IMA Journal of Numerical Analysis (2025) **00**, 1–66
https://doi.org/10.1093/imanum/draf028

The parametric complexity of operator learning

Samuel Lanthaler* and Andrew M. Stuart
California Institute of Technology, 1200 E California Blvd, Pasadena, CA 91125, USA
*Corresponding author: slanth@caltech.edu

[Received on 28 February 2024; revised on 21 October 2024]

Neural operator architectures employ neural networks to approximate operators mapping between Banach spaces of functions; they may be used to accelerate model evaluations via emulation, or to discover models from data. Consequently, the methodology has received increasing attention over recent years, giving rise to the rapidly growing field of operator learning. The first contribution of this paper is to prove that for general classes of operators, which are characterized only by their C^r- or Lipschitz-regularity, operator learning suffers from a 'curse of parametric complexity', which is an infinite-dimensional analogue of the well-known curse of dimensionality encountered in high-dimensional approximation problems. The result is applicable to a wide variety of existing neural operators, including PCA-Net, DeepONet and the Fourier neural operator. The second contribution of the paper is to prove that this general curse can be overcome for solution operators defined by the Hamilton–Jacobi (HJ) equation; this is achieved by leveraging additional structure in the underlying solution operator, going beyond regularity. To this end a novel neural operator architecture is introduced, termed HJ-Net, which explicitly takes into account characteristic information of the underlying Hamiltonian system. Error and complexity estimates are derived for HJ-Net, which show that this architecture can provably beat the curse of parametric complexity related to the infinite-dimensional input and output function spaces.

Keywords: operator learning; complexity; Hamilton–Jacobi; deep learning.

1. Introduction

This paper is devoted to a study of the computational complexity of the approximation of maps between Banach spaces by means of neural operators. The paper has two main focii: establishing a complexity barrier for general classes of C^r- or Lipschitz regular maps; and then showing that this barrier can be beaten for Hamilton–Jacobi (HJ) equations. In Subsection 1.1 we give a detailed literature review; we set in context the definition of *'the curse of parametric complexity'* that we introduce and use in this paper; and we highlight our main contributions. Then, in Subsection 1.2, we overview the organization of the remainder of the paper.

1.1 *Context and literature review*

The use of neural networks to learn operators, typically mapping between Banach spaces of functions defined over subsets of finite-dimensional Euclidean space and referred to as *neural operators*, is receiving growing interest in the computational science and engineering community (Chen & Chen, 1995; Zhu & Zabaras, 2018; Bhattacharya *et al.*, 2021; Khoo *et al.*, 2021; Li *et al.*, 2021; Nelsen & Stuart, 2021; Lu *et al.*, 2021b; Boullé *et al.*, 2022). The methodology has the potential for *accelerating* numerical methods for solving partial differential equations (PDEs) when a model relating inputs and outputs is known; and it has the potential for *discovering* input–output maps from data when no model is available.

The computational complexity of learning and evaluating such neural operators is crucial to understanding when the methods will be effective. Numerical experiments addressing this issue may be found in de Hoop *et al.* (2022) and the analysis of linear problems from this perspective may be found in Boullé & Townsend (2022); de Hoop *et al.* (2023). Universal approximation theorems, applicable beyond the linear setting, may be found in Chen & Chen (1995); Bhattacharya *et al.* (2021); Kovachki *et al.* (2021); Lu *et al.* (2021b); Lanthaler *et al.* (2022); You *et al.* (2022); Kovachki *et al.* (2023); Lanthaler *et al.* (2023a), but such theorems do not address the cost of achieving a given small error.

Early work on operator approximation (Mhaskar & Hahm, 1997) presents first quantitative bounds; most notably, this work identifies the continuous nonlinear n-widths of a space of continuous functionals defined on L^2-spaces, showing that these n-widths decay at most (poly-)logarithmically in n. Both upper and lower bounds are derived in this specific setting. More recently, upper bounds on the computational complexity of recent approaches to operator learning based on deep neural networks, including the DeepONet (Lu *et al.*, 2021b) and the Fourier neural operator (FNO) (Li *et al.*, 2021), have been studied in more detail. Specific operator learning tasks arising in PDEs have been considered in the papers Schwab & Zech (2019); Kovachki *et al.* (2021); Deng *et al.* (2022); Herrmann *et al.* (2024); Lanthaler *et al.* (2022); Ryck & Mishra (2022); Marcati & Schwab (2023). Related complexity analysis for the PCA-Net architecture (Bhattacharya *et al.*, 2021) has recently been established in Lanthaler (2023). These papers studying computational complexity focus on the issue of beating a form of the *'curse of dimensionality (CoD)'* in these operator approximation tasks.

In these operator learning problems the input and output spaces are infinite-dimensional, and hence the meaning of the CoD could be ambiguous. In this infinite-dimensional context 'beating the curse' is interpreted as identifying problems, and operator approximations applied to those problems, for which a measure of evaluation cost (referred to as their complexity) grows only algebraically with the inverse of the desired error. As shown rigorously in Lanthaler (2023) this is a nontrivial issue: for the PCA-Net architecture it has been established that such algebraic complexity and error bounds *cannot* be achieved for general Lipschitz (and even more regular) operators.

As will be explained in detail in this work, this fact is not specific to PCA-Net, but extends to many other neural operator architectures. In fact, it can be interpreted as a scaling limit of the conventional CoD; this conventional curse affects finite-dimensional approximation problems when the underlying dimension d is very large. It can be shown that (ReLU) neural networks cannot overcome this curse, in general. As a consequence, neural operators, which build on neural networks, suffer from the scaling limit of this curse in infinite dimensions. To distinguish this infinite-dimensional phenomenon from the conventional CoD encountered in high-but-finite-dimensional approximation problems we will refer to the scaling limit identified in this work as *'the curse of parametric complexity'*.

The first contribution of this paper is to prove that for general classes of operators, which are characterized only by their C^r- or Lipschitz-regularity, operator learning suffers from such a curse of parametric complexity: Theorem 2.11 (and a variant thereof, Theorem 2.27) shows that, in this setting, there exist operators (and indeed even real-valued functionals) that are $\epsilon-$approximable only with parametric model complexity that grows exponentially in ϵ^{-1}.

To overcome the general curse of parametric complexity implied by Theorem 2.11 (and Theorem 2.27) efficient operator learning frameworks therefore have to leverage additional structure present in the operators of interest, going beyond C^r- or Lipschitz-regularity. Previous work on overcoming this curse for operator learning has mostly focused on operator holomorphy (Schwab & Zech, 2019; Herrmann *et al.*, 2020; Lanthaler *et al.*, 2022) and the emulation of numerical methods (Kovachki *et al.*, 2021; Lanthaler *et al.*, 2022; Lanthaler, 2023; Marcati & Schwab, 2023) as two basic mechanisms for overcoming the curse of parametric complexity for specific operators of interest. A notable exception

are the complexity estimates for DeepONets in (Deng *et al.*, 2022). These estimates are based on explicit representation of the solution; most prominently, this is achieved via the Cole–Hopf transformation for the viscous Burgers equation. Another distinct mechanism to overcome the curse of parametric complexity is highlighted by the notion of Barron spaces, e.g., Barron (1993); Bach (2017); Ma *et al.* (2022), which have recently been extended to the operator learning setting in Korolev (2022). In the finite-dimensional context functions belonging to such Barron spaces can be approximated at Monte-Carlo rates, which are independent of the dimension of the underlying domain. As shown in Korolev (2022) similar rates can be established in an infinite-dimensional Barron class setting. The curse of parametric complexity derived in this work indicates that spaces of r-times Fréchet differentiable operators cannot embed into operator Barron spaces, no matter the degree of differentiability r.

An abstract characterization of the entire class of operators that allow for efficient approximation by neural operators would be very desirable. Unfortunately, this appears to be out of reach, at the current state of analysis. Indeed, as far as the authors are aware, there does not even exist such a characterization for any class of standard numerical methods, such as finite difference, finite element or spectral, viewed as operator approximators. Therefore, in order to identify settings in which operator learning can be effective (without suffering from the general curse of parametric complexity) we restrict attention to specific classes of operators of interest.

The HJ equations present an application area that has the potential to be significantly impacted by the use of ideas from neural networks, especially regarding the solution of problems for functions defined over subsets of high-dimensional (d) Euclidean space (Darbon & Osher, 2016; Chow *et al.*, 2017, 2019; Darbon *et al.*, 2020); in particular beating the CoD with respect to this dimension d has been the focus. We highlight works that propose adapted neural network architectures, reflecting known structure of viscosity solutions of the HJ equation, to approximate and represent *individual solutions* (Darbon & Meng, 2021; Meng *et al.*, 2022). This research includes studies of analytical convergence rates for such architectures, as well as empirical works studying their practical performance (Nakamura-Zimmerer *et al.*, 2020; Huré *et al.*, 2021; Darbon *et al.*, 2023). However, this work does not include operator learning, as it concerns settings in which only fixed instances of the PDE are solved. The purpose of the second part of this paper is to study the design and analysis of neural operators to approximate the solution operator for HJ equations; this operator maps the initial condition (a function) to the solution at a later time (another function).

The second contribution of the paper is to prove in Theorem 5.1 that the general curse of parametric complexity can be overcome for maps defined by the solution operator of the HJ equation; this is achieved by exposing additional structure in the underlying solution operator, different from holomorphy and emulation and going beyond regularity, that can be leveraged by neural operators; for the HJ equations, the identified structure relies on representation of solutions of the HJ equations in terms of characteristics. In this paper the dimension d of the underlying spatial domain will be fixed and we do not study the CoD with respect to d. Instead, we demonstrate that it is possible to beat the curse of parametric complexity with respect to the infinite-dimensional nature of the input function for fixed (and moderate) d.

1.2 *Organization*

In Section 2 we present the first contribution: Theorem 2.11, together with the closely related Theorem 2.27, which extends the general, but not exhaustive setting Theorem 2.11, to include the FNO, establishes that the curse of parametric complexity is to be expected in operator learning. The following sections then focus on the second contribution, and hence on solution operators associated with the HJ equation; in Theorem 5.1 we prove that additional structure in the solution operator for this equation can be leveraged

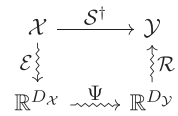

FIG. 1. Diagrammatic illustration of operator learning based on an encoding $\mathcal{E}$, a neural network Ψ and a reconstruction $\mathcal{R}$.

to beat the curse of parametric complexity. In Section 3 we describe the precise setting for the HJ equation employed throughout this paper; we recall the method of characteristics for solution of the equations; and we describe a short-time existence theorem. Section 4 introduces the proposed neural operator, the HJ-Net, based on learning the flow underlying the method of characteristics and combining it with scattered data approximation. In Section 5 we state our approximation theorem for the proposed HJ-Net, resulting in complexity estimates that demonstrate that the curse of parametric complexity is avoided in relation to the infinite-dimensional nature of the input space (of initial conditions). Section 6 contains concluding remarks. While the high-level structure of the proofs is contained in the main body of the paper, many technical details are collected in the appendix, to promote readability.

2. The curse of parametric complexity

Operator learning seeks to employ neural networks to efficiently approximate operators mapping between infinite-dimensional Banach spaces of functions. To enable implementation of these methods in practice maps between the formally infinite-dimensional spaces have to be approximated using only a finite number of degrees of freedom.

Commonly, operator learning frameworks can therefore be written in terms of an encoding, a neural network and a reconstruction step as shown in Fig. 1. The first step $\mathcal{E}$ encodes the infinite-dimensional input using a finite number of degrees of freedom. The second approximation step Ψ maps the encoded input to an encoded, finite-dimensional output. The final reconstruction step $\mathcal{R}$ reconstructs an output function, given the finite-dimensional output of the approximation mapping. The composition of these encoding, approximation and reconstruction mappings thus takes an *input function* and maps it to another *output function*, and hence defines an operator. Existing operator learning frameworks differ in their particular choice of the encoder, neural network architecture and reconstruction mappings.

We start by giving background discussion on the CoD in finite dimensions, in Subsection 2.1. We then describe the subject in detail for neural network-based operator learning, resulting in our notion of the curse of parametric complexity, in Subsection 2.2. In Subsection 2.3 we state our main theorem concerning the curse of parametric complexity for neural operators. Subsection 2.4 demonstrates that the main theorem applies to PCA-Net, DeepONet and the NOMAD neural network architectures. Subsection 2.5 extends the main theorem to the FNO since it sits outside the framework introduced in Subsection 2.2.

