon is illustrated in Figure C.3.

C.8 Additional implementation details for error analysis experiments

All the trained models were trained on an Nvidia P100 GPU for approximately 6 hours. The evaluation scripts were run on a Mac laptop with an M2 processor.

C.9 Implementation details for adaptive subsampling

Our model has 4 hidden layers, channel width 64, and Fourier cut-off 12. Our results are based on 9000 training samples and 500 test samples. For training with a subsampling scheduler, we include an additional 500 samples for validation. Models are trained for 300 epochs on an Nvidia P100 GPU.

Appendix D

APPENDIX TO CHAPTER 5

Data and code availability

Links to datasets and all code used to produce the numerical results and figures in this chapter are available at

https://github.com/nickhnelsen/fourier-neural-mappings .

D.1 Proofs for Section 5.3: Universal approximation theory for Fourier Neural Mappings

This appendix begins with some universal approximation results for neural operators before establishing similar universal approximation results for neural mappings (i.e., neural functionals and decoders).

Supporting approximation results for neural operators

We need the following two lemmas that are simple generalizations of the universal approximation theorem for FNOs [20, Theorem 9, p. 9] to the setting where only one of the input or output domain is the torus. These results may be extracted from the proof of [20, Theorem 9, p. 9].

***Lemma* D.1.1** (universal approximation for FNO: periodic output domain)**.** Let Assumption 5.3.1 hold. Let $s \geq 0$ and $s' \geq 0$, $\mathcal{D} \subset \mathbb{R}^d$ be an open Lipschitz domain such that $\overline{\mathcal{D}} \subset (0,1)^d$, and $\mathcal{U} = H^s(\mathcal{D}; \mathbb{R}^{d_u})$. Let $\mathcal{Y} = H^{s'}(\mathbb{T}^d; \mathbb{R}^{d_y})$ and $\mathcal{G} \colon \mathcal{U} \to \mathcal{Y}$ be a continuous operator. There exists a continuous linear extension operator $E \colon \mathcal{U} \to H^s(\mathbb{T}^d; \mathbb{R}^{d_u})$ such that $(Eu)|_{\mathcal{D}} = u$ for all $u \in \mathcal{U}$. Moreover, let $K \subset \mathcal{U}$ be compact in $\mathcal{U}$. For any $\varepsilon > 0$, there exists a Fourier Neural Operator $\Psi \colon H^s(\mathbb{T}^d; \mathbb{R}^{d_u}) \to \mathcal{Y}$ of the form (5.2.4) (with $\mathcal{E} = \mathrm{Id}$, $\mathcal{R} = \mathrm{Id}$, and items (*i*) and (*ii*) both holding true) such that

$$\sup_{u \in K} \|\mathcal{G}(u) - \Psi(Eu)\|_{\mathcal{Y}} < \varepsilon \,. \tag{D.1.1}$$

◊

The next lemma is analogous to the previous one and deals with periodic input domains.

***Lemma* D.1.2** (universal approximation for FNO: periodic input domain)**.** Let Assumption 5.3.1 hold. Let $s \geq 0$ and $s' \geq 0$, $\mathcal{D} \subset \mathbb{R}^d$ be an open Lipschitz domain such that $\overline{\mathcal{D}} \subset (0,1)^d$, and $\mathcal{U} = H^s(\mathbb{T}^d; \mathbb{R}^{d_u})$. Let $\mathcal{Y} = H^{s'}(\mathcal{D}; \mathbb{R}^{d_y})$ and $\mathcal{G} \colon \mathcal{U} \to \mathcal{Y}$ be a continuous operator. Denote by $R \in \mathcal{L}(H^{s'}(\mathbb{T}^d; \mathbb{R}^{d_y}); \mathcal{Y})$ the restriction operator $y \mapsto y|_{\mathcal{D}}$. Let $K \subset \mathcal{U}$ be compact in $\mathcal{U}$. For any $\varepsilon > 0$, there exists a Fourier Neural Operator $\Psi \colon \mathcal{U} \to H^{s'}(\mathbb{T}^d; \mathbb{R}^{d_y})$ of the form (5.2.4) (with $\mathcal{E} = \mathrm{Id}$, $\mathcal{R} = \mathrm{Id}$, and items *(i)* and *(ii)* both holding true) such that

$$\sup_{u \in K} \|\mathcal{G}(u) - R\Psi(u)\|_{\mathcal{Y}} < \varepsilon \,. \tag{D.1.2}$$

◊

Universal approximation proofs

The remainder of this appendix provides proofs of the main universal approximation theorems found in Section 5.3 for the proposed FNM family of architectures. We begin with the function-to-vector Fourier Neural Functionals (FNF) architecture.

***Theorem* 5.3.2** (universal approximation: function-to-vector mappings)**.** Let $s \geq 0$, $\mathcal{D} \subset \mathbb{R}^d$ be an open Lipschitz domain such that $\overline{\mathcal{D}} \subset (0,1)^d$, and $\mathcal{U} = H^s(\mathcal{D}; \mathbb{R}^{d_u})$. Let $\Psi^\dagger \colon \mathcal{U} \to \mathbb{R}^{d_y}$ be a continuous mapping. Let $K \subset \mathcal{U}$ be compact in $\mathcal{U}$. Under Assumption 5.3.1, for any $\varepsilon > 0$, there exist Fourier Neural Functionals $\Psi \colon \mathcal{U} \to \mathbb{R}^{d_y}$ of the form (5.2.8) with modification (M-F2V) such that

$$\sup_{u \in K} \left\| \Psi^\dagger(u) - \Psi(u) \right\|_{\mathbb{R}^{d_y}} < \varepsilon \,. \tag{5.3.1}$$

◊

Proof. Let $\mathcal{Y} \coloneqq L^2(\mathbb{T}^d; \mathbb{R}^{d_y})$ and $\mathbf{1} \colon x \mapsto 1$ be the constant function on $\mathbb{T}^d$. We first convert the function-to-vector mapping $\Psi^\dagger$ to the function-to-function operator $\mathcal{G}^\dagger \colon \mathcal{U} \to \mathcal{Y}$ defined by $u \mapsto \Psi^\dagger(u)\mathbf{1}$. We then establish the existence of a FNO that approximates $\mathcal{G}^\dagger$. Finally, from this FNO we construct a FNF that approximates $\Psi^\dagger$. To this end, fix $\varepsilon' > 0$. By the continuity of $\Psi^\dagger$, there

exists $\delta > 0$ such that $\|u_1 - u_2\|_{\mathcal{U}} < \delta$ implies $\|\Psi^\dagger(u_1) - \Psi^\dagger(u_2)\|_{\mathbb{R}^{d_y}} < \varepsilon'$. Then

$$\begin{aligned}\|\mathcal{G}^\dagger(u_1) - \mathcal{G}^\dagger(u_2)\|_{\mathcal{Y}}^2 &= \int_{\mathbb{T}^d} \|\Psi^\dagger(u_1)\mathbf{1}(x) - \Psi^\dagger(u_2)\mathbf{1}(x)\|_{\mathbb{R}^{d_y}}^2 \, dx \\ &= |\mathbb{T}^d| \|\Psi^\dagger(u_1) - \Psi^\dagger(u_2)\|_{\mathbb{R}^{d_y}}^2 \\ &< (\varepsilon')^2 .\end{aligned}$$

We used the fact that $|\mathbb{T}^d| = 1$ for the identification $\mathbb{T}^d \equiv (0,1)^d_{\mathrm{per}}$. This shows the continuity of $\mathcal{G}^\dagger \colon \mathcal{U} \to \mathcal{Y}$. By the universal approximation theorem for FNOs (Lemma D.1.1, applied with $s = s$, $s' = 0$, $d_y = d_y$, and $\mathcal{G} = \mathcal{G}^\dagger$), there exists a continuous linear operator $E \colon \mathcal{U} \to H^s(\mathbb{T}^d; \mathbb{R}^{d_u})$ and a FNO $\mathcal{G} \colon H^s(\mathbb{T}^d; \mathbb{R}^{d_u}) \to \mathcal{Y}$ of the form (5.2.4) (with $\mathcal{R} = \mathrm{Id}$, $\mathcal{E} = \mathrm{Id}$, and items *(i)* and *(ii)* both holding true) such that

$$\sup_{u \in K} \|\mathcal{G}^\dagger(u) - \mathcal{G}(Eu)\|_{\mathcal{Y}} < \varepsilon .$$