2.1 *CoD for neural networks*

Since the neural network mapping $\Psi : \mathbb{R}^{D_\mathcal{X}} \to \mathbb{R}^{D_\mathcal{Y}}$ in the decomposition shown in Fig. 1 typically maps between *high-dimensional* (encoded) spaces, with $D_\mathcal{X}, D_\mathcal{Y} \gg 1$, most approaches to operator learning employ neural networks to learn this mapping. The motivation for this is that, empirically, neural networks have been found to be exceptionally well suited for the approximation of such high-dimensional functions in diverse applications (Goodfellow *et al.*, 2016). Detailed investigation of the approximation theory of neural networks, including quantitative upper and lower approximation error

bounds, has thus attracted a lot of attention in recent years (Yarotsky, 2017; Yarotsky & Zhevnerchuk, 2020; DeVore *et al.*, 2021; Kohler & Langer, 2021; Lu *et al.*, 2021a). Since we build on this analysis we summarize the relevant part of it here, restricting attention to ReLU neural networks in this work, as defined next; generalization to the use of other (piecewise polynomial) activation functions is possible.

2.1.1 *ReLU neural networks.* Fix integer L and integers $\{d_\ell\}_{\ell=0}^{L+1}$. Let $A_\ell \in \mathbb{R}^{d_{\ell+1}\times d_\ell}$ and $b_\ell \in \mathbb{R}^{d_{\ell+1}}$ for $\ell = 0, \dots, L$. A ReLU neural network $\Psi : \mathbb{R}^{D_\mathcal{X}} \to \mathbb{R}^{D_\mathcal{Y}}, x \mapsto \Psi(x)$ is a mapping of the form

$$\begin{cases} x_0 = x, \\ x_{\ell+1} = \sigma(A_\ell x_\ell + b_\ell), \quad \ell = 0, \cdots, L-1, \\ \Psi(x) = A_L x_L + b_L, \end{cases} \tag{2.1}$$

where $d_0 = D_\mathcal{X}$ and $d_{L+1} = D_\mathcal{Y}$. Here, the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is extended pointwise to act on any Euclidean space; and in what follows we employ the ReLU activation function $\sigma(x) = \min\{0, x\}$. We let $\theta := \{A_\ell, b_\ell\}_{\ell=0}^{L}$ and note that we have defined parametric mapping $\Psi(\,\cdot\,) = \Psi(\,\cdot\,;\theta)$. We define the depth of Ψ as the number of layers, and the size of Ψ as the number of nonzero weights and biases, i.e.,

$$\operatorname{depth}(\Psi) = L, \quad \operatorname{size}(\Psi) = \sum_{\ell=0}^{L} \left\{ \|A_\ell\|_0 + \|b_\ell\|_0 \right\},$$

where $\|\cdot\|_0$ counts the number of nonzero entries of a matrix or vector.

2.1.2 *Two simple facts from ReLU neural network calculus.* The following facts will be used without further comment (see e.g. (Opschoor *et al.*, 2022, section 2.2.3) for a discussion of more general results): If $\Psi : \mathbb{R}^{D_\mathcal{X}} \to \mathbb{R}^{D_\mathcal{Y}}$ is an ReLU neural network, and $A \in \mathbb{R}^{D_\mathcal{X}\times d}$ is a matrix, then there exists an ReLU neural network $\widetilde{\Psi} : \mathbb{R}^d \to \mathbb{R}^{D_\mathcal{Y}}$, such that

$$\begin{cases} \widetilde{\Psi}(x) = \Psi(Ax), & \text{for all } x \in \mathbb{R}^d, \\ \operatorname{depth}(\widetilde{\Psi}) = \operatorname{depth}(\Psi) + 1, \\ \operatorname{size}(\widetilde{\Psi}) \le 2\|A\|_0 + 2\operatorname{size}(\Psi). \end{cases} \tag{2.2}$$

Similarly, if $V \in \mathbb{R}^{d\times D_\mathcal{Y}}$ is a matrix then there exists an ReLU neural network $\widehat{\Psi} : \mathbb{R}^{D_\mathcal{X}} \to \mathbb{R}^d$, such that

$$\begin{cases} \widehat{\Psi}(x) = V\Psi(x), & \text{for all } x \in \mathbb{R}^d, \\ \operatorname{depth}(\widehat{\Psi}) = \operatorname{depth}(\Psi) + 1, \\ \operatorname{size}(\widehat{\Psi}) \le 2\|V\|_0 + 2\operatorname{size}(\Psi). \end{cases} \tag{2.3}$$

The main nontrivial issue in (2.2) and (2.3) is to preserve the potentially sparse structure of the underlying neural networks; this is based on a concept of 'sparse concatenation' from Petersen & Voigtlaender (2018).

2.1.3 *Approximation theory and CoD for ReLU networks.* One overall finding of research into the approximation power of ReLU neural networks is that, for function approximation in spaces characterized by smoothness, neural networks cannot entirely overcome the CoD (Yarotsky, 2017; Yarotsky & Zhevnerchuk, 2020; Kohler & Langer, 2021; Lu *et al.*, 2021a). This is illustrated by the following result, which builds on (Yarotsky, 2017, theorem 5) derived by D. Yarotsky:

PROPOSITION 2.1 (Neural network CoD). Let $r \in \mathbb{N}$ be given. For any dimension $D \in \mathbb{N}$ there exists $f_{D,r} \in C^r([0,1]^D; \mathbb{R})$ and constant $\overline{\epsilon}, \gamma > 0$, such that any ReLU neural network $\Psi : \mathbb{R}^D \to \mathbb{R}$ achieving accuracy

$$\sup_{x \in [0,1]^D} |f_{D,r}(x) - \Psi(x)| \leq \epsilon,$$

with $\epsilon \leq \overline{\epsilon}$, has size at least $\text{size}(\Psi) \geq \epsilon^{-\gamma D/r}$. The constant $\overline{\epsilon} = \overline{\epsilon}(r) > 0$ depends only on r, and $\gamma > 0$ is universal. ◊

The proof of Proposition 2.1 is included in Appendix A1. Proposition 2.1 shows that neural network approximation of a function between high-dimensional Euclidean spaces suffers from a CoD, characterized by an algebraic complexity with a potentially large exponent proportional to the dimension $D \gg 1$. This lower bound is similar to the approximation rates (upper bounds) achieved by traditional methods,[1] such as polynomial approximation. This fact suggests that the empirically observed efficiency of neural networks may well rely on additional structure of functions f of practical interest, beyond their smoothness; for relevant results in this direction see, for example, Mhaskar *et al.* (2016); Gribonval *et al.* (2022).

2.2 *Curse of parametric complexity in operator learning*

In this work we consider the approximation of an underlying operator $\mathcal{S}^\dagger : \mathcal{X} \to \mathcal{Y}$ acting between Banach spaces; specifically, we assume that the dimensions of the spaces $\mathcal{X}, \mathcal{Y}$ are *infinite*. Given the CoD in the finite-dimensional setting, Proposition 2.1 and letting $D \to \infty$ one would generally expect a super-algebraic, potentially even exponential, lower bound on the 'complexity' of neural operators $\mathcal{S}$ approximating such $\mathcal{S}^\dagger$, as a function of the accuracy ϵ. In this subsection we make this statement precise for a general class of neural operators, in Theorem 2.11. This is preceded by a discussion of relevant structure of compact sets in infinite-dimensional function spaces and a discussion of a suitable class of 'neural operators'.

2.2.1 *Infinite-dimensional hypercubes.* Proposition 2.1 was stated for the unit cube $[0,1]^D$ as the underlying domain. In the finite-dimensional setting of Proposition 2.1 the approximation rate turns out to be independent of the underlying compact domain, provided that the domain has nonempty interior and assuming a Lipschitz continuous boundary. This is in contrast to the infinite-dimensional case, where compact domains necessarily have empty interior and where the convergence rate depends on the specific structure of the domain. To state our complexity bound for operator approximation we will therefore need to discuss the prototypical structure of compact subsets $K \subset \mathcal{X}$.

[1] Ignoring the potentially beneficial factor γ.

To fix intuition we temporarily consider $\mathcal{X}$ a function space (for example, a Hölder, Lebesgue or Sobolev space). In this case, the most common way to define a compact subset $K \subset \mathcal{X}$ is via a smoothness constraint, as illustrated by the following concrete example:

EXAMPLE 2.2. Assume that $\mathcal{X} = C^s(\Omega)$ is the space of s-times continuously differentiable functions on a bounded domain $\Omega \subset \mathbb{R}^d$. Then, for $\rho > s$ and upper bound $M > 0$, the subset $K \subset \mathcal{X}$, defined as

$$K = \left\{ u \in C^\rho(\Omega) \,\middle|\, \|u\|_{C^\rho} \leq M \right\}, \tag{2.4}$$

is a compact subset of $\mathcal{X}$. Here, we define the C^ρ-norm as

$$\|u\|_{C^\rho} = \max_{|\nu| \leq \rho} \sup_{x \in \Omega} |D^\nu u(x)|. \tag{2.5}$$

◊

To better understand the approximation theory of operators $\mathcal{S}^\dagger : K \subset \mathcal{X} \to \mathcal{Y}$ we would like to understand the structure of such K. Our point of view is inspired by Fourier analysis, according to which smoothness of $u \in K$ roughly corresponds to a decay rate of the Fourier coefficients of u. In particular, u is guaranteed to belong to the set (2.4), if u is of the form

$$u = A \sum_{j=1}^{\infty} j^{-\alpha} y_j e_j, \qquad y_j \in [0,1] \text{ for all } j \in \mathbb{N}, \tag{2.6}$$

for a sufficiently large decay rate $\alpha > 0$, small constant $A > 0$ and where $e_j : \mathbb{R}^d \to \mathbb{R}$ denotes the periodic Fourier (sine/cosine) basis, restricted to Ω. We include a proof of this fact at the end of this subsection (see Lemma 2.7), where we also identify relevant decay rate α. In this sense the set K in (2.4) could be said to 'contain' an infinite-dimensional hypercube $\prod_{j=1}^{\infty}[0, Aj^{-\alpha}]$, with decay rate α. Such hypercubes will replace the finite-dimensional unit cube $[0,1]^D$ in our analysis of operator approximation in infinite dimensions.

We would like to point out that a similar observation holds for many other sets K defined by a smoothness constraint, such as sets in Sobolev spaces defined by a smoothness constraint, $\{\|u\|_{H^\rho} \leq M\} \subset H^s(\Omega)$, or more generally $\{\|u\|_{W^{\rho,p}} \leq M\} \subset W^{s,p}(\Omega)$, for any $1 \leq p \leq \infty$, but also Besov spaces, spaces of functions of bounded variation and others share a similar structure. Bounded balls in all of these spaces contain infinite-dimensional hypercubes, consisting of elements of the form (2.6). We note in passing that, in general, it may be more natural to replace the trigonometric basis in (2.6) by some other choice of basis, such as polynomials, splines, wavelets or a more general (frame) expansion. We refer to e.g., Christensen *et al.* (2003); Heil (2010) for general background and Herrmann *et al.* (2024) for an example of such a setting in the context of holomorphic operators.

The above considerations lead us to the following definition of an abstract hypercube:

DEFINITION 2.3. Let $e_1, e_2, \cdots \in \mathcal{X}$ be a sequence of linearly independent and normed elements, $\|e_j\|_{\mathcal{X}} = 1$. Given constants $A > 0$, $\alpha > 1$, we say that $K \subset \mathcal{X}$ **contains an infinite-dimensional cube** $Q_\alpha = Q_\alpha(A; e_1, e_2, \dots)$, if the following hold:

(1) K contains the set Q_α consisting of all u of the form (2.6);

(2) the set $\{e_j\}_{j\in\mathbb{N}}$ possesses a bounded bi-orthogonal sequence of functionals, labelled $e_1^*, e_2^*, \cdots \in \mathcal{X}^*$, in the continuous dual of $\mathcal{X}$,[2] i.e., we assume that $e_k^*(e_j) = \delta_{jk}$ for all $j, k \in \mathbb{N}$, and that there exists $M > 0$, such that $\|e_k^*\|_{\mathcal{X}^*} \leq M$ for all $k \in \mathbb{N}$.