To complete the proof, we construct a FNF by appending a specific linear layer to the output of $\mathcal{G} \circ E$. To this end, let $\overline{P} \colon \mathcal{Y} \to \mathbb{R}^{d_y}$ be the averaging operator

$$u \mapsto \overline{P}u \coloneqq \int_{\mathbb{T}^d} u(x) \, dx .$$

Clearly $\overline{P}$ is linear. It is continuous on $\mathcal{Y}$ because

$$\|\overline{P}u\|_{\mathbb{R}^{d_y}} \le \int_{\mathbb{T}^d} \|u(x)\|_{\mathbb{R}^{d_y}} \mathbf{1}(x) \, dx \le \|u\|_{\mathcal{Y}}$$

by the triangle and Cauchy–Schwarz inequalities. Now define $\Psi \coloneqq (\overline{P} \circ \mathcal{G} \circ E) \colon \mathcal{U} \to \mathbb{R}^{d_y}$. This map has the representation

$$\Psi = \overline{P} \circ \widetilde{\mathcal{Q}} \circ F \circ \widetilde{\mathcal{S}} \circ E$$

for some local linear operators $\widetilde{\mathcal{Q}}$ (identified with $\widetilde{Q} \in \mathbb{R}^{d_y \times d_v}$ for channel dimension d_v) and $\widetilde{\mathcal{S}}$ (identified with $\widetilde{S} \in \mathbb{R}^{d_v \times d_u}$), and where F denotes the repeated composition of all nonlinear FNO layers of the form $\mathscr{L}_t$ as in (5.2.2). We claim that Ψ belongs to the FNF class, i.e., (5.2.8) with modification (M-F2V). To see this, choose $Q = I_{\mathbb{R}^{d_y}} \in \mathbb{R}^{d_y \times d_y}$ and $S = I_{\mathbb{R}^{d_u}} \in \mathbb{R}^{d_u \times d_u}$ (which we identify with $\mathrm{Id}_{\mathcal{U}} \in \mathcal{L}(\mathcal{U})$). Let $\mathcal{E} \coloneqq (\widetilde{\mathcal{S}} \circ E) \colon \mathcal{U} \to H^s(\mathbb{T}^d; \mathbb{R}^{d_v})$. Define the linear functional layer $\mathscr{G} \coloneqq (\overline{P} \circ \widetilde{\mathcal{Q}}) \colon L^2(\mathbb{T}^d; \mathbb{R}^{d_v}) \to \mathbb{R}^{d_y}$ which has the kernel linear functional representation

$$u \mapsto \mathscr{G}u = \int_{\mathbb{T}^d} \kappa(x) u(x) \, dx , \quad \text{where} \quad x \mapsto \kappa(x) \coloneqq \mathbf{1}(x)\widetilde{Q} \in \mathbb{R}^{d_y \times d_v}$$

as in (5.2.5). Thus,

$$\begin{aligned}\Psi &= \overline{P} \circ \widetilde{\mathcal{Q}} \circ F \circ \widetilde{\mathcal{S}} \circ E \\ &= I_{\mathbb{R}^{d_y}} \circ (\overline{P} \circ \widetilde{\mathcal{Q}}) \circ F \circ (\widetilde{\mathcal{S}} \circ E) \circ \mathrm{Id}_{\mathcal{U}} \\ &= Q \circ \mathscr{G} \circ F \circ \mathcal{E} \circ S\end{aligned}$$

as claimed. Finally, using the fact that $\overline{P}(z\mathbf{1}) = z$ for any $z \in \mathbb{R}^{d_y}$, it holds that

$$\sup_{u\in K}\|\Psi^\dagger(u) - \Psi(u)\|_{\mathbb{R}^{d_y}} = \sup_{u\in K}\|\overline{P}\mathcal{G}^\dagger(u) - \overline{P}\mathcal{G}(Eu)\|_{\mathbb{R}^{d_y}} \le \sup_{u\in K}\|\mathcal{G}^\dagger(u) - \mathcal{G}(Eu)\|_{\mathcal{Y}}\,.$$

The rightmost expression is less than ε and hence (5.3.1) holds. □

The universality proof for the vector-to-function Fourier Neural Decoder (FND) architecture follows similar arguments.

***Theorem* 5.3.3** (universal approximation: vector-to-function mappings)**.** Let $t \ge 0$, $\mathcal{D} \subset \mathbb{R}^d$ be an open Lipschitz domain such that $\overline{\mathcal{D}} \subset (0,1)^d$, and $\mathcal{Y} = H^t(\mathcal{D};\mathbb{R}^{d_y})$. Let $\Psi^\dagger\colon \mathbb{R}^{d_u} \to \mathcal{Y}$ be a continuous mapping. Let $\mathcal{Z} \subset \mathbb{R}^{d_u}$ be compact. Under Assumption 5.3.1, for any $\varepsilon > 0$, there exists a Fourier Neural Decoder $\Psi\colon \mathbb{R}^{d_u} \to \mathcal{Y}$ of the form (5.2.8) with modification (M-V2F) such that

$$\sup_{z\in\mathcal{Z}}\big\|\Psi^\dagger(z) - \Psi(z)\big\|_{\mathcal{Y}} < \varepsilon\,. \tag{5.3.2}$$

◊

Proof. Let $\mathcal{U} \coloneqq L^2(\mathbb{T}^d;\mathbb{R}^{d_u})$ and $\mathbf{1}\colon \mathbb{T}^d \to \mathbb{R}$ be the constant function $x \mapsto 1$. Define the map $\mathsf{L}\colon \mathbb{R}^{d_u} \to \mathcal{U}$ by $z \mapsto z\mathbf{1}$. Clearly L is linear. To see that it is continuous, we compute

$$\|\mathsf{L}z\|_{\mathcal{U}}^2 = \int_{\mathbb{T}^d}\|z\mathbf{1}(x)\|_{\mathbb{R}^{d_u}}^2\,dx = |\mathbb{T}^d|\|z\|_{\mathbb{R}^{d_u}}^2 = \|z\|_{\mathbb{R}^{d_u}}^2\,. \tag{D.1.3}$$

Thus, L is injective with $\|\mathsf{L}\|_{\mathcal{L}(\mathbb{R}^{d_u};\mathcal{U})} = 1$. Choose $K \coloneqq \mathsf{L}\mathcal{Z} = \{\mathsf{L}z\colon z \in \mathcal{Z}\} \subset \mathcal{U}$, which is compact in $\mathcal{U}$ because continuous functions map compact sets to compact sets. Define $\mathcal{G}^\dagger\colon K \to \mathcal{Y}$ by $\mathsf{L}z \mapsto \Psi^\dagger(z)$. First, we show that $\mathcal{G}^\dagger$ is continuous. Fix $\varepsilon' > 0$. By the continuity of $\Psi^\dagger$, there exists $\delta > 0$ such that if $\|\mathsf{L}z_1 - \mathsf{L}z_2\|_{\mathcal{U}} = \|z_1 - z_2\|_{\mathbb{R}^{d_u}} < \delta$, then $\|\Psi^\dagger(z_1) - \Psi^\dagger(z_2)\|_{\mathcal{Y}} < \varepsilon'$. Thus for any $u_1 = \mathsf{L}z_1 \in K$ and $u_2 = \mathsf{L}z_2 \in K$ with $\|u_1 - u_2\|_{\mathcal{U}} < \delta$, we have

$$\|\mathcal{G}^\dagger(u_1) - \mathcal{G}^\dagger(u_2)\|_{\mathcal{Y}} = \|\Psi^\dagger(z_1) - \Psi^\dagger(z_2)\|_{\mathcal{Y}} < \varepsilon'\,.$$

It follows that $\mathcal{G}^\dagger\colon K \to \mathcal{Y}$ is continuous. By the Dugundji extension theorem [182], there exists a continuous operator $\widetilde{\mathcal{G}}^\dagger\colon \mathcal{U} \to \mathcal{Y}$ such that $\widetilde{\mathcal{G}}^\dagger(u) = \mathcal{G}^\dagger(u)$ for every $u \in K$. By the universal approximation theorem for FNOs (Lemma D.1.2, applied with $s = 0$, $s' = t$, $d_u = d_u$, and $\mathcal{G} = \widetilde{\mathcal{G}}^\dagger$), there exists a FNO $\mathcal{G}\colon \mathcal{U} \to H^t(\mathbb{T}^d; \mathbb{R}^{d_y})$ of the form (5.2.4) (with $\mathcal{R} = \mathrm{Id}$, $\mathcal{E} = \mathrm{Id}$, and items (*i*) and (*ii*) both holding true) such that

$$\sup_{u\in K} \|\widetilde{\mathcal{G}}^\dagger(u) - R\mathcal{G}(u)\|_{\mathcal{Y}} = \sup_{u\in K} \|\mathcal{G}^\dagger(u) - R\mathcal{G}(u)\|_{\mathcal{Y}} < \varepsilon\,.$$

In the preceding display, $R \in \mathcal{L}(H^t(\mathbb{T}^d; \mathbb{R}^{d_y}); \mathcal{Y})$ denotes the restriction operator $y \mapsto y|_D$. Now define the map $\Psi \coloneqq (R \circ \mathcal{G} \circ \mathsf{L})\colon \mathbb{R}^{d_u} \to \mathcal{Y}$. This map has the representation

$$\Psi = R \circ \widetilde{\mathcal{Q}} \circ F \circ \widetilde{\mathcal{S}} \circ \mathsf{L}$$

for some local linear operators $\widetilde{\mathcal{Q}}$ (identified with $\widetilde{Q} \in \mathbb{R}^{d_y \times d_v}$ for channel dimension d_v) and $\widetilde{\mathcal{S}}$ (identified with $\widetilde{S} \in \mathbb{R}^{d_v \times d_u}$), and where F denotes the repeated composition of all nonlinear FNO layers of the form $\mathscr{L}_t$ as in (5.2.2). We claim that Ψ is of the FND form, i.e., (5.2.8) with modification (M-V2F). To see this, choose $Q = I_{\mathbb{R}^{d_y}} \in \mathbb{R}^{d_y \times d_y}$ (which we identify with $\mathrm{Id}_{\mathcal{Y}} \in \mathcal{L}(\mathcal{Y})$) and $S = I_{\mathbb{R}^{d_u}} \in \mathbb{R}^{d_u \times d_u}$. Let $\mathcal{R} \coloneqq (R \circ \widetilde{\mathcal{Q}})\colon H^t(\mathbb{T}^d; \mathbb{R}^{d_v}) \to \mathcal{Y}$. Define the linear decoder layer $\mathscr{D} \coloneqq (\widetilde{\mathcal{S}} \circ \mathsf{L})\colon \mathbb{R}^{d_u} \to L^2(\mathbb{T}^d; \mathbb{R}^{d_v})$ which has the kernel function product representation

$$z \mapsto \mathscr{D}z = \kappa(\cdot)z\,, \quad \text{where} \quad x \mapsto \kappa(x) \coloneqq \mathbf{1}(x)\widetilde{S} \in \mathbb{R}^{d_v \times d_u}$$

as in (5.2.5). Thus,

$$\begin{aligned}
\psi &= R \circ \widetilde{\mathcal{Q}} \circ F \circ \widetilde{\mathcal{S}} \circ \mathsf{L} \\
&= \mathrm{Id}_{\mathcal{Y}} \circ (R \circ \widetilde{\mathcal{Q}}) \circ F \circ (\widetilde{\mathcal{S}} \circ \mathsf{L}) \circ I_{\mathbb{R}^{d_u}} \\
&= Q \circ \mathcal{R} \circ F \circ \mathscr{D} \circ S
\end{aligned}$$

as claimed. Finally, by the injectivity of L implied by (D.1.3), any $u' \in K$ has the representation $u' = \mathsf{L}z'$ for some unique $z' \in \mathcal{Z} \subset \mathbb{R}^{d_u}$. It follows that

$$\sup_{u\in K} \|\mathcal{G}^\dagger(u) - R\mathcal{G}(u)\|_{\mathcal{Y}} \geq \|\mathcal{G}^\dagger(u') - R\mathcal{G}(u')\|_{\mathcal{Y}} = \|\Psi^\dagger(z') - \Psi(z')\|_{\mathcal{Y}}\,.$$

This implies the asserted result (5.3.2). □

Appendix E

APPENDIX TO CHAPTER 6

E.1 A result on neural network approximation

We here derive a technical result, which shows that ReLU neural networks can approximate C^1 functions in the $W^{1,\infty}$-norm over compact subsets. This result follows from well-known techniques, but we couldn't find a reference. We hence include a statement and proof (sketch), below.

***Lemma* E.1.1.** Let $\sigma(x) = \max(x, 0)$ denote the ReLU activation. Let $D \subset \mathbb{R}^d$ be a bounded Lipschitz domain. For any $f \in C^1(D)$ and $\epsilon > 0$, there exists a shallow ReLU-neural network $\psi : \mathbb{R}^d \to \mathbb{R}$, of the form,

$$\psi(x) = \sum_{j=1}^{N} a_j \sigma(w_j^T x + b_j), \quad \text{with } a_j, b_j \in \mathbb{R},\ w_j \in \mathbb{R}^d,\ \text{for } j = 1, \dots, N, \tag{E.1.1}$$

such that $\|\psi - f\|_{W^{1,\infty}(D)} \leq \epsilon$. ◊

Proof. **Step 1:** Fix a compactly supported smooth function $\rho : \mathbb{R} \to \mathbb{R}$, with $\text{supp}(\rho) \subset (-1, 1)$, $\rho \geq 0$, and such that ρ is even, i.e. $\rho(x) = \rho(-x)$ for all $x \in \mathbb{R}$. Let $\sigma_\rho(x) := (\sigma * \rho)(x)$ denote the convolution. We note that $\sigma_\rho \in C^\infty(\mathbb{R})$ is smooth and monotonically increasing and that σ_ρ is equal to ReLU on $\mathbb{R} \setminus (-1, 1)$, i.e.

$$\sigma_\rho(x) = \begin{cases} 0, & (x \leq -1), \\ x, & (x \geq 1). \end{cases}$$

We first note that for any $\epsilon > 0$, we can find $N \in \mathbb{N}$ and coefficients $\alpha_j, \beta_j, \omega_j \in \mathbb{R}$, for $j = 1, \dots, N$, such that

$$\left\| \sigma_\rho(x) - \textstyle\sum_{j=1}^{N} \alpha_j \sigma(\omega_j x + \beta_j) \right\|_{W^{1,\infty}(\mathbb{R})} \leq \epsilon.$$

To see this, we temporarily fix $M \in \mathbb{N}$, introduce an equidistant partition $x_m := -1 + 2m/M$ of $[-1, 1]$, for $m = 0, \dots, M$, denote $c_m := \sigma_\rho'(x_m) -$

$\sigma'_\rho(x_{m-1})$ and note that

$$\left|\sigma'_\rho(x) - \sum_{m=1}^{M} c_m 1_{[x_m,\infty)}(x)\right| = \begin{cases} |\sigma'_\rho(x)|, & x \in (-\infty, -1), \\ |\sigma'_\rho(x) - \sigma'_\rho(x_{m_0})|, & x \in [x_{m_0}, x_{m_0+1}), \\ |\sigma'_\rho(x) - \sigma'_\rho(x_M)|, & x \in [1, \infty). \end{cases} \tag{E.1.2}$$