◊

REMARK 2.4. Property (2) in Definition 2.3, i.e., the assumed existence of a bi-orthogonal sequence e_j^*, ensures that there exist 'coordinate projections': if u is of the form (2.6) then the jth coefficient y_j can be retrieved from u as $y_j = A^{-1}j^{\alpha}e_j^*(u)$. This allows us to uniquely identify $u \in Q_\alpha$ with a set of coefficients $(y_1, y_2, \dots) \in [0,1]^{\infty}$. ◊

REMARK 2.5. The decay rate α of the infinite-dimensional cube $Q_\alpha \subset K$ provides a measure of its 'asymptotic size' or 'complexity'. In terms of our complexity bounds this decay rate will play a special role. Hence, we will usually retain this dependence explicitly by writing Q_α, but suppress the additional dependence on A and $e_1, e_2, \dots$ in the following text. ◊

The notion of infinite-dimensional cubes introduced in Definition 2.3 is only a minor generalization of an established notion of cube embeddings, introduced by Donoho (Donoho, 2001) in a Hilbert space setting. We refer to (Dahlke *et al.*, 2015, chapter 5) for a pedagogical exposition of such cube embeddings in the Hilbert space setting, and their relation to the Kolmogorov entropy of K.

REMARK 2.6. The complexity bounds established in this work will be based on infinite-dimensional hypercubes. An interesting question, left for future work, is whether our main result on the curse of parametric complexity, Theorem 2.11, could be stated directly in terms of the Kolmogorov complexity of K, or other notions such as the Kolmogorov n-width (DeVore, 1998). ◊

Our definition of an infinite-dimensional hypercube is natural in view of the following lemma, the discussion following Example 2.2 and other similar results.

LEMMA 2.7. Assume that $\mathcal{X} = C^s(\Omega)$ is the space of s-times continuously differentiable functions on a bounded domain $\Omega \subset \mathbb{R}^d$. Choose $\rho > s$ and define K, compact in $\mathcal{X}$, by

$$K = \left\{u \in C^\rho(\Omega) \,\middle|\, \|u\|_{C^\rho} \leq M\right\},$$

with constant $M > 0$. Then K contains an infinite-dimensional hypercube Q_α, for any $\alpha > 1 + \frac{\rho - s}{d}$. ◊

The proof of Lemma 2.7 is included in Appendix A. While we have focused on spaces of continuously differentiable functions, similar considerations also apply to other smoothness spaces, such as the Sobolev spaces $W^{s,p}$, and more general Besov spaces.

2.2.2 *Curse of parametric complexity.* The main question to be addressed in this section is the following: given $K \subset \mathcal{X}$ compact, $\mathcal{S}^\dagger : \mathcal{X} \to \mathcal{Y}$ an r-times Fréchet differentiable operator to be approximated by a neural operator $\mathcal{S} : \mathcal{X} \to \mathcal{Y}$ and given a desired approximation accuracy $\epsilon > 0$ how many tunable parameters (in the architecture of $\mathcal{S}$) are required to achieve

$$\sup_{u\in K} \|\mathcal{S}^\dagger(u) - \mathcal{S}(u)\|_{\mathcal{Y}} \leq \epsilon?$$

[2] If $\mathcal{X}$ is a Hilbert space such a bi-orthogonal sequence always exists for independent $e_1, e_2, \cdots \in \mathcal{X}$.

The answer to this question clearly depends on our assumptions on $K \subset \mathcal{X}$, $\mathcal{Y}$ and the class of neural operators $\mathcal{S}$.

Assume $K \subset \mathcal{X}$ contains a hypercube Q_α. Consistent with our discussion in the last subsection we will assume that $K \subset \mathcal{X}$ contains an infinite-dimensional hypercube Q_α, as introduced in Definition 2.3, with algebraic decay rate $\alpha > 0$.

Assume $\mathcal{Y} = \mathbb{R}$, i.e., $\mathcal{S}^\dagger$ is a functional. The approximation of an operator $\mathcal{S}^\dagger : \mathcal{X} \to \mathcal{Y}$ with potentially infinite-dimensional output space $\mathcal{Y}$ is generally harder to achieve than the approximation of a functional $\mathcal{S}^\dagger : \mathcal{X} \to \mathbb{R}$ with one-dimensional output; indeed, if $\dim(\mathcal{Y}) \geq 1$ then $\mathbb{R}$ can be embedded in $\mathcal{Y}$ and any functional $\mathcal{X} \to \mathbb{R}$ gives rise to an operator $\mathcal{X} \to \mathcal{Y}$ under this embedding. To simplify our discussion we will therefore initially restrict attention to the approximation of functionals, with the aim of showing that even the approximation of r-times Fréchet differentiable functionals is generally intractable.

Assume $\mathcal{S}$ is of neural network-type. Assuming that $\mathcal{Y} = \mathbb{R}$ we must finally introduce a rigorous notion of the relevant class of approximating functionals $\mathcal{S} : \mathcal{X} \to \mathbb{R}$, i.e., define a class of 'neural operators/functionals' approximating the functional $\mathcal{S}^\dagger$.

DEFINITION 2.8 (Functional of neural network-type). We will say that a (neural) functional $\mathcal{S} : \mathcal{X} \to \mathbb{R}$ is a 'functional of neural network-type', if it can be written in the form

$$\mathcal{S}(u) = \Phi(\mathcal{L}u), \quad \forall\, u \in K, \tag{2.7}$$

where for some $\ell \in \mathbb{N}$, $\mathcal{L} : \mathcal{X} \to \mathbb{R}^\ell$ is a linear map, and $\Phi : \mathbb{R}^\ell \to \mathbb{R}$ is an ReLU neural network (potentially sparse). ◊

If $\mathcal{S}$ is a functional of neural network-type, we define the complexity of $\mathcal{S}$, denoted $\mathrm{cmplx}(\mathcal{S})$, as the smallest size of a neural network Φ for which there exists linear map $\mathcal{L}$ such that a representation of the form (2.7) holds, i.e.,

$$\mathrm{cmplx}(\mathcal{S}) := \min_{\Phi} \mathrm{size}(\Phi), \tag{2.8}$$

where the minimum is taken over all possible Φ in (2.7).

REMARK 2.9. Without loss of generality we may assume that $\ell \leq \mathrm{size}(\Phi)$ in (2.7) and (2.8). Indeed, if this is not the case then $\mathrm{size}(\Phi) < \ell$ and we can show that it is possible to construct another representation pair $(\widetilde{\Phi}, \widetilde{\mathcal{L}})$ in (2.7), consisting of a neural network $\widetilde{\Phi} : \mathbb{R}^{\widetilde{\ell}} \to \mathbb{R}$, linear map $\widetilde{\mathcal{L}} : \mathcal{X} \to \mathbb{R}^{\widetilde{\ell}}$ and such that $\widetilde{\ell} \leq \mathrm{size}(\widetilde{\Phi})$: To see why let us assume that $\widetilde{\ell} := \mathrm{size}(\Phi) < \ell$. Let A be the weight matrix in the first input layer of Φ. Since

$$\|A\|_0 \leq \widetilde{\ell} < \ell,$$

at most $\widetilde{\ell}$ columns of A can be nonvanishing. Write the matrix $A = [a_1, a_2, \ldots, a_\ell]$ in terms of its column vectors. Up to permutation we may assume that $a_{\widetilde{\ell}+1} = \cdots = a_\ell = 0$. We now drop the corresponding columns in the input layer of Φ and remove these unused components from the output of the linear map $\mathcal{L}$ in (2.7). This leads to a new map $\widetilde{\mathcal{L}} : \mathcal{X} \to \mathbb{R}^{\widetilde{\ell}}$, with output components $(\widetilde{\mathcal{L}}u)_j = (\mathcal{L}u)_j$ for $j = 1, \ldots, \widetilde{\ell}$, and we define $\widetilde{\Phi} : \mathbb{R}^{\widetilde{\ell}} \to \mathbb{R}$, as the neural network that is obtained from Φ by replacing the input matrix $A = [a_1, \ldots, a_\ell]$ by $\widetilde{A} = [a_1, \ldots, a_{\widetilde{\ell}}]$. Then, $\widetilde{\Phi} \circ \widetilde{\mathcal{L}} = \Phi \circ \mathcal{L}$, so that $\widetilde{\mathcal{L}}$ and $\widetilde{\Phi}$ satisfy a representation of

the form (2.7), but the dimension $\widetilde{\ell}$ satisfies $\widetilde{\ell} = \text{size}(\Phi) = \text{size}(\widetilde{\Phi})$; the first equality is by definition of $\widetilde{\ell}$, and the last equality holds because we only removed zero weights from Φ. In particular, this ensures that $\widetilde{\ell} \leq \text{size}(\widetilde{\Phi})$ for this new representation, without affecting the size of the underlying neural network, i.e., $\text{size}(\Phi) = \text{size}(\widetilde{\Phi})$. ◊

More generally, we can consider $\mathcal{Y} = \mathcal{Y}(\Omega; \mathbb{R}^p)$ a function space, consisting of functions $v : \Omega \to \mathbb{R}^p$ with domain $\Omega \subset \mathbb{R}^d$. Given $y \in \Omega$, we introduce the point-evaluation map

$$\text{ev}_y : \mathcal{Y} \to \mathbb{R}^p, \quad \text{ev}_y(v) := v(y).$$

Provided that point-evaluation ev_y is well-defined for all $v \in \mathcal{Y}$ we can readily extend the above notion to operators, as follows:

Definition 2.10 (Operator of neural network-type). Let $\mathcal{Y} = \mathcal{Y}(\Omega; \mathbb{R}^p)$ be a function space on which point-evaluation is well-defined. We will say that a (neural) operator $\mathcal{S} : \mathcal{X} \to \mathcal{Y}$ is an 'operator of neural network-type', if for any evaluation point $y \in \Omega$, the composition $\text{ev}_y \circ \mathcal{S} : \mathcal{X} \to \mathbb{R}^p$, $\text{ev}_y(\mathcal{S}(u)) := \mathcal{S}(u)(y)$, can be written in the form

$$\text{ev}_y \circ \mathcal{S}(u) = \Phi_y(\mathcal{L}u), \quad \forall\, u \in K, \tag{2.9}$$

where $\mathcal{L} : \mathcal{X} \to \mathbb{R}^\ell$ is a linear operator, and $\Phi_y : \mathbb{R}^\ell \to \mathbb{R}^p$ is an ReLU neural network that may depend on the evaluation point $y \in \Omega$. In this case, we define

$$\text{cmplx}(\mathcal{S}) := \sup_{y \in \Omega} \min_{\Phi_y} \text{size}(\Phi_y).$$

◊

We next state our main result demonstrating a 'curse of parametric complexity' for functionals (and operators) of neural network-type. This is followed by a detailed discussion of the implications of this abstract result for four representative examples of operator learning frameworks: PCA-Net, DeepONet, NOMAD and the Fourier neural operator.

2.3 *Main theorem on curse of parametric complexity*

The following result formalizes an analogue of the CoD in infinite-dimensions:

Theorem 2.11 (Curse of parametric complexity). Let $K \subset \mathcal{X}$ be a compact set in an infinite-dimensional Banach space $\mathcal{X}$. Assume that K contains an infinite-dimensional hypercube Q_α for some $\alpha > 1$. Then, for any $r \in \mathbb{N}$ and $\delta > 0$, there exists $\overline{\epsilon} > 0$ and an r-times Fréchet differentiable functional $\mathcal{S}^\dagger : K \subset \mathcal{X} \to \mathbb{R}$, such that approximation to accuracy $\epsilon \leq \overline{\epsilon}$ by a functional $\mathcal{S}_\epsilon$ of neural network-type,

$$\sup_{u \in K} |\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u)| \leq \epsilon, \tag{2.10}$$

implies complexity bound $\text{cmplx}(\mathcal{S}_\epsilon) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$; here $c, \overline{\epsilon} > 0$ are constants depending only on α, δ and r. ◊

Before providing a sketch of the proof of Theorem 2.11 we note the following simple corollary, whose proof is given in Appendix A.4.