Since $\sigma'_\rho(x) \equiv 0$ for $x \leq -1$ and $\sigma'_\rho(x) \equiv 1$ for $x \geq x_M = 1$, it follows that

$$\left|\sigma'_\rho(x) - \sum_{m=1}^{M} c_m 1_{[x_m,\infty)}(x)\right| = 0, \quad \forall\, x \in \mathbb{R} \setminus [-1, 1).$$

On the other hand, we also have

$$\|\sigma''_\rho\|_{L^\infty(\mathbb{R})} = \|\sigma' * \rho'\|_{L^\infty(\mathbb{R})} \leq \|\sigma'\|_{L^\infty(\mathbb{R})} \|\rho'\|_{L^1(\mathbb{R})} \leq \|\rho'\|_{L^1(\mathbb{R})}.$$

For any $x \in [-1, 1)$, we can find $m_0 \in \{0, \ldots, M-1\}$, such that $x \in [x_{m_0}, x_{m_0+1})$, and from (E.1.2), we obtain

$$\begin{aligned} \left|\sigma'_\rho(x) - \sum_{m=1}^{M} c_m 1_{[x_m,\infty)}(x)\right| &\leq |\sigma'_\rho(x) - \sigma'_\rho(x_{m_0})| \\ &\leq \|\sigma''\|_{L^\infty} |x - x_{m_0}| \\ &\leq \frac{2\|\rho'\|_{L^1(\mathbb{R})}}{M}. \end{aligned}$$

Given $\epsilon > 0$, choose M sufficiently large so that $2\|\rho'\|_{L^1(\mathbb{R})}/M \leq \epsilon/2$. Then, noting that $\sigma'(x - x_m) = 1_{[x_m,\infty)}(x)$ pointwise a.e., it follows that

$$\left\|\sigma'_\rho(x) - \sum_{m=1}^{M} c_m \sigma'(x - x_m)\right\|_{L^\infty(\mathbb{R})} \leq \epsilon/2.$$

Taking into account that $\sigma_\rho(x) \equiv 0 \equiv \sum_{m=1}^{M} c_m \sigma(x - x_m)$ for $x < -1$ and $\sigma_\rho(x) \equiv x \equiv \sum_{m=1}^{M} c_m \sigma(x - x_m)$ for $x > 1$, this in turn implies that

$$\left\|\sigma_\rho(x) - \sum_{m=1}^{M} c_m \sigma(x - x_m)\right\|_{L^\infty(\mathbb{R})} \leq \int_{-1}^{1} \left\|\sigma'_\rho(x) - \sum_{m=1}^{M} c_m \sigma'(x - x_m)\right\|_{L^\infty(\mathbb{R})} dx' \leq \epsilon.$$

Since ϵ was arbitrary, we have shown that σ_ρ belongs to the $W^{1,\infty}(\mathbb{R})$-closure of the set of shallow σ-neural networks, in one spatial dimension. In turn, this implies that any shallow σ_ρ-neural network in d dimensions can be approximated by a shallow σ-neural network to any desired accuracy in the $W^{1,\infty}(\mathbb{R}^d)$-norm.

Step 2: Given a compact domain $D \subset \mathbb{R}^d$, it follows from [189, Theorem 4.1] and the fact that σ_ρ is smooth and non-polynomial, that the set of shallow σ_ρ-neural networks is dense in $C^1(D)$. Thus, for any $f \in C^1(D)$ and given $\epsilon > 0$, we can first find a shallow σ_ρ-neural network ψ_ρ, such that

$$\|f - \psi_\rho\|_{W^{1,\infty}(D)} = \|f - \psi_\rho\|_{C^1(D)} \leq \epsilon/2.$$

Second, as a result of Step 1 we can find a σ-neural network ψ, such that

$$\|\psi_\rho - \psi\|_{W^{1,\infty}(D)} \leq \epsilon/2.$$

By the triangle inequality, we conclude that, for this ψ, we have

$$\|f - \psi\|_{W^{1,\infty}(D)} \leq \|f - \psi_\rho\|_{W^{1,\infty}(D)} + \|\psi_\rho - \psi\|_{W^{1,\infty}(D)} \leq \epsilon.$$

□

In the following corollary, we weaken the C^1 requirements for Lemma E.1.1 for chosen inputs.

***Corollary* E.1.2.** Let $D_1 \subset \mathbb{R}^{d_1}$ and $D_2 \subset \mathbb{R}^{d_2}$ be compact domains. Let $f : D_1 \times D_2 \to \mathbb{R}$, $(\eta, x) \mapsto f(\eta, x)$ be a measurable function such that f is C^1 in η and integrable in x, such that

$$\int_{D_2} \|f(\,\cdot\,, x)\|_{C^1(D_1)} \,\mathsf{d}x < \infty.$$

The for any $\epsilon > 0$, there exists a shallow ReLU-neural network $\psi : \mathbb{R}^{d_1+d_2} \to \mathbb{R}$ of the form E.1.1 such that

$$\int_{D_2} \|\psi(\cdot, x) - f(\cdot, x)\|_{W^{1,\infty}(D_1)} \,\mathsf{d}x \leq \epsilon. \tag{E.1.3}$$

◊

Proof. The assumption says that f belongs to $L^1_x(D_2; C^1_\eta(D_1))$. Clearly, we have $C^1(D_1 \times D_2) \subset L^1_x(D_2; C^1_\eta(D_1))$. Upon mollifying $f(\eta, x)$ in the second variable, one checks that $C^1(D_1 \times D_2)$ is in fact dense in $L^1_x(D_2; C^1_\eta(D_1))$. Thus, there exists $f_\epsilon \in C^1(D_1 \times D_2)$, such that

$$\int_{D_2} \|f(\,\cdot\,, x) - f_\epsilon(\,\cdot\,, x)\|_{C^1(D_1)} \,\mathsf{d}x \leq \epsilon/2.$$

From Lemma E.1.1, there exists a shallow ReLU-neural network $\psi : \mathbb{R}^{d_1+d_2} \to \mathbb{R}$ such that

$$\|\psi - f_\epsilon\|_{W^{1,\infty}(D_1 \times D_2)} \leq \epsilon/(2|D_2|).$$

This in turn implies that

$$\int_{D_2} \|\psi(\cdot, x) - f_\epsilon(\cdot, x)\|_{W^{1,\infty}(D_1)} \, \mathsf{d}x \leq \epsilon/2.$$

Combining the above estimates, and using the triangle inequality, we conclude that

$$\int_{D_2} \|\psi(\,\cdot\,, x) - f(\,\cdot\,, x)\|_{W^{1,\infty}(D_1)} \, \mathsf{d}x \leq \epsilon/2 + \epsilon/2 = \epsilon.$$

□

E.2 Proofs for Section 6.3

Proof of Lemma 6.3.9

Our proof of Lemma 6.3.9 will make use of the following version of the contraction mapping theorem.

***Lemma* E.2.1** (Lemma 6.6.6. of [190])**.** Let $B(0,r)$ be a ball in $\mathbb{R}^n$ centered at the origin and let $g : B(0,r) \to \mathbb{R}^n$ be a map such that $g(0) = 0$ and

$$|g(x) - g(y)| \leq \frac{1}{2}|x - y| \text{ for all } x, y \in B(0,r).$$

Then the function $F : B(0,r) \to \mathbb{R}^n$ defined by $F(x) = x + g(x)$ is one-to-one, and the image $F(B(0,r))$ of this map contains the ball $B(0, \frac{r}{2})$. ◊

We now come to the proof of Lemma 6.3.9 in the main text.