COROLLARY 2.12. Let $K \subset \mathcal{X}$ be a compact set in an infinite-dimensional Banach space $\mathcal{X}$. Assume that K contains an infinite-dimensional hypercube Q_α for some $\alpha > 1$. Let $\mathcal{Y} = \mathcal{Y}(\Omega; \mathbb{R}^p)$ be a function space with continuous embedding in $C(\Omega; \mathbb{R}^p)$. Then, for any $r \in \mathbb{N}$ and $\delta > 0$, there exists $\overline{\epsilon} > 0$ and an r-times Fréchet differentiable functional $\mathcal{S}^\dagger : K \subset \mathcal{X} \to \mathcal{Y}$, such that approximation to accuracy $\epsilon \leq \overline{\epsilon}$ by an operator $\mathcal{S}_\epsilon : \mathcal{X} \to \mathcal{Y}$ of neural network-type,

$$\sup_{u \in K} \|\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u)\| \leq \epsilon, \tag{2.11}$$

implies complexity bound $\mathrm{cmplx}(\mathcal{S}_\epsilon) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$; here $c, \overline{\epsilon} > 0$ are constants depending only on α, δ and r. ♢

Proof of Theorem 2.11 (Sketch). Let $\mathcal{S}_\epsilon : \mathcal{X} \to \mathbb{R}$ be any functional of neural network-type, achieving approximation accuracy (2.10). In view of our definition of $\mathrm{cmplx}(\mathcal{S}_\epsilon)$ in (2.8), to prove the claim, it suffices to show that if $\mathcal{L} : \mathcal{X} \to \mathbb{R}^\ell$ is a linear map and $\Phi : \mathbb{R}^\ell \to \mathbb{R}$ is an ReLU neural network, such that

$$\mathcal{S}_\epsilon(u) = \Phi(\mathcal{L}u), \quad \forall\, u \in \mathcal{X},$$

then $\mathrm{size}(\Phi) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$.

The idea behind the proof of this fact is that if $K \subset \mathcal{X}$ contains a hypercube Q_α then for any $D \in \mathbb{N}$ a suitable rescaling of the finite-dimensional cube $[0,1]^D$ can be embedded in K. More precisely, for any $D \in \mathbb{N}$ there exists an injective linear map $\iota_D : [0,1]^D \to K$.

If we now consider the composition $\mathcal{S}_\epsilon \circ \iota_D : \mathbb{R}^D \to \mathbb{R}$ then we observe that we have a decomposition $\mathcal{S}_\epsilon \circ \iota_D(x) = \Phi(\mathcal{L} \circ \iota_D(x))$, where

$$\mathcal{L} \circ \iota_D : \mathbb{R}^D \to \mathbb{R}^\ell,$$

is linear, and

$$\Phi : \mathbb{R}^\ell \to \mathbb{R}$$

is an ReLU neural network. In particular, there exists a matrix $A \in \mathbb{R}^{\ell \times D}$, such that $\mathcal{L} \circ \iota_D(x) = Ax$ for all $x \in \mathbb{R}^D$, and the mapping $\Phi_D(x) := \mathcal{S}_\epsilon \circ \iota_D(x)$ defines an ReLU neural network $\Phi_D(x) = \Phi(Ax)$, whose size can be bounded by

$$\begin{aligned}
\mathrm{size}(\Phi_D) &\leq 2\,\mathrm{size}(\Phi) + 2\|A\|_0 && \text{(Equation 2.2)} \\
&\leq 2\,\mathrm{size}(\Phi) + 2\ell D \\
&\leq 2\,\mathrm{size}(\Phi) + 2\,\mathrm{size}(\Phi)D && \text{(Remark 2.9)} \\
&\leq 4\,\mathrm{size}(\Phi)D.
\end{aligned}$$

Using Proposition 2.1, for any $D \in \mathbb{N}$, we are then able to construct a functional $\mathcal{F}_D : K \subset \mathcal{X} \to \mathbb{R}$, mimicking the function $f_D : [0,1]^D \subset \mathbb{R}^D \to \mathbb{R}$ constructed in Proposition 2.1, and such that uniform

approximation of $\mathcal{F}_D$ by $\mathcal{S}_\epsilon$ (implying similar approximation of f_D by Φ_D) incurs a lower complexity bound

$$\text{size}(\Phi) \geq (4D)^{-1} \text{size}(\Phi_D) \geq C_D \epsilon^{-\gamma D/r},$$

where $C_D > 0$ is a constant depending on D. For this particular functional $\mathcal{F}_D$, and given the uniform lower bound on size(Φ) above, it then follows that

$$\text{cmplx}(\mathcal{S}_\epsilon) \geq C_D \epsilon^{-\gamma D/r}.$$

The first challenge is to make this strategy precise, and to determine the D-dependency of the constant C_D. As we will see this argument leads to a lower bound of roughly the form $\text{cmplx}(\mathcal{S}_\epsilon) \gtrsim (D^{r(1+\alpha)}\epsilon)^{-\gamma D/r}$.

At this point the argument is still for fixed $D \in \mathbb{N}$, and would only lead to an algebraic complexity in ϵ^{-1}. To extend this to an *exponential* lower bound in ϵ^{-1} we next observe that if the estimate $\text{cmplx}(\mathcal{S}_\epsilon) \gtrsim (D^{r(1+\alpha)}\epsilon)^{-\gamma D/r}$ could in fact be established for all $D \in \mathbb{N}$ simultaneously, i.e., if we could construct a *single* functional $\mathcal{S}^\dagger$, for which the lower complexity bound $\text{cmplx}(\mathcal{S}_\epsilon) \gtrsim \sup_{D\in\mathbb{N}}(D^{r(1+\alpha)}\epsilon)^{-\gamma D/r}$ were to hold, then setting $D \approx (e\epsilon)^{-1/(1+\alpha)r}$ on the right would imply that

$$\text{cmplx}(\mathcal{S}_\epsilon) \gtrsim \sup_{D\in\mathbb{N}} (D^{r(1+\alpha)}\epsilon)^{-\gamma D/r} \gtrsim \exp\left(c\epsilon^{-1/(1+\alpha)r}\right),$$

with suitable $c > 0$. Leading to an exponential lower complexity bound for such $\mathcal{S}^\dagger$ the second main challenge is thus to construct a single $\mathcal{S}^\dagger : K \subset \mathcal{X} \to \mathbb{R}$ that effectively 'embeds' an infinite family of functionals $\mathcal{F}_D : K \subset \mathcal{X} \to \mathbb{R}$ with complexity bounds as above. This will be achieved by defining $\mathcal{S}^\dagger$ as a weighted sum of suitable functionals $\mathcal{F}_D$. The detailed proof is provided in Appendix A3. □

Several remarks are in order:

REMARK 2.13. Theorem 2.11 shows rigorously that *in general* operator learning suffers from a curse of parametric complexity, in the sense that it is not possible to achieve better than exponential complexity bounds for general classes of operators, which are merely determined by their (C^r- or Lipschitz-) regularity. As explained above this is a natural infinite-dimensional analogue of the CoD in finite-dimensions (cp. Proposition 2.1), and motivates our terminology. We note that the lower bound of Theorem 2.11 qualitatively matches general upper bounds for DeepONets derived in Liu *et al.* (2024). It would be of interest to determine sharp rates. ♢

REMARK 2.14. Theorem 2.11 is derived for ReLU neural networks. With some effort, the argument could be extended to more general, piecewise polynomial activation functions. While we believe that the curse of parametric complexity has a fundamental character, we would like to point out that, *for nonstandard neural network architectures*, algebraic approximation rates have been obtained (Schwab *et al.*, 2023); these results build on either 'superexpressive' activation functions or other nonstandard architectures. Since these networks are not ordinary feedforward ReLU neural networks the algebraic approximation rates of Schwab *et al.* (2023) are not in contradiction with Theorem 2.11. While the parametric complexity of the nonstandard neural operators in Schwab *et al.* (2023) is exponentially smaller than the lower bound of Theorem 2.11, it is conceivable that storing the neural network weights in

practice would require exponential accuracy (number of bits), to account for the highly unstable character of super-expressive constructions. ♢

REMARK 2.15. Theorem 2.11 differs from previous results on the limitations of operator learning frameworks, as e.g., addressed in Lanthaler *et al.* (2022); Seidman *et al.* (2022); Lanthaler *et al.* (2023b). Earlier work focuses on the limitations imposed by a linear choice of the reconstruction mapping $\mathcal{R}$. In contrast, the results of this work exhibit C^k-smooth operators and functionals, which are fundamentally hard to approximate by neural network-based methods (with ReLU activation), irrespective of the choice of reconstruction. ♢

REMARK 2.16. We finally link our main theorem to a related result for PCA-Net (Lanthaler, 2023, theorem 3.3), there derived in a complementary Hilbert space setting for $\mathcal{X}$ and $\mathcal{Y}$; the result of Lanthaler (2023) shows that, for PCA-Net, no fixed algebraic convergence rate can be achieved in the operator learning of general C^r-operators between Hilbert spaces; this can be viewed as a milder version of the full curse of parametric complexity identified in this work, expressed by an *exponential lower complexity* bound in Theorem 2.11. ♢

To further illustrate an implication of Theorem 2.11 we provide the following example:

EXAMPLE 2.17 (Operator learning CoD). Let $\Omega \subset \mathbb{R}^d$ be a domain. Let $s, \rho \in \mathbb{Z}_{\geq 0}$ be given, with $s < \rho$, and consider the compact set

$$K = \left\{ u \in C^\rho(\Omega) \,\middle|\, \|u\|_{C^\rho} \leq 1 \right\} \subset C^s(\Omega).$$

Fix $r \in \mathbb{N}$. By Lemma 2.7 K contains an infinite-dimensional hypercube Q_α for *any* $\alpha > 1 + \frac{\rho - s}{d}$. Fix such α. Applying Theorem 2.11 it follows that there exists an r-times Fréchet differentiable functional $\mathcal{S}^\dagger : C^s(\Omega) \to \mathbb{R}$ and constant $c, \bar{\epsilon} > 0$, such that any family $\mathcal{S}_\epsilon : C^s(\Omega) \to \mathbb{R}$ of functionals of neural network-type, achieving accuracy

$$\sup_{u \in K} |\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u)| \leq \epsilon, \quad \forall \epsilon \leq \bar{\epsilon},$$

has complexity at least $\mathrm{cmplx}(\mathcal{S}_\epsilon) \geq \exp(c\epsilon^{-1/(1+\alpha)r})$ for $\epsilon \leq \bar{\epsilon}$. Furthermore, the constants $c, \bar{\epsilon} > 0$ depend only on the parameters r, s, ρ, α. ♢

In the next subsection we aim to show the relevance of the above abstract result for concrete neural operator architectures. Specifically, we show that three operator learning architectures from the literature are of neural network-type (PCA-Net, DeepONet, NOMAD), and relate our notion of complexity to the required number of tunable parameters for each. Finally, we show that even frameworks that are not necessarily of neural network-type could suffer from a similar curse of parametric complexity. We make this explicit for the Fourier neural operator in Subsection 2.5.

2.4 *Examples of operators of neural network-type*

We describe three representative neural operator architectures and show that they can be cast in the above framework.