Proof of Lemma 6.3.9. Since $V \subset \mathbb{R}^d$ is open, we may pick $r > 0$ and $y_0 \in V$ such that $B(y_0, r) \subset V$. We will show that the claim holds with

$$\epsilon_0 := \min\left(\tfrac{1}{2}, \tfrac{r}{4}\right), \quad V_0 := B(y_0, \tfrac{r}{4}).$$

Our proof relies on the contraction mapping theorem, formulated as Lemma E.2.1. To this end, we first define $\widetilde{F} : B(0,r) \to \mathbb{R}^d$, by

$$\widetilde{F}(y) = F(y + y_0) - F(y_0).$$

We have that $\widetilde{F}(0) = 0$ and, by assumption, the spectral norm of the Jacobian satisfies $\|D\widetilde{F}(y) - I_d\|_2 \leq \|F - \mathrm{id}\|_{W^{1,\infty}(V)} \leq \epsilon_0$ for all $y \in B(0,r)$. Defining

$g = \widetilde{F} - \mathrm{id}$, this implies that $\|Dg(y)\|_2 \leq \epsilon_0$ for all $y \in B(0,r)$. We assume that $0 < \epsilon_0 \leq 1/2$. Since,

$$\begin{aligned} g(y) - g(y') &= \int_0^1 \frac{d}{dt} g(y' + t(y-y'))\, \mathsf{d}t \\ &= \int_0^1 Dg(y' + t(y-y'))\, \mathsf{d}t \cdot (y - y'), \end{aligned}$$

this implies that,

$$\begin{aligned} |g(y) - g(y')| &\leq \int_0^1 \|Dg(y' + t(y-y'))\|_2\, \mathsf{d}t |y - y'| \\ &\leq \frac{1}{2}|y - y'| \end{aligned}$$

for $y, y' \in B(0,r)$ by our assumption on ϵ_0. As a consequence of the contraction mapping theorem (cp. Lemma E.2.1), $\widetilde{f}$ is injective on $B(0,r)$, and $B(0, \frac{r}{2}) \subset \widetilde{F}(B(0,r))$. The next step is to return to $B(y_0, r)$. As

$$F(y + y_0) = \widetilde{F}(y) + F(y_0),$$

and since $F(y_0)$ is a constant shift, F is injective on $B(y_0, r)$, and $B(F(y_0), \frac{r}{2}) \subset F(B(y_0, r))$. The center of ball $B(F(y_0), \frac{r}{2})$ clearly depends on the value of $F(y_0)$. However, we argue that $B(y_0, \frac{r}{4}) \subset B(F(y_0), \frac{r}{2})$: this follows from the fact that, by assumption on F, we have

$$|F(y_0) - y_0| \leq \|F - \mathrm{id}\|_{W^{1,\infty}(V)} \leq \epsilon_0 \leq \frac{r}{4}.$$

Thus, $B(F(y_0), \frac{r}{2})$ contains a ball $B(y_0, r_0)$ of radius $r_0 = \frac{r}{2} - \epsilon_0 \geq \frac{r}{4}$. It follows that

$$B(y_0, \tfrac{r}{4}) \subset B(F(y_0), \tfrac{r}{2}) \subset F(B(y_0, r)).$$

This shows that the image of $F : V \to \mathbb{R}^d$ contains $V_0 := B(y_0, \frac{r}{4})$. We finally verify that there exists a constant $c_0 > 0$ such that

$$F_{\#}\mathrm{Unif}(V) \geq c_0\, \mathrm{Unif}(V_0).$$

To this end, we recall that $\mathrm{Unif}(V) = |V|^{-1}\, \mathbf{1}_V(y)\, \mathsf{d}y$ is just a rescaling of the Lebesgue measure $\mathbf{1}_V(y)\, \mathsf{d}y$. Since $F : F^{-1}(V_0) \to V_0$ is bijective, the push-forward of the Lebesgue measure under F satisfies

$$F_{\#}(\mathbf{1}_V(y)\, \mathsf{d}y) \geq F_{\#}(\mathbf{1}_{F^{-1}(V_0)}(y)\, \mathsf{d}y) = |\mathrm{det} DF(F^{-1}(z))|^{-1} \mathbf{1}_{V_0}(z)\, \mathsf{d}z.$$

The spectral norm bound $\|DF(y) - I_d\|_2 \leq \epsilon_0 \leq \frac{1}{2}$, which holds for almost all $y \in V$, now implies that

$$\left(\frac{1}{2}\right)^d \leq |\det(DF(F^{-1}(z)))| \leq \left(\frac{3}{2}\right)^d, \quad \mathsf{d}z\text{-almost everywhere.}$$

Thus,

$$\begin{aligned} F_{\#}\mathrm{Unif}(V) &= |V|^{-1} F_{\#}(\mathbf{1}_V(y)\, \mathsf{d}y) \\ &\geq |V|^{-1} |\det DF(F^{-1}(z))|^{-1} \mathbf{1}_{V_0}(z)\, \mathsf{d}z \\ &\geq |V|^{-1} \left(\frac{2}{3}\right)^d \mathbf{1}_{V_0}(z)\, \mathsf{d}z \\ &= \frac{|V_0|}{|V|} \left(\frac{2}{3}\right)^d \mathrm{Unif}(V_0). \end{aligned}$$

The claim thus follows with $c_0 := \frac{|V_0|}{|V|}\left(\frac{2}{3}\right)^d > 0$.

□

Proof of Lemma 6.3.7

Proof of Lemma 6.3.7. Before discussing our construction of A, we recall that, by definition, $\mathcal{A} : \mathbf{U} \to L^p(\mu)$ is of the form

$$\mathcal{A}(\Psi) = \mathcal{Q}(\Psi(u_1), \ldots, \Psi(u_N)),$$

where $u_1, \ldots, u_N$ are fixed and $\mathcal{Q} : \mathbb{R}^N \to L^p(\mu)$ is a reconstruction from point-values. Given $\psi \in U_\ell^{\alpha,\infty}$, we now define $A(\psi) \in L^p([0,1]^d)$ by the conditional expectation,

$$A(\psi)(x) := \mathbb{E}_{u\sim\mu}[\mathcal{A}(\psi \circ \mathcal{E})(u) \,|\, \mathcal{E}(u) = x], \quad \forall\, x \in [0,1]^d.$$

This conditional expectation is well-defined for $\mathcal{E}_{\#}\mu$ almost every x. By assumption (6.3.7), we have $\mathcal{E}_{\#}\mu \geq c \cdot \mathrm{Unif}([0,1]^d)$, and hence $A(\psi)(x)$ is well-defined for (Lebesgue-) almost every $x \in [0,1]^d$.

We also note that $A(\psi)(x)$ is of the form $A(\psi) = Q(\psi(x_1), \ldots, \psi(x_N))$: indeed, by definition, we have $\mathcal{A}(\psi \circ \mathcal{E})(u) = \mathcal{Q}(\psi(\mathcal{E}(u_1)), \ldots, \psi(\mathcal{E}(u_N)))(u)$. Hence, upon defining $x_j := \mathcal{E}(u_j) \in \mathbb{R}^N$, and

$$Q(y_1, \ldots, y_N)(x) := \mathbb{E}_{u\sim\mu}[\mathcal{Q}(y_1, \ldots, y_N)(u) \,|\, \mathcal{E}(u) = x],$$

we then have $A(\psi) = Q(\psi(x_1), \ldots, \psi(x_n))$ for all $\psi \in U_\ell^{\alpha,\infty}$.

To simplify notation for the following calculations, we define $\Psi_{\mathcal{A}} := \mathcal{A}(\Psi)$ with $\Psi := \psi \circ \mathcal{E}$. We can then write

$$A(\psi)(x) = \mathbb{E}_u[\Psi_{\mathcal{A}}(u) \,|\, \mathcal{E}(u) = x].$$

Here $\mathbb{E}_u[\dots \,|\, \mathcal{E}(u) = x]$ is the conditional expectation over a random variable $u \sim \mu$, with conditioning on $\mathcal{E}(u) = x$.

For $p \in [1, \infty)$, we then have

$$|A(\psi)(x)|^p = \left|\mathbb{E}_u[\Psi_{\mathcal{A}}(u) \,|\, \mathcal{E}(u) = x]\right|^p \leq \mathbb{E}_u[|\Psi_{\mathcal{A}}(u)|^p \,|\, \mathcal{E}(u) = x], \tag{E.2.1}$$

by conditional Jensen's inequality. It follows that

$$\begin{aligned}
\int_{[0,1]^d} |A(\psi)(x)|^p \, dx &\leq c^{-1} \int_{\mathbb{R}^d} |A(\psi)(x)|^p \, \mathcal{E}_{\#}\mu(dx) \\
&= c^{-1} \, \mathbb{E}_{x \sim \mathcal{E}_{\#}\mu}\Big[|A(\psi)(x)|^p\Big] \\
&\leq c^{-1} \, \mathbb{E}_{x \sim \mathcal{E}_{\#}\mu}\Big[\mathbb{E}_u\Big[|\Psi_{\mathcal{A}}(u)|^p \,\Big|\, \mathcal{E}(u) = x\Big]\Big] \\
&= c^{-1} \, \mathbb{E}_{u \sim \mu}\Big[|\Psi_{\mathcal{A}}(u)|^p\Big].
\end{aligned}$$

The first inequality is by assumption $\mathcal{E}_{\#}\mu \geq c \cdot \mathrm{Unif}([0,1]^d)$, the second inequality on the third row is (E.2.1) above. The final equality follows from basic properties of the conditional expectation. Thus, recalling that $\Psi_{\mathcal{A}} = \mathcal{A}(\Psi) \in L^p(\mu)$, it follows that

$$\int_{[0,1]^d} |A(\psi)(x)|^p \, dx \leq c^{-1} \, \|\mathcal{A}(\Psi)\|^p_{L^p(\mu)} < \infty.$$

This shows that $A : U^{\alpha,\infty}_{\ell} \to L^p([0,1]^d)$ is well-defined.

It remains to show that A satisfies the claimed lower bound (6.3.10). To see this, we once more apply conditional Jensen's inequality, to obtain

$$\begin{aligned}
\|\Psi - \mathcal{A}(\Psi)\|^p_{L^p(\mu)} &= \|\Psi - \Psi_{\mathcal{A}}\|^p_{L^p(\mu)} \\
&= \mathbb{E}_{u \sim \mu}[|\psi(\mathcal{E}(u)) - \Psi_{\mathcal{A}}(u)|^p] \\
&= \mathbb{E}_{x \sim \mathcal{E}_{\#}\mu} \mathbb{E}_u\Big[\Big|\psi(\mathcal{E}(u)) - \Psi_{\mathcal{A}}(u)\Big|^p \,|\, \mathcal{E}(u) = x\Big] \\
&\geq \mathbb{E}_{x \sim \mathcal{E}_{\#}\mu}\Big[\Big|\psi(x) - \mathbb{E}_u[\Psi_{\mathcal{A}}(u) \,|\, \mathcal{E}(u) = x]\Big|^p\Big] \\
&= \mathbb{E}_{x \sim \mathcal{E}_{\#}\mu}[|\psi(x) - A(\psi)(x)|^p] \\
&= \|\psi - A(\psi)\|^p_{L^p(\mathcal{E}_{\#}\mu)}.
\end{aligned}$$

Recalling (6.3.7), this implies that

$$\|\Psi - \mathcal{A}(\Psi)\|^p_{L^p(\mu)} \geq c\|\psi - A(\psi)\|^p_{L^p([0,1]^d)}.$$

Since $\psi \in U^{\alpha,\infty}_\ell$ was arbitrary and $\Psi = \psi \circ \mathcal{E}$, the proof of (6.3.10) is complete. □

Proof of Lemma 6.3.10

Proof of Lemma 6.3.10. By assumption on $F_\xi = F(\,\cdot\,;\xi)$, we have

$$\|F_\xi - \mathrm{id}\|_{W^{1,\infty}(V)} \leq \epsilon_0, \quad \forall\, \xi \in K.$$

By Lemma 6.3.9, there exists $V_0 \subset V$ and a constant $c_0 > 0$, such that

$$(F_\xi)_\# \mathrm{Unif}(V) \geq c_0\, \mathrm{Unif}(V_0), \quad \forall\, \xi \in K. \tag{E.2.2}$$

We want to show that the push-forward under F of the product measure $\mathrm{Unif}(V) \otimes \mathbb{P} \in \mathcal{P}(V \times \Omega)$ satisfies

$$F_\#(\mathrm{Unif}(V) \otimes \mathbb{P}) \geq c_0 \mathbb{P}(K)\, \mathrm{Unif}(V_0).$$

Given a non-negative, bounded measurable function $\phi : \mathbb{R}^d \to [0,\infty)$, we have

$$\begin{aligned}\mathbb{E}_{z\sim F_\#(\mathrm{Unif}(V)\otimes\mathbb{P})}[\phi(z)] &= \mathbb{E}_{(y,\xi)\sim \mathrm{Unif}(V)\otimes\mathbb{P}}[\phi(F(y;\xi))] \\ &= \mathbb{E}_{\xi\sim\mathbb{P}}\big[\mathbb{E}_{y\sim\mathrm{Unif}(V)}[\phi(F_\xi(y))]\big]\end{aligned}$$

By (E.2.2), we have

$$\mathbb{E}_{y\sim\mathrm{Unif}(V)}[\phi(F_\xi(y))] \geq c_0\, \mathbb{E}_{z\sim\mathrm{Unif}(V_0)}[\phi(z)], \quad \forall\, \xi \in K,$$

and hence

$$\begin{aligned}\mathbb{E}_{z\sim F_\#(\mathrm{Unif}(V)\otimes\mathbb{P})}[\phi(z)] &\geq \mathbb{E}_{\xi\sim\mathbb{P}}\big[\mathbf{1}_K(\xi)\, \mathbb{E}_{y\sim\mathrm{Unif}(V)}[\phi(F_\xi(y))]\big] \\ &\geq c_0\, \mathbb{E}_{\xi\sim\mathbb{P}}\big[\mathbf{1}_K(\xi)\, \mathbb{E}_{z\sim\mathrm{Unif}(V_0)}[\phi(z)]\big] \\ &= c_0 \mathbb{P}(K)\, \mathbb{E}_{z\sim\mathrm{Unif}(V_0)}[\phi(z)].\end{aligned}$$

Since $\phi \geq 0$ was arbitrary, the claim follows. □

Proof of Proposition 6.3.11

Proof of Proposition 6.3.11. Let $I_1, \ldots, I_d$ denote the open intervals in Assumption 6.3.1. Define $V := I_1 \times \cdots \times I_d \subset \mathbb{R}^d$. Instead of proving (6.3.16)

directly, we will consider the simplified encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^d$, with components of the form

$$\mathcal{E}_j(u) = \sum_{k=1}^{d_0} c_{jk} \ell_k(u), \tag{E.2.3}$$

and where the coefficient c_{jk} are chosen to ensure (6.3.14) for a $\delta > 0$ to be determined. Our goal is to show that there exists an open set $V_0 \subset V$ and constant $c > 0$, such that

$$\mathcal{E}_{\#}\mu \geq c \, \mathrm{Unif}(V_0). \tag{E.2.4}$$

The general claim (6.3.16) then follows by introducing a scaling factor $\gamma > 0$ and bias $b \in \mathbb{R}^d$, such that

$$[0,1]^d \subset \gamma \cdot V_0 + b,$$

and replacing the simple encoder $\mathcal{E}$ (E.2.3) by

$$\widetilde{\mathcal{E}}_j(u) := b_j + \sum_{k=1}^{d_0} a_{jk} \ell_k(u),$$

where $a_{jk} := \gamma \, c_{jk}$. Thus, it only remains to show (E.2.4).