PCA-Net. We start with the PCA-Net architecture from Bhattacharya *et al.* (2021), anticipated in the work Hesthaven & Ubbiali (2018). If $\mathcal{X}$ and $\mathcal{Y}$ are Hilbert spaces then a neural network can be combined with principal component analysis (PCA) for the encoding and reconstruction on the

underlying spaces, to define a neural operator architecture termed PCA-Net; the ingredients of this architecture are orthonormal PCA bases $\phi_1^{\mathcal{X}}, \dots, \phi_{D_{\mathcal{X}}}^{\mathcal{X}} \in \mathcal{X}$ and $\phi_1^{\mathcal{Y}}, \dots, \phi_{D_{\mathcal{Y}}}^{\mathcal{Y}} \in \mathcal{Y}$, and a neural network mapping $\Psi : \mathbb{R}^{D_{\mathcal{X}}} \to \mathbb{R}^{D_{\mathcal{Y}}}$. The encoder $\mathcal{E}$ is obtained by projection onto the $\{\phi_j^{\mathcal{X}}\}_{j=1}^{D_{\mathcal{X}}}$, whereas the reconstruction $\mathcal{R}$ is defined by a linear expansion in $\{\phi_j^{\mathcal{Y}}\}_{j=1}^{D_{\mathcal{Y}}}$. The resulting PCA-Net neural operator is defined as

$$\mathcal{S}(u)(y) = \sum_{k=1}^{D_{\mathcal{Y}}} \Psi_k(\alpha_1, \dots, \alpha_{D_{\mathcal{X}}}) \phi_k^{\mathcal{Y}}(y), \qquad \text{with } \alpha_j := \langle u, \phi_j^{\mathcal{X}} \rangle,\ j = 1, \dots, D_{\mathcal{X}}. \tag{2.12}$$

Here, the neural network $\Psi : \mathbb{R}^{D_{\mathcal{X}}} \to \mathbb{R}^{D_{\mathcal{Y}}}$, with components $\Psi_k(\cdot) = \Psi_k(\cdot\,; \theta)$, depends on parameters that are optimized during training of the PCA-Net. The PCA basis functions $\phi_1^{\mathcal{Y}}, \dots \phi_{D_{\mathcal{Y}}}^{\mathcal{Y}} : \Omega \to \mathbb{R}^p$, defining the reconstruction, are precomputed from the data using PCA analysis.

Given an evaluation point $y \in \Omega$ the composition $\mathrm{ev}_y \circ \mathcal{S}$, between $\mathcal{S}$ and the point-evaluation mapping $\mathrm{ev}_y(v) = v(y)$, can now be written in the form

$$\mathrm{ev}_y \circ \mathcal{S}(u) = \Phi_y(\mathcal{L}u),$$

where $\mathcal{L} : \mathcal{X} \to \mathbb{R}^{D_{\mathcal{X}}}$, $\mathcal{L}u := (\langle u, \phi_1^{\mathcal{X}} \rangle, \dots, \langle u, \phi_{D_{\mathcal{X}}}^{\mathcal{X}} \rangle$ is a linear mapping, and $\Phi_y(\alpha) := \sum_{k=1}^{D_{\mathcal{Y}}} \Psi_k(\alpha) \phi_k^{\mathcal{Y}}(y)$, for fixed y, is the composition of a neural network Ψ with a linear read-out; thus, Φ_y is itself a neural network for fixed y. This shows that PCA-Net is of neural network-type.

The following lemma shows that the complexity of PCA-Net gives a lower bound on the number of free parameters for the underlying neural network architecture.

LEMMA 2.18 (Complexity of PCA-Net). Assume that $\mathcal{X}$ and $\mathcal{Y}$ are Hilbert spaces, so that PCA-Net is well-defined. Any PCA-Net $\mathcal{S} = \mathcal{R} \circ \Psi \circ \mathcal{E}$ is of neural network-type, and

$$\mathrm{size}(\Psi) \geq (2p + 2)^{-1} \mathrm{cmplx}(\mathcal{S}),$$

where the dimension $p \in \mathbb{N}$ is fixed by the output-function space $\mathcal{Y} = \mathcal{Y}(\Omega; \mathbb{R}^p)$. ◊

Thus, Lemma 2.18 implies a lower complexity bound $\mathrm{size}(\Psi) \gtrsim \mathrm{cmplx}(\mathcal{S})$. The detailed proof is given in Appendix A.5.1. It thus follows from Corollary 2.12 that operator learning with PCA-Net suffers from a curse of parametric complexity:

PROPOSITION 2.19 (Curse of parametric complexity for PCA-Net). Assume the setting of Corollary 2.12, with $\mathcal{X}, \mathcal{Y}$ being Hilbert spaces. Then, for any $r \in \mathbb{N}$ and $\delta > 0$, there exists $\overline{\epsilon} > 0$ and an r-times Fréchet differentiable functional $\mathcal{S}^\dagger : K \subset \mathcal{X} \to \mathcal{Y}$, such that approximation to accuracy $\epsilon \leq \overline{\epsilon}$ by a PCA-Net $\mathcal{S}_\epsilon : \mathcal{X} \to \mathcal{Y}$

$$\sup_{u \in K} \|\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u)\| \leq \epsilon, \tag{2.13}$$

with encoder $\mathcal{E}$, neural network Ψ and reconstruction $\mathcal{R}$, implies complexity bound $\mathrm{size}(\Psi) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$; here $c, \overline{\epsilon} > 0$ are constants depending only on α, δ, r and p. ◊

DeepONet. A conceptually similar approach is followed by the DeepONet of Lu *et al.* (2021b), which differs by learning the form of the representation in $\mathcal{Y}$ concurrently with the coefficients, and by allowing for quite general input linear functionals $\{\alpha_j\}_{j=1}^{D_\mathcal{X}}$.

The DeepONet architecture defines the encoding $\mathcal{E} : \mathcal{X} \to \mathbb{R}^{D_\mathcal{X}}$ by a fixed choice of general linear functionals $\ell_1, \ldots, \ell_{D_\mathcal{X}}$; these may be obtained, for example, by point evaluation at distinct 'sensor points' or by projection onto PCA modes as in PCA-Net. The reconstruction $\mathcal{R} : \mathbb{R}^{D_\mathcal{Y}} \to \mathcal{Y}$ is defined by expansion with respect to a set of functions $\phi_1, \ldots, \phi_{D_\mathcal{Y}} \in \mathcal{Y}$ that are themselves finite-dimensional neural networks to be learned. The resulting DeepONet can be written as

$$\mathcal{S}(u)(y) = \sum_{k=1}^{D_\mathcal{Y}} \Psi_k(\alpha_1, \ldots, \alpha_{D_\mathcal{X}})\phi_k(y), \qquad \text{with } \alpha_j := \ell_j(u),\ j = 1, \ldots, D_\mathcal{X}. \tag{2.14}$$

Here, both the neural networks $\Psi : \mathbb{R}^{D_\mathcal{X}} \to \mathbb{R}^{D_\mathcal{Y}}$, with components $\Psi_k = \Psi_k(\cdot\,;\theta)$, and $\phi : \Omega \to \mathbb{R}^{p \times D_\mathcal{Y}}$ with components $\phi_k = \phi_k(\cdot\,;\theta)$, depend on parameters that are optimized during training of the DeepONet.

Given an evaluation point $y \in \Omega$ the composition $\mathrm{ev}_y \circ \mathcal{S}$, with the point-evaluation mapping $\mathrm{ev}_y(v) = v(y)$, can again be written in the form

$$\mathrm{ev}_y \circ \mathcal{S}(u) = \Phi_y(\mathcal{L}u),$$

where $\mathcal{L} : \mathcal{X} \to \mathbb{R}^{D_\mathcal{X}}$, $\mathcal{L}u := (\ell_1(u), \ldots, \ell_{D_\mathcal{X}}(u))$ is linear, where

$$\Phi_y(\alpha) := \sum_{k=1}^{D_\mathcal{Y}} \Psi_k(\alpha)\phi_k(y),$$

and for fixed y the values $\phi_k(y) \in \mathbb{R}^p$ are just (constant) vectors. Thus, Φ_y is the composition of a neural network Ψ with a linear read-out, and hence is itself a neural network. This shows that DeepONet is of neural network-type.

The next lemma shows that the size can be related to the complexity of DeepONet: also in this case $\mathrm{cmplx}(\mathcal{S})$ provides a lower bound on the total number of nonzero degrees of freedom of a DeepONet. The detailed proof is provided in Appendix A.5.2.

LEMMA 2.20 (Complexity of DeepONet). Any DeepONet $\mathcal{S} = \mathcal{R} \circ \Psi \circ \mathcal{E}$, defined by a branch-net Ψ and trunk-net ϕ, is of neural network-type, and

$$2(\mathrm{size}(\Psi) + \mathrm{size}(\phi)) \geq \mathrm{cmplx}(\mathcal{S}).$$

$\Diamond$

The following result is now an immediate consequence of Corollary 2.12 and the above lemma.

PROPOSITION 2.21 (Curse of parametric complexity for DeepONet). Assume the setting of Corollary 2.12. Then, for any $r \in \mathbb{N}$ and $\delta > 0$, there exists $\overline{\epsilon} > 0$ and an r-times Fréchet differentiable functional

$\mathcal{S}^\dagger : K \subset \mathcal{X} \to \mathcal{Y}$, such that approximation to accuracy $\epsilon \leq \overline{\epsilon}$ by a DeepONet $\mathcal{S}_\epsilon : \mathcal{X} \to \mathcal{Y}$

$$\sup_{u \in K} \|\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u)\| \leq \epsilon, \tag{2.15}$$

with branch net Ψ and trunk net ϕ, implies complexity bound $\text{size}(\Psi) + \text{size}(\phi) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$; here $c, \overline{\epsilon} > 0$ are constants depending only on α, δ and r. ◊

NOMAD. The linearity in the reconstruction in $\mathcal{Y}$ for both PCA-Net and DeepONet imposes a fundamental limitation on their approximation capability (Lanthaler *et al.*, 2022; Seidman *et al.*, 2022; Lanthaler *et al.*, 2023b). To overcome this limitation nonlinear extensions of DeepONet have recently been proposed. The following NOMAD architecture (Seidman *et al.*, 2022) provides an example:

NOMAD (NOnlinear MAnifold Decoder) employs encoding by point evaluation at a fixed set of sensor points, or more general linear functionals $\ell_1, \ldots, \ell_{D_\mathcal{X}} : \mathcal{X} \to \mathbb{R}$. The reconstruction $\mathcal{R} : \mathbb{R}^{D_\mathcal{Y}} \to \mathcal{Y}$ in the output space $\mathcal{Y} = \mathcal{Y}(\Omega; \mathbb{R}^p)$ is defined via a neural network $Q : \mathbb{R}^{D_\mathcal{Y}} \times \Omega \to \mathbb{R}^p$, which depends jointly on encoded output coefficients in $\mathbb{R}^{D_\mathcal{Y}}$ and the evaluation point $y \in \Omega$; as in the two previous examples, $\Psi : \mathbb{R}^{D_\mathcal{X}} \to \mathbb{R}^{D_\mathcal{Y}}$ is again a neural network. The resulting NOMAD mapping, for $u \in \mathcal{X}$ and $y \in \Omega$, is given by

$$\mathcal{S}(u)(y) = Q(\Psi(\alpha_1, \ldots, \alpha_{D_\mathcal{X}}), y), \qquad \text{with } \alpha_j := \ell_j(u), \tag{2.16}$$

for $j = 1, \ldots, D_\mathcal{X}$. We note that the main difference between DeepONet and NOMAD is that the linear expansion in (2.14) has been replaced by a nonlinear composition with the neural network Q. Both neural networks Ψ and Q are optimized during training.

Given evaluation point $y \in \Omega$ the composition $\text{ev}_y \circ \mathcal{S}$ with the point-evaluation can be written in the form

$$\text{ev}_y \circ \mathcal{S}(u) = \Phi_y(\mathcal{L}u),$$

where $\mathcal{L} : \mathcal{X} \to \mathbb{R}^{D_\mathcal{X}}$, $\mathcal{L}u := (\ell_1(u), \ldots, \ell_{D_\mathcal{X}}(u))$ is linear, and $\alpha \mapsto \Phi_y(\alpha) := Q(\Psi_k(\alpha), y)$ defines a neural network for fixed y. This shows that NOMAD is of neural network-type. Finally, the following lemma provides an estimate on the complexity of NOMAD:

LEMMA 2.22 (Complexity of NOMAD). Any NOMAD operator $\mathcal{S} = \mathcal{R} \circ \Psi \circ \mathcal{E}$ defined by a branch-net Ψ and nonlinear reconstruction Q is of neural network-type, and

$$2(\text{size}(\Psi) + \text{size}(Q)) \geq \text{cmplx}(\mathcal{S}). \tag{2.17}$$

◊

The expression on the left-hand side of (2.17) represents the total number of nonzero degrees of freedom in the NOMAD architecture and, as for PCA-Net and DeepONet, it is lower bounded by our notion of complexity. For the proof, we refer to Appendix A.5.3.

The following result is now an immediate consequence of Corollary 2.12 and the above lemma.