Fix $\delta > 0$ for the moment. We will determine conditions on δ which imply that (E.2.4) holds for the encoder (E.2.3), where c_{jk} are chosen according to (6.3.14). With this specific choice of c_{jk}, we can then write (6.3.14) in the form

$$\left\| e_j^* - \mathcal{E}_j \right\|_{\mathcal{X}^*} \leq \delta, \quad \forall \, j = 1, \ldots, d. \tag{E.2.5}$$

Our goal is to apply Lemma 6.3.10. To this end, we recall the decomposition $u = u(y;\xi) = \xi + \sum_{j=1}^d y_j e_j$ of (6.3.2) and the probability measures $\mu_d \in \mathcal{P}(\mathbb{R}^d)$, $\mu_d^\perp \in \mathcal{P}(\Omega_d)$ of (6.3.4), (6.3.5). Given this decomposition, we let $F : V \times \Omega_d \to \mathbb{R}^d$ be defined by $F(y;\xi) := \mathcal{E}(u(y;\xi))$, and denote $F_\xi(y) := F(y;\xi)$. We will apply Lemma 6.3.10 with this choice of F_ξ and with $\mathbb{P} := \mu_d^\perp$. As our final ingredient, we recall from Lemma 6.3.2 that there exists a set $B > 0$, such that $\mathbb{P}(K) > 0$ for $K := \{\xi \in \Omega_d \,|\, \|\xi\|_{\mathcal{X}} \leq B\}$. Given these preparatory remarks, our goal now is to show that (6.3.13) holds, provided that $\delta > 0$ in (6.3.14) is sufficiently small.

To this end, we first note that $(y,\xi) \mapsto F(y;\xi)$ is linear, and hence $y \mapsto F_\xi(y)$ is affine for fixed ξ, and

$$F_\xi(y) = F(y;\xi) = F(y;0) + F(0,\xi) = F_0(y) + F_\xi(0).$$

Thus, we can write

$$\|F_\xi(y) - y\|_{W^{1,\infty}(V)} \leq \|F_0(y) - y\|_{W^{1,\infty}(V)} + \|F_\xi(0)\|_{W^{1,\infty}(V)}. \tag{E.2.6}$$

We will bound both terms on the right, individually.

The last term is constant in y, and hence $\|F_\xi(0)\|_{W^{1,\infty}(V)} = |F_\xi(0)|$. By (E.2.3), the j-th component of $F_\xi(0) = \mathcal{E}(u(0;\xi)) = \mathcal{E}(\xi)$ is given by

$$\mathcal{E}_j(\xi) := \sum_{k=1}^{d_0} c_{jk}\ell_k(\xi).$$

Since $j \leq d$, it follows that $e_j^*(\xi) = 0$. From (E.2.5), we conclude that

$$\begin{aligned} |\mathcal{E}_j(\xi)| &= \left|\mathcal{E}_j(\xi) - e_j^*(\xi)\right| \\ &\leq \left\|\mathcal{E}_j - e_j^*\right\|_{\mathcal{X}^*}\|\xi\|_{\mathcal{X}} \\ &\leq \delta\|\xi\|_{\mathcal{X}}. \end{aligned}$$

For $\xi \in K = \{\xi \,|\, \|\xi\|_{\mathcal{X}} \leq B\}$, then it follows that

$$|F_\xi(0)| \leq d \max_{j\in[d]} |\mathcal{E}_j(\xi)| \leq dB\,\delta.$$

Thus, for $\delta \leq \epsilon_0/(3dB)$, we obtain

$$\|F_\xi(0)\|_{W^{1,\infty}(V)} \equiv |F_\xi(0)| \leq \epsilon_0/3, \quad \forall\, \xi \in K. \tag{E.2.7}$$

This provides our estimate for the second term in (E.2.6). To bound the first term, we note that $F_0(y) = \mathcal{E}(u(y;0))$, and hence

$$\begin{aligned} |F_0(y)_j - y_j| &= |\mathcal{E}_j(u(y;0)) - e_j^*(u(y;0))| \\ &\leq \|\mathcal{E}_j - e_j^*\|_{\mathcal{X}^*}\|u(y;0)\|_{\mathcal{X}}. \end{aligned}$$

Since $y \in V$ is from a bounded set, we can assume without loss of generality that $B > 0$ is chosen sufficiently large such that $\|u(y;0)\|_{\mathcal{X}} \leq B$ for all $y \in V$, and hence, we obtain,

$$\|F_0(y) - y\|_{L^\infty(V)} \leq \epsilon_0/3, \tag{E.2.8}$$

whenever $\delta \leq \epsilon_0/(3dB)$. It remains to derive a similar bound on

$$\|D_y F_0(y) - D_y y\|_{L^\infty(V)} = \|D_y F_0(y) - I\|_{L^\infty(V)}.$$

To this end, we recall that $y \mapsto F_0(y)$ is linear, and hence is represented by a matrix $A \in \mathbb{R}^{d\times d}$, i.e. $F_0(y) = Ay$. It follows that

$$\|D_y F_0(y) - D_y y\|_{L^\infty(V)} = \|A - I\|_2,$$

where $\|\cdot\|_2$ is the operator norm. Retracing the argument above, it follows that any $y \in \mathbb{R}^d$, we have

$$\begin{aligned} |Ay - y|_{\ell^2} &\le d \max_{j\in[d]} |\mathcal{E}_j(u(y;0)) - e_j^*(u(y;0))| \\ &\le d\|\mathcal{E}_j - e_j^*\|_{\mathcal{X}^*}\|u(y;0)\|_{\mathcal{X}} \\ &\le d\delta\|u(y;0)\|_{\mathcal{X}}. \end{aligned}$$

We can furthermore find a constant $C > 0$, depending only on d and $e_1, \dots, e_d$, such that

$$\|u(y;0)\|_{\mathcal{X}} \le C|y|_{\ell^2}, \quad \forall\, y \in \mathbb{R}^d.$$

Thus,

$$|Ay - y|_{\ell^2} \le dC\delta\, |y|_{\ell^2},$$

which, upon taking the supremum over all $|y|_{\ell^2} = 1$, implies that

$$\|A - I\|_{op} \le (dC)\,\delta.$$

Hence, for $\delta \le \min(\epsilon_0/(3dC), \epsilon_0/(3dB))$, we conclude that

$$\|D_y F_0(y) - D_y y\|_{L^\infty(V)} \le \epsilon_0/3,$$

and by (E.2.8),

$$\begin{aligned} \|F_0(y) - y\|_{W^{1,\infty}(V)} &= \|D_y F_0(y) - D_y y\|_{L^\infty(V)} + \|F_0(y) - y\|_{L^\infty(V)} \\ &\le 2\epsilon_0/3. \end{aligned}$$

Combining the last estimate, (E.2.7) and (E.2.6), we conclude that

$$\|F_\xi(y) - y\|_{W^{1,\infty}(V)} \le \epsilon_0, \quad \forall\, \xi \in K.$$

Thus, by Lemma 6.3.10, there exists $V_0 \subset V$ and $c_0 > 0$, such that

$$\begin{aligned} \mathcal{E}_{\#}\mu &= F_{\#}(\mu_d \otimes \mathbb{P}) \\ &\ge c_\rho F_{\#}(\mathrm{Unif}(V) \otimes \mathbb{P}) \\ &\ge \underbrace{c_\rho c_0 \mathbb{P}(K)}_{=:c}\, \mathrm{Unif}(V_0). \end{aligned}$$

□

Proof of Lemma 6.3.13

Proof of Lemma 6.3.13. By the assumption that $f \in U_\ell^{\alpha,\infty}(D)$ it follows that for each $n \in \mathbb{N}$ there is $\psi_n \in \Sigma_n^\ell$ with

$$\|f - R_\sigma(\psi_n)\|_{L^\infty(D)} \leq n^{-\alpha}. \tag{E.2.9}$$

Recall that, by definition, $\psi_n = ((A_1, b_1), \dots, (A_L, b_l)) \in \bigtimes_{l=1}^{L}(\mathbb{R}^{d_l \times d_{l-1}} \times \mathbb{R}^{d_l})$ with some $(d_0, d_1, \dots, d_L) \in \mathbb{N}^{L+1}$, where $d_0 = d$, $L \leq \ell(n)$, $W(\psi_n) \leq n$ and $\|\psi_n\|_{\mathsf{NN}} \leq 1$.

Consider the functions

$$h_n : \begin{cases} \mathbb{R}^e \to \mathbb{R}, \\ \quad x \mapsto R_\sigma(\psi_n)(Cx + b), \end{cases} \tag{E.2.10}$$

which clearly satisfy that

$$h_n = R_\sigma(\varphi_n), \tag{E.2.11}$$

where

$$\varphi_n = \left(\left(\widetilde{A}_l, \widetilde{b}_l\right)\right)_{l=1}^{L} \in \left(\mathbb{R}^{e \times d_1} \times \mathbb{R}^{d_1}\right) \times \left(\bigtimes_{l=2}^{L}\left(\mathbb{R}^{d_l \times d_{l-1}} \times \mathbb{R}^{d_l}\right)\right),$$

and

$$\widetilde{A}_1 = A_1 C,\ \widetilde{b}_1 = A_1 b + b_1 \quad \text{and} \quad \widetilde{A}_l = A_l,\ \widetilde{b}_l = b_l \quad \text{for } l = 2, \dots, L.$$

This, and the fact that $W(\psi_n) \leq n$ and $\|\psi_n\|_{\mathsf{NN}} \leq 1$ readily yields that

$$W(\varphi_n) \leq W(\psi_n) \cdot (\|C\|_{\ell^0} + \|b\|_{\ell^0}) = n \cdot (\|C\|_{\ell^0} + \|b\|_{\ell^0}). \tag{E.2.12}$$

and

$$\|\varphi_n\|_{\mathsf{NN}} \leq \max\left(\{1\} \cup \bigcup_{k=1}^{e}\left\{\sum_{j=1}^{d}|C_{j,k}|\right\} \cup \left\{1 + \sum_{j=1}^{d}|b_j|\right\}\right) =: T. \tag{E.2.13}$$

Note that T is independent of n. Now pick $R \geq T$ to be determined later and denote

$$\tau_n := \left(\left(\frac{1}{R}\widetilde{A}_1, \frac{1}{R}\widetilde{b}_1\right), \left(A_2, \frac{1}{R}b_2\right), \dots, \left(A_L, \frac{1}{R}b_L\right)\right).$$

By the homogeinity of the ReLU activation function σ it holds that

$$R_\sigma(\tau_n) = \frac{1}{R}h_n. \tag{E.2.14}$$

Moreover, due to (E.2.13), the fact that $R \geq \max\{1, T\}$ and (E.2.12) it holds that

$$\|\tau_n\|_{\mathsf{NN}} \leq 1 \quad \text{and} \quad W(\tau_n) \leq n \cdot (\|C\|_{\ell^0} + \|b\|_{\ell^0}), \tag{E.2.15}$$

which implies that

$$\tau_n \in \Sigma^{\ell(\cdot/(\|C\|_{\ell^0}+\|b\|_{\ell^0}))}_{n\cdot(\|C\|_{\ell^0}+\|b\|_{\ell^0})} \subset \Sigma^{\ell}_{n\cdot(\|C\|_{\ell^0}+\|b\|_{\ell^0})}, \tag{E.2.16}$$

where the last inclusion follows from the fact that ℓ is non-decreasing.

Moreover, by (E.2.9), (E.2.10), (E.2.14) and the definition of E it holds that

$$\left\| \frac{1}{R} f(C \cdot + b) - R_\sigma(\tau_n)(\cdot) \right\|_{L^\infty(E)} \leq \frac{1}{R} \cdot n^{-\alpha} \tag{E.2.17}$$

for all $n \in \mathbb{N}$.

Let $m \in \mathbb{N}$ be arbitrary and define $\mu_m := \tau_{\lfloor m/(\|C\|_{\ell^0}+\|b\|_{\ell^0})\rfloor}$. Equations (E.2.16) and (E.2.17) now readily yield that $\mu_m \in \Sigma^\ell_m$ and

$$\left\| \frac{1}{R} f(C \cdot + b) - R_\sigma(\mu_m)(\cdot) \right\|_{L^\infty(E)} \leq \frac{2^\alpha \cdot (\|C\|_{\ell^0} + \|b\|_{\ell^0})^\alpha}{R} m^{-\alpha}.$$

By choosing R sufficiently large this implies that

$$d_{L^\infty(E)}\left(\frac{1}{R} f(C \cdot + b), \Sigma^\ell_m \right) \leq m^{-\alpha} \quad \text{for all} \quad m \in \mathbb{N},$$

which implies that $f(C \cdot + b) \in R \cdot U^{\alpha,\infty}_\ell(E)$, as claimed. □

Proof of Lemma 6.3.21

Proof. By construction, the encoder $\mathcal{E}$ is of the form

$$\mathcal{E}(u) = \fint_D R(u(x), x)\, dx,$$

where $R : \mathbb{R} \times \mathbb{R}^{d_D} \to \mathbb{R}^d$ is a shallow neural network. Let $\psi = ((A_1, b_1), \ldots, (A_L, b_L))$ be a neural network with $L \leq \ell_0$ layers. We define a shallow neural network,

$$\widetilde{R}(\eta, x) := A_1 R(\eta, x) + b_1,$$

and $\widetilde{\mathcal{R}}(u)(x) := \widetilde{R}(u(x), x)$. We next define the first hidden ANO layer as,

$$\mathcal{L}_1(v)(x) = \sigma\left(\fint_D v(x)\, dx \right),$$

i.e. a hidden layer with weight matrix and bias $W_1 = 0$, $b_1 = 0$. For $j = 2, \dots, L-2$, we define

$$\mathcal{L}_j(v) = \sigma\bigg((A_j - I)v(x) + b_j + \fint_D v(x)\,dx\bigg),$$

where I denotes the unit matrix and finally,

$$Q(v) = A_L\sigma(A_{L-1}v + b_{L-1}) + b_L.$$

Let $\psi_1(\xi) = \sigma(A_1\xi + b_1)$ denote the first layer of ψ. Then we have

$$\psi_1(\mathcal{E}(u)) = \sigma\bigg(A_1 \fint_D R(u(x), x)\,dx + b_1\bigg) = \sigma\bigg(\fint_D \widetilde{R}(u(x), x)\,dx\bigg) = \mathcal{L}_1(\widetilde{\mathcal{R}}(u)).$$

Since the output $v(x) := \psi_1(\mathcal{E}(u)) = \mathcal{L}_1(\widetilde{\mathcal{R}}(u))$ is a constant function, it follows that

$$\mathcal{L}_2(v) = \sigma\bigg((A_2 - I)v(x) + b_2 + \fint_D v(x)\,dx\bigg) = \sigma(A_2v + b_2).$$

Thus, $\mathcal{L}_2(\mathcal{L}_1(\widetilde{\mathcal{R}}(u))) = \sigma(A_2\psi_1(\mathcal{E}(u)) + b_2)$ agrees with the output of the second hidden layer of ψ. Continuing recursively, it follows that

$$\Psi(u) := \mathcal{Q} \circ \mathcal{L}_{L-2} \circ \cdots \circ \mathcal{L}_1 \circ \widetilde{\mathcal{R}}(u) = \psi \circ \mathcal{E}(u), \quad \forall\, u \in \mathcal{X}(D).$$

Thus, $\psi \circ \mathcal{E}$ is equal to an ANO of depth $L-2 \leq \ell_0 - 2$, with input layer $\widetilde{\mathcal{R}}$ and output layer $\mathcal{Q}$. Employing a rescaling argument similar to the proof of Lemma 6.3.13, relying on the homogeneity of ReLU as well as the fact that the total number of layers is bounded by $\ell_0 < \infty$, it follows that for sufficiently large $\gamma > 0$, depending only on $\mathcal{E}$, ℓ_0 and α, we have $\frac{1}{\gamma}\Psi \in \mathbf{U}^{\alpha}_{\ell,\mathrm{NO}}$, i.e. $\Psi \in \gamma \cdot \mathbf{U}^{\alpha}_{\ell,\mathrm{NO}}$. Here we have also made use of the fact that $\ell_0 - 2 \leq \ell^*$. □