PROPOSITION 2.23 (Curse of parametric complexity for NOMAD). Assume the setting of Corollary 2.12. Then, for any $r \in \mathbb{N}$ and $\delta > 0$, there exists $\overline{\epsilon} > 0$ and an r-times Fréchet differentiable functional

$\mathcal{S}^\dagger : K \subset \mathcal{X} \to \mathcal{Y}$, such that approximation to accuracy $\epsilon \leq \overline{\epsilon}$ by a NOMAD neural operator $\mathcal{S}_\epsilon : \mathcal{X} \to \mathcal{Y}$

$$\sup_{u \in K} \|\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u)\| \leq \epsilon, \tag{2.18}$$

with neural networks Ψ and Q, implies complexity bound $\text{size}(\Psi) + \text{size}(Q) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$; here $c, \overline{\epsilon} > 0$ are constants depending only on α, δ and r. ◊

Discussion. For all examples above a general lower bound on the complexity $\text{cmplx}(\mathcal{S})$ of operators of neural network-type implies a lower bound on the total number of degrees of freedom of the particular architecture. In particular, a lower bound on $\text{cmplx}(\mathcal{S})$ gave a lower bound on the smallest possible number of nonzero parameters that are needed to implement $\mathcal{S}$ in practice. This observation motivates our nomenclature for the complexity.

We emphasize that our notion of complexity only relates to the size of the neural network at the core of these architectures; by design, it does not take into account other factors, such as the additional complexity associated with the practical evaluation of inner products in the PCA-Net architecture, evaluation of linear encoding functionals of DeepONets or the numerical representation of an (optimal) output PCA basis for PCA-Net or neural network basis for DeepONet. The important point is that the aforementioned factors can only increase the overall complexity of any concrete implementation; correspondingly, our proposed notion of $\text{cmplx}(\mathcal{S})$, which neglects some of these additional contributions, can be used to derive rigorous *lower* bounds on the overall complexity of any implementation.

REMARK 2.24. In passing, we point out similar approaches (Hua & Lu, 2023; Zhang *et al.*, 2023; Lanthaler *et al.*, 2023b; Patel *et al.*, 2024) to the PCA-Net, DeepONet and NOMAD architectures, which share a closely related underlying structure to the examples given above. We fully expect that the curse of parametric complexity applies to all of these architectures. ◊

In the next Subsection 2.5 we will show that, even for operator architectures that are not of neural network-type according to the above definition, we may nevertheless be able to link them with an associated operator of neural network-type. Specifically, we will show this for the FNO in Theorem 2.27. There, we will see that the size (number of tunable parameters) of the FNO can be linked to the complexity of an associated operator of neural network-type. And hence lower bounds on the complexity of operators of neural network-type imply corresponding lower bounds on the FNO.

2.5 *The curse of (parametric) complexity for fourier neural operators*

The definition of *operators of neural network-type* introduced in the previous subsection does not include the FNO, a widely adopted neural operator architecture. However, in this subsection we show that a result similar to Theorem 2.11, stated as Theorem 2.27, can be obtained for the FNO.

Due to intrinsic constraints on the domain on which FNO can be (readily) applied we will assume that the spatial domain $\Omega = \prod_{j=1}^d [a_j, b_j] \subset \mathbb{R}^d$ is rectangular. We recall that an FNO,

$$\mathcal{S} : \mathcal{X}(\Omega; \mathbb{R}^k) \to \mathcal{Y}(\Omega; \mathbb{R}^p),$$

can be written as a composition, $\mathcal{S} = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{P}$, of a lifting layer $\mathcal{P}$, hidden layers $\mathcal{L}_1, \ldots, \mathcal{L}_L$ and an output layer $\mathcal{Q}$. In the following text let us denote by $\mathcal{V}(\Omega; \mathbb{R}^{d_v})$ a generic space of functions from Ω to $\mathbb{R}^{d_v}$.

The nonlinear **lifting layer**

$$\mathcal{P} : \mathcal{X}(\Omega;\mathbb{R}^k) \to \mathcal{V}(\Omega;\mathbb{R}^{d_v}), \quad u(x) \mapsto \chi(x, u(x)),$$

is defined by a neural network $\chi : \Omega \times \mathbb{R}^k \to \mathbb{R}^{d_v}$, depending jointly on the evaluation point $x \in \Omega$ and the components of the input function u evaluated at x, namely $u(x) \in \mathbb{R}^k$. The dimension d_v is a free hyperparameter, and determines the number of 'channels' (or the 'lifting dimension').

Each **hidden layer** $\mathcal{L}_\ell$, $\ell = 1, \dots, L$, of an FNO is of the form

$$\mathcal{L}_\ell : \mathcal{V}(\Omega;\mathbb{R}^{d_v}) \to \mathcal{V}(\Omega;\mathbb{R}^{d_v}), \quad v \mapsto \sigma\left(W_\ell v + K_\ell v + b_\ell\right),$$

where σ is a nonlinear activation function applied componentwise, $W_\ell \in \mathbb{R}^{d_v \times d_v}$ is a matrix, the bias $b_\ell \in \mathcal{V}(\Omega;\mathbb{R}^{d_v})$ is a function and $K_\ell : \mathcal{V}(\Omega;\mathbb{R}^{d_v}) \to \mathcal{V}(\Omega;\mathbb{R}^{d_v})$ is a nonlocal operator defined in terms of **Fourier multipliers**,

$$Kv(x) = \mathcal{F}^{-1}(\widehat{T}_k[\mathcal{F}v]_k)(x),$$

where $[\mathcal{F}v]_k$ denotes the kth Fourier coefficient of v for $k \in \mathbb{Z}^d$, $\widehat{T}_k \in \mathbb{C}^{d_v \times d_v}$ is a Fourier multiplier matrix indexed by k and $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. In practice, a Fourier cut-off $k_{\max} \in \mathbb{Z}$ is introduced, and only a finite number of Fourier modes[3] $\|k\|_{\ell^\infty} \le k_{\max}$ is retained. In particular, the number of nonzero components of $\widehat{T} = \{\widehat{T}_k\}_{k\in\mathbb{Z}^d}$ is bounded by $\|\widehat{T}\|_0 \le d_v^2(2k_{\max}+1)^d$. In the following text we will also assume that the bias functions b_ℓ are determined by their Fourier components $[\widehat{b}_\ell]_k \in \mathbb{C}^{d_v}$, $\|k\|_{\ell^\infty} \le k_{\max}$.

Finally, the **output layer**

$$\mathcal{Q} : \mathcal{V}(\Omega;\mathbb{R}^{d_v}) \to \mathcal{Y}(\Omega;\mathbb{R}^p), \quad v \mapsto q(x, v(x)),$$

is defined in terms of a neural network $q : \Omega \times \mathbb{R}^{d_v} \to \mathbb{R}^p$, a joint function of the evaluation point $x \in \Omega$ and the components of the output v of the previous layer evaluated at x, namely $v(x) \in \mathbb{R}^{d_v}$.

To define the **size of an FNO** we note that its tunable parameters are given by (i) the weights and biases of the neural network χ defining the lifting layer $\mathcal{R}$, (ii) the components of the matrices $W_\ell \in \mathbb{R}^{d_v \times d_v}$, (iii) the components of the Fourier multipliers $\widehat{T}_k \in \mathbb{C}^{d_v \times d_v}$ for $\|k\|_{\ell^\infty} \le k_{\max}$, (iv) the Fourier coefficients $[\widehat{b}_\ell]_k \in \mathbb{C}^{d_v}$, for $\|k\|_{\ell^\infty} \le k_{\max}$, and (v) the number of weights and biases of the neural network q defining the output layer $\mathcal{Q}$. Given an FNO $\mathcal{S} : \mathcal{X}(\Omega;\mathbb{R}^k) \to \mathcal{Y}(\Omega;\mathbb{R}^p)$ we define size($\mathcal{S}$) of an FNO as the total number of nonzero parameters in this construction. We also follow the convention that for a matrix (or vector) A with complex entries the number of parameters is defined as $\|A\|_0 = \|\mathrm{Re}(A)\|_0 + \|\mathrm{Im}(A)\|_0$.

REMARK 2.25 (FNO approximation of functionals). If $\mathcal{S}^\dagger : \mathcal{X}(\Omega;\mathbb{R}^k) \to \mathbb{R}$ is a (scalar-valued) functional then we will again identify the output-space $\mathcal{Y}(\Omega;\mathbb{R}^p)$ with a space of constant functions. In this case, it is natural to add an averaging operation after the last output layer $\mathcal{Q} : \mathcal{V}(\Omega;\mathbb{R}^{d_v}) \to \mathcal{Y}(\Omega;\mathbb{R})$,

[3] Throughout this paper $\|\cdot\|_{\ell^\infty}$ denotes the maximum norm on finite-dimensional Euclidean space.

i.e., we replace $\mathcal{Q}$ by $\widetilde{\mathcal{Q}} : \mathcal{V}(\Omega; \mathbb{R}^{d_v}) \to \mathbb{R}$, given by

$$\widetilde{\mathcal{Q}}(v) := \frac{1}{|\Omega|} \int_\Omega q(x, v(x))\, \mathrm{d}x. \tag{2.19}$$

This does not introduce any additional degrees of freedom, and ensures that the output is constant. We also note that (2.19) is a special case of a Fourier multiplier K, involving only the $k = 0$ Fourier mode. ◊

In the following text we will restrict attention to the approximation of functionals, taking into account Remark 2.25. We first mention the following result, proved in Appendix A.6, which shows that FNOs are not of neural network-type, in general:

LEMMA 2.26. Let σ be the ReLU activation function. Let $\mathbb{T} \simeq [0, 2\pi]$ denote the 2π-periodic torus. The FNO,

$$\mathcal{S} : L^2(\mathbb{T}; \mathbb{R}) \to \mathbb{R}, \quad \mathcal{S}(u) := \int_\Omega \sigma(u(x))\, \mathrm{d}x,$$

is not of neural network-type. ◊

The fact that FNO is not of neural network-type is closely related to the fact that the Fourier transforms at the core of the FNO mapping $\mathcal{S} : \mathcal{X}(\Omega; \mathbb{R}^k) \to \mathbb{R}$ cannot be computed exactly. In practice, the FNO therefore needs to be **discretized**.

A simple discretization $\mathcal{S}^N$ of $\mathcal{S}$ is readily obtained and commonly used in applications of FNOs. To this end fix $N \in \mathbb{N}$, and let $x_{j_1,\ldots,j_d} \in \Omega, j_1, \ldots, j_d \in \{1, \ldots, N\}$ be a grid consisting of N equidistant points in each coordinate direction. A numerical approximation $\mathcal{S}^N : \mathcal{X}(\Omega; \mathbb{R}^k) \to \mathbb{R}$ of $\mathcal{S}$ is obtained by replacing the Fourier transform $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$ in each hidden layer by the *discrete* Fourier transform $\mathcal{F}_N$ and its inverse $\mathcal{F}_N^{-1}$, computed on the equidistant grid. Similarly, the exact average (2.19) is replaced by an average over the grid values. This 'discretized' FNO $\mathcal{S}^N$ thus defines a mapping

$$\mathcal{S}^N : \mathcal{X}(\Omega; \mathbb{R}^k) \to \mathbb{R}, \quad u \mapsto \mathcal{S}^N(u),$$

which depends only on the grid values $u(x_{j_1,\ldots,j_d}) \in \mathbb{R}^k$ of the input function u, for multi-indices $(j_1, \ldots, j_d) \in \{1, \ldots, N\}^d$. In contrast to $\mathcal{S}$ the discretization $\mathcal{S}^N$ is readily implemented in practice. We expect that $\mathcal{S}^N(u) \approx \mathcal{S}(u)$ for sufficiently large N. Note also that $\mathrm{size}(\mathcal{S}^N) = \mathrm{size}(\mathcal{S})$ by construction.

Given these preparatory remarks we can now state our result on the curse of parametric complexity for FNOs, with proof in Appendix A7.

THEOREM 2.27. Let $\Omega \subset \mathbb{R}^d$ be a cube. Let $K \subset \mathcal{X}$ be a compact subset of a Banach space $\mathcal{X} = \mathcal{X}(\Omega; \mathbb{R}^k)$. Assume that K contains an infinite-dimensional hypercube Q_α for some $\alpha > 1$. Then, for any $r \in \mathbb{N}$ and $\delta > 0$, there exists $\overline{\epsilon} > 0$ and an r-times Fréchet differentiable functional $\mathcal{S}^\dagger : K \subset \mathcal{X} \to \mathbb{R}$, such that approximation to accuracy $\epsilon \leq \overline{\epsilon}$ by a discretized FNO $\mathcal{S}_\epsilon^{N_\epsilon}$,

$$\sup_{u \in K} |\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon^{N_\epsilon}(u)| \leq \epsilon,$$

requires either (i) complexity bound $\text{size}(\mathcal{S}_\epsilon^{N_\epsilon}) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$, or (ii) discretization parameter $N_\epsilon \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$; here $c, \bar{\epsilon} > 0$ are constants depending only on α, δ and r. ◊

Proof (*Sketch*). The proof of Theorem 2.27 relies on the curse of parametric complexity for operators of neural network-type in Theorem 2.11. The first step is to show that discrete FNOs are of neural network-type. As a consequence, Theorem 2.11 implies a lower bound of the form $\text{cmplx}(\mathcal{S}_\epsilon^{N_\epsilon}) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$. The main additional difficulty is that the discrete FNO is not a standard ReLU neural network according to our definition, since it employs a very nonstandard architecture involving convolution and Fourier transforms. Hence, more work is needed, in order to relate $\text{cmplx}(\mathcal{S}_\epsilon^{N_\epsilon})$ to $\text{size}(\mathcal{S}_\epsilon^{N_\epsilon})$; while the complexity is the minimal number of parameters required to represent $\mathcal{S}_\epsilon^{N_\epsilon}$ by an *ordinary* ReLU network architecture, we recall that the $\text{size}(\mathcal{S}_\epsilon^{N_\epsilon})$ is defined as the number of parameters defining $\mathcal{S}_\epsilon^{N_\epsilon}$ via the *nonstandard* FNO architecture. Our analysis leads to an upper bound of the form

$$\text{cmplx}(\mathcal{S}_\epsilon^{N_\epsilon}) \lesssim N_\epsilon^{2d} \text{size}(\mathcal{S}_\epsilon^{N_\epsilon}). \tag{2.20}$$

As a consequence of the exponential lower bound, $\text{cmplx}(\mathcal{S}_\epsilon^{N_\epsilon}) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$, inequality (2.20) implies that either N_ϵ or $\text{size}(\mathcal{S}_\epsilon^{N_\epsilon})$ have to be exponentially large in ϵ^{-1}, proving the claim. For the detailed proof we refer to Appendix A7. □

Remark 2.28. We would like to point out that the additional factor in (2.20), depending on N_ϵ, is natural in view of the fact that even a linear discretized FNO layer $v \mapsto Wv$, with matrix $W \in \mathbb{R}^{d_v \times d_v}$, actually corresponds to a mapping $\mathbb{R}^{N_\epsilon^d \times d_v} \to \mathbb{R}^{N_\epsilon^d \times d_v}$, $(v(x_{j_1,\dots,j_d})) \mapsto (Wv(x)_{j_1,\dots,j_d})$, i.e., the matrix multiplication should be viewed as being carried out in parallel at all N_ϵ^d grid points. Representing this mapping by an ordinary matrix multiplication $\mathbb{R}^{N_\epsilon^d \times d_v} \to \mathbb{R}^{N_\epsilon^d \times d_v}$ requires $N_\epsilon^d \|W\|_0$ degrees of freedom, and thus depends on N_ϵ in addition to the number of FNO parameters $\|W\|_0$. ◊

Remark 2.29. Theorem 2.27 shows that FNO suffers from a similar curse of complexity as do the variants on DeepONet and PCA-Net covered by Theorem 2.11: approximation to accuracy ϵ of general (C^r- or Lipschitz-) operators requires either an exponential number of nonzero degree of freedom, or requires exponential computational resources to evaluate even a single forward pass. We note the difference in how the curse is expressed in Theorem 2.27 compared with Theorem 2.11; this is due to the fact that FNO is not of neural network-type (see Lemma 2.26). Instead, as outlined in the proof sketch above, only upon discretization does the FNO define an operator/functional of neural network type. The computational complexity of this discretized FNO is determined by both the FNO coefficients and the discretization parameter N. ◊

2.5.1 *Discussion.* To overcome the general curse of parametric complexity implied by Theorem 2.11 (and Theorem 2.27) efficient operator learning frameworks therefore have to leverage additional structure present in the operators of interest, going beyond C^r- or Lipschitz-regularity. Previous work on overcoming the curse of parametric complexity for operator learning has mostly focused on operator holomorphy (Schwab & Zech, 2019; Herrmann *et al.*, 2020; Lanthaler *et al.*, 2022) and the emulation of numerical methods (Kovachki *et al.*, 2021; Deng *et al.*, 2022; Lanthaler *et al.*, 2022) as two basic mechanisms for overcoming the curse of parametric complexity for specific operators of interest. An abstract characterization of the entire class of operators that allow for efficient approximation by neural operators would be very desirable. Unfortunately, this appears to be out of reach, at the current state of analysis. Indeed, as far as the authors are aware, there does not even exist such a characterization for

any class of standard numerical methods, such as finite difference, finite element or spectral, viewed as operator approximators.

The second contribution of this work is to expose additional structure, different from holomorphy and emulation, that can be leveraged by neural operators. In particular, in the remainder of this paper we identify such structure in the HJ equations, and propose a neural operator framework that can build on this structure to provably overcome the curse of parametric complexity in learning the associated solution operator.

3. The HJ equation

In the previous section we demonstrate that, generically, a scaling-limit of the curse of dimensionality is to be expected in operator approximation, if only C^r- regularity of the map is assumed. However, we also outlined in the introduction the many cases where specific structure can be exploited to overcome this curse. In the remainder of the paper we show how the curse can be removed for HJ equations. To this end we start, in this section, by setting up the theoretical framework for operator learning in this context.

We are interested in deriving error and complexity estimates for the approximation of the solution operator $\mathcal{S}_t^\dagger : C^r_{\text{per}}(\Omega) \to C^r_{\text{per}}(\Omega)$ associated with the following HJ PDE:

$$\begin{cases} \partial_t u + H(q, \nabla_q u) = 0, (x,t) \in \Omega \times (0,T], \\ \qquad u(t=0) = u_0, (x,t) \in \Omega \times \{0\}. \end{cases} \tag{HJ}$$

To simplify our analysis we consider only the case of a domain $\Omega = [0, 2\pi]^d$ with periodic boundary conditions (so that we may identify Ω with $\mathbb{T}^d$). We denote by $C^r_{\text{per}}(\Omega)$ the space of r-times continuously differentiable real-valued functions with 2π-periodic derivatives, with norm

$$\|u\|_{C^r(\Omega)} := \sup_{|\alpha| \le r} \sup_{x \in \Omega} |D^\alpha u(x)|.$$

By slight abuse of notation we will similarly denote by $u(q,t) \in C^r_{\text{per}}(\Omega \times [0,T])$ and $H(q,p) \in C^r_{\text{per}}(\Omega \times \mathbb{R}^d)$ functions that have C^r regularity in all variables, and that are periodic in the first variable $q \in \Omega$, so that in particular,

$$q \mapsto u(q,t) \quad \text{and} \quad q \mapsto H(q,p),$$

belong to $C^r_{\text{per}}(\Omega)$, for fixed $p \in \mathbb{R}^d$ or $t \in [0,T]$.

In equation (HJ) $u : \Omega \times [0,T] \to \mathbb{R}$, $(q,t) \mapsto u(q,t)$ is a function, depending on the spatial variable $q \in \Omega$ and on time $t \ge 0$. We will restrict attention to problems for which a *classical solution* $u \in C^r_{\text{per}}(\Omega \times [0,T])$, $r \ge 2$, exists. In this setting the initial value problem (HJ) can be solved by the method of characteristics and a unique solution may be proved to exist for some $T = T(u_0) > 0$. We may then define solution operator $\mathcal{S}_t^\dagger$ with property $u(\cdot,t) = \mathcal{S}_t^\dagger(u_0)$; the maximal time T of existence will, in general, depend on the input u_0 to $\mathcal{S}_t$. The next two subsections describe, respectively, the method of characteristics and the existence of the solution operator on a time-interval $t \in [0,T]$, for all initial data from compact $\mathcal{F}$ in $C^r_{\text{per}}(\Omega)$, for $r \ge 2$. Thus, we define $\{\mathcal{S}_t^\dagger : \mathcal{F} \subset C^r_{\text{per}}(\Omega) \to C^r_{\text{per}}(\Omega)\}$ for all $t \in [0,T]$, $T = T(\mathcal{F})$ sufficiently small.

We recall that throughout this paper, use of a superscript † denotes an object defined through construction of an exact solution of (HJ), or objects used in the construction of the solution; identical notation without a superscript † denotes an approximation of that object.

3.1 *Method of characteristics for (HJ).*

We briefly summarize this methodology; for more details see (Evans, 2010, section 3.3). Consider the following Hamiltonian system for $t \mapsto (q(t), p(t)) \in \Omega \times \mathbb{R}^d$ defined by

$$\begin{cases} \dot{q} = \nabla_p H(q,p), \quad q(0) = q_0, \\ \dot{p} = -\nabla_q H(q,p), \quad p(0) = p_0, \end{cases} \tag{3.1a}$$

$$p_0 = \nabla_q u_0(q_0). \tag{3.1b}$$

If the solution $u(q,t)$ of (HJ) is twice continuously differentiable then u can be evaluated by solving the ordinary differential equation (ODE) (3.1a) with the specific parameterized initial conditions (3.1b). Given these trajectories and $t \geq 0$, the values $u(q(t),t)$ can be computed in terms of the 'action' along this trajectory:

$$u(q(t),t) = u_0(q_0) + \int_0^t \mathcal{L}(q(t),p(t))\,\mathrm{d}\tau, \tag{3.2}$$

where $\mathcal{L} : \Omega \times \mathbb{R}^d \to \mathbb{R}$ is the Lagrangian associated with H, i.e.,

$$\mathcal{L}(q,p) := p \cdot \nabla_p H(q,p) - H(q,p). \tag{3.3}$$

Equivalently, the solution $z(t) = u(q(t),t)$ can be expressed in terms of the solution of the following system of ODEs, $t \mapsto (q(t), p(t), z(t))$:

$$\begin{cases} \dot{q} = \nabla_p H(q,p), \quad q(0) = q_0, \\ \dot{p} = -\nabla_q H(q,p), \quad p(0) = p_0, \\ \dot{z} = \mathcal{L}(q,p), \quad z(0) = z_0. \end{cases} \tag{3.4a}$$

$$p_0 = \nabla_q u_0(q_0), \quad z_0 = u_0(q_0). \tag{3.4b}$$

The system of ODEs (3.4) is defined on $\Omega \times \mathbb{R}^d \times \mathbb{R}$, with a 2π-periodic spatial domain $\Omega = [0, 2\pi]^d$. It can be shown that

$$p(t) \equiv \nabla_q u(q(t),t), \quad \text{for } t \geq 0, \tag{3.5}$$

tracks the evolution of the gradient of u along this trajectory. To ensure the existence of solutions to (3.1), i.e., to avoid trajectories escaping to infinity, we make the following assumption on H, in which $|\cdot|$ denotes the Euclidean distance on $\mathbb{R}^d$:

ASSUMPTION 3.1 (Growth at Infinity). There exists $L_H > 0$, such that

$$\sup_{q\in\Omega}\left\{-p\cdot\nabla_q H(q,p)\right\} \le L_H(1+|p|^2), \tag{3.6}$$

for all $p \in \mathbb{R}^d$. ◊

In the following text we will denote by

$$\begin{cases} \Psi_t^\dagger : \Omega\times\mathbb{R}^d\times\mathbb{R}\to\Omega\times\mathbb{R}^d\times\mathbb{R}, \\ \qquad (q_0,p_0,z_0)\mapsto (q(t),p(t),z(t)), \end{cases} \tag{3.7}$$

the semigroup (flow map) generated by the system of ODEs (3.4a); existence of this semigroup, and hence of the solution operator $\mathcal{S}_t^\dagger$, is the topic of the next subsection.

3.2 *Short-time existence of C^r-solutions*

The goal of this section is to show that for any $r \ge 2$, and for any compact subset $\mathcal{F} \subset C^r_{\text{per}}(\Omega)$, there exists a maximal $T^* = T^*(\mathcal{F}) > 0$, such that for any $u_0 \in \mathcal{F}$ there exists a solution $u : \Omega\times[0,T^*) \to \mathbb{R}$ of (HJ), with property $q \mapsto u(q,t)$ belongs to $C^r_{\text{per}}(\Omega)$ for all $t \in [0,T^*)$. Thus, we may take any $T < T^*$ in (HJ), for data in $\mathcal{F}$. Our proof of this fact relies on the Banach space version of the implicit function theorem; see Appendix B, Theorem B1, for a precise statement.

In the following text, given $t \ge 0$ and given initial values $(q_0,p_0) \in \Omega\times\mathbb{R}^d$, we define $q_t(q_0,p_0)$, $p_t(q_0,p_0)$ as the spatial- and momenta-components of the solution of the Hamiltonian ODE (3.1a), respectively; i.e., the solution of (3.1a) is given by $t \mapsto (q_t(q_0,p_0),p_t(q_0,p_0))$ for any initial data $(q_0,p_0) \in \Omega\times\mathbb{R}^d$.

PROPOSITION 3.2 (Short-time existence of classical solutions). Let Assumption 3.1 hold, let $r \ge 2$ and assume that the Hamiltonian $H \in C^{r+1}_{\text{per}}(\Omega\times\mathbb{R}^d)$. Then, for any compact subset $\mathcal{F} \subset C^r_{\text{per}}(\Omega)$, there exists $T^* = T^*(\mathcal{F}) > 0$ such that for any $u_0 \in \mathcal{F}$ there exists a classical solution $u \in C^r_{\text{per}}(\Omega\times[0,T^*))$ of the Hamilton–Jacobi equation (HJ). Furthermore, for any $T < T^*$, there exists a constant $C = C(T,r,\mathcal{F},H) > 0$ such that $\sup_{t\in[0,T]}\|u(\cdot,t)\|_{C^r} \le C$. ◊

The proof, which may be found in Appendix B, uses the following two lemmas, also proved in Appendix B. The first result shows that, under Assumption 3.1, the semigroup $\Psi_t^\dagger$ in (3.7) is well defined, globally in time.

LEMMA 3.3. Let $H \in C^r_{\text{per}}(\Omega\times\mathbb{R}^d)$ for $r \ge 1$. Let Assumption 3.1 hold. Then, the mapping $\Psi_t^\dagger$ given by (3.7) exists for any $t \ge 0$ and $t \mapsto \Psi_t^\dagger$ defines a semigroup. In particular, for any $(q_0,p_0) \in \Omega\times\mathbb{R}^d$, there exists a solution $t \mapsto (q(t),p(t))$ of the ODE system (3.1a) with initial data $q(0) = q_0$, $p(0) = p_0$, for all $t \ge 0$. ◊

The following result will be used to show that the method of characteristics can be used to construct solutions, at least over a sufficiently short time-interval.

LEMMA 3.4. Let Assumption 3.1 hold, let $r \ge 2$ and assume that the Hamiltonian $H \in C^{r+1}_{\text{per}}(\Omega\times\mathbb{R}^d)$. Let $\mathcal{F} \subset C^r_{\text{per}}(\Omega)$ be compact. Then, there exists a (maximal) time $T^* = T^*(\mathcal{F}) > 0$, such that for all

$u_0 \in \mathcal{F}$, the *spatial characteristic mapping*

$$\Phi_t^\dagger(\,\cdot\,;u_0) : \Omega \to \Omega, \quad q_0 \mapsto q_t\big(q_0, \nabla_q u_0(q_0)\big),$$

defined by (3.1) on the periodic domain $\Omega = [0, 2\pi]^d$, is a C^{r-1}-diffeomorphism for any $t \in [0, T^*)$. $\Diamond$

Note that the map $\Phi_t^\dagger(\,\cdot\,;u_0)$ is defined by the semigroup $\Psi_t^\dagger$ for (q, p, z); however it only requires solution of the Hamiltonian system (3.1) for (q, p). In contrast to the flowmap $\Psi_t^\dagger$ the spatial characteristic map $\Phi_t^\dagger(\,\cdot\,;u_0)$ is in general only invertible over a sufficiently short time-interval, leading to a corresponding short-time existence result for the solution operator $\mathcal{S}_t^\dagger$ associated with (HJ) in Proposition 3.2.

3.3 *Limitations of the setting considered in this work*

In this paper we study the curse of parametric complexity for operator learning, and discuss settings where it can be overcome. The primary role of our study of HJ equations is to highlight a setting where operator learning does not suffer from this curse. However, our analysis is restricted to the setting of smooth solutions and nonintersecting characteristics. We close this section on HJ equations by highlighting this important limitation of this work, and mention a possible extension.

The main limitation of this work is the assumption that classical solutions of (HJ) exist. Indeed, it is well known, e.g., Albano & Cannarsa (2002); Albano *et al.* (2020), that classical solutions of the HJ equations can develop singularities in finite time due to the potential crossing of the spatial characteristics $q(t)$ emanating from different points; indeed, if two spatial characteristics emanating from q_0 and q_0' cross in finite time then (3.2) does not lead to a well-defined function $u(q, t)$, and the method of characteristics breaks down. Thus, our existence theorem is generally restricted to a finite time interval $[0, T^*]$.

In certain special cases the crossing of characteristics is ruled out, and classical solutions exist for all time, i.e., with $T^* = \infty$. One such example is the advection equation with velocity field $v(q)$, and corresponding Hamiltonian $H(q, \nabla_q u) = v(q) \cdot \nabla_q u$, for which the complexity of operator approximation is studied computationally in de Hoop *et al.* (2022).

Finding more general conditions for the existence of classical solutions is a delicate question, which is discussed in Barron *et al.* (1999); under certain convexity assumptions on the Hamiltonian it is possible to relate the existence of a classical solution of (HJ) to its time-reversibility. For example, it is shown in (Barron *et al.*, 1999, theorem 2.5) that if $H(q, p)$ is (i) strictly convex and superlinear in p, and (ii) Lipschitz in q, and $u(x, t)$ is a locally Lipschitz continuous (viscosity) solution of (HJ) both forward and backward in time then u must be smooth. Under certain technical assumptions on the convexity of H and the initial data $u(t = 0)$ it can in fact be shown (Barron *et al.*, 1999) that a forward solution $u(x, t)$ and a backward solution $w(x, t)$ of (HJ) must be equal everywhere, provided that they coincide at the endpoints, i.e., $u(x, T) = w(x, T)$ and $u(x, 0) = w(x, 0) \Rightarrow u \equiv w$ and $u, w \in C^1([0, T] \times \mathbb{R}^d)$.

The existence of classical characteristics will play a fundamental role in the neural network-based approach to be developed in the next section. An interesting question for future work is whether classical characteristics could be replaced by a suitable notion of generalized characteristics, as studied in Albano & Cannarsa (2002); Albano *et al.* (2020). The main challenge in such an extension is the nonuniqueness of generalized characteristics; it is not immediately clear whether a neural network-based approach can coherently approximate such generalized characteristics when there is *a priori* no flow map $(q_t, p_t) = \phi_t(q_0, p_0)$ in the classical sense. In contrast, if such a flow map exists then this represents sufficient structure to beat the curse of parametric complexity, as shown in the following section.

4. HJ neural operator: HJ-Net

In this section we will describe an operator learning framework to approximate the solution operator $\mathcal{S}_t^\dagger$ defined by (HJ) for initial data $u_0 \in C^r_{\text{per}}(\Omega)$ with $r \geq 2$. The main motivation for our choice of framework is the observation that the flow map $\Psi_t^\dagger$ associated with the system of ODEs (3.4a) can be computed *independently of the underlying solution u*. Hence, an operator learning framework for (HJ) can be constructed based on a suitable approximation $\Psi_t \approx \Psi_t^\dagger$, where $\Psi_t : \Omega \times \mathbb{R}^d \times \mathbb{R} \to \Omega \times \mathbb{R}^d \times \mathbb{R}$ is a neural network approximation of the flow $\Psi_t^\dagger$. Given such Ψ_t, and upon fixing evaluation points $\{q_0^j\}_{j=1}^N \subset \Omega$, we propose to approximate the forward operator $\mathcal{S}_t^\dagger$ of the Hamilton–Jacobi equation (HJ) by $\mathcal{S}_t$, using the following three steps:

Step (a) encode the initial data $u_0 \in C^r_{\text{per}}(\Omega)$ by evaluating it at the points q_0^j:

$$\mathcal{E} : C^r_{\text{per}}(\Omega) \to [\Omega \times \mathbb{R}^d \times \mathbb{R}]^N,$$
$$u_0 \mapsto \{(q_0^j, p_0^j, z_0^j)\}_{j=1}^N,$$

with $(q_0^j, p_0^j, z_0^j) := (q_0^j, \nabla_q u_0(q_0^j), u_0(q_0^j))$;

Step (b) for each $j = 1, \ldots, N$, apply the approximate flow $\Psi_t : \Omega \times \mathbb{R}^d \times \mathbb{R}$ to the encoded data, resulting in a map

$$\Psi_t^N : [\Omega \times \mathbb{R}^d \times \mathbb{R}]^N \to [\Omega \times \mathbb{R}^d \times \mathbb{R}]^N,$$
$$\{(q_0^j, p_0^j, z_0^j)\}_{j=1}^N \mapsto \{(q_t^j, p_t^j, z_t^j)\}_{j=1}^N,$$

where $(q_t^j, p_t^j, z_t^j) := \Psi_t(q_0^j, p_0^j, z_0^j)$, for $j = 1, \ldots, N$;

Step (c) approximate the underlying solution at a fixed time $t \in [0, T]$, for T sufficiently small, by interpolating the data (input/output pairs) $\{(q_t^j, z_t^j)\}_{j=1}^N$, leading to a reconstruction map:

$$\mathcal{R} : [\Omega \times \mathbb{R}]^N \to C^r(\Omega),$$
$$\{(q_t^j, z_t^j)\} \mapsto s_{z,Q}.$$

If we let $\mathcal{P}$ denote the projection map, which takes $[\Omega \times \mathbb{R}^d \times \mathbb{R}]^N$ to $[\Omega \times \mathbb{R}]^N$, then, for fixed T sufficiently small, we have defined an approximation of $\mathcal{S}_t^\dagger : C^r_{\text{per}}(\Omega) \to C^r_{\text{per}}(\Omega)$, denoted $\mathcal{S}_t : C^r_{\text{per}}(\Omega) \to C^r(\Omega)$, and defined by

$$\mathcal{S}_t = \mathcal{R} \circ \mathcal{P} \circ \Psi_t^N \circ \mathcal{E}. \tag{4.1}$$

It is a consequence of Proposition 3.2 that our approximation $\mathcal{S}_t$ is well defined for all inputs u_0 from compact subset $\mathcal{F} \subset C^r_{\text{per}}(\Omega)$, $r \geq 2$, in some interval $t \in [0, T]$, for T sufficiently small. However, the resulting approximation does not obey the semigroup property with respect to t and should be interpreted as holding for a fixed $t \in [0, T]$, T sufficiently small. Furthermore, iterating the map obtained for any such

fixed t is not in general possible unless $\mathcal{S}_t$ maps $\mathcal{F}$ into itself. The requirement that $\mathcal{S}_t^\dagger$ maps $\mathcal{F}$ into itself would also be required to prove the existence of a semigroup for (HJ); for our operator approximator $\mathcal{S}_t$ we would additionally need to ensure periodicity of the reconstruction step, something we do not address in this paper.

If the underlying solution $u(q,t)$ of (HJ) exists up to time t, and if it is C^2 then the method of characteristics can be applied, and the above procedure would reproduce the underlying solution, in the absence of approximation errors in Steps (b) and (c), i.e., in the absence of approximation errors of the Hamiltonian flow $\Psi_t \approx \Psi_t^\dagger$, and in the absence of reconstruction errors. We will study the effect of approximating Step (b) by use of an ReLU neural network approximation of the flow $\Psi_t^\dagger$ and by use of a moving least squares interpolation for Step (c). In the following two subsections we define these two approximation steps, noting that doing so leads to a complete specification of $\mathcal{S}_t$. This complete specification is summarized in the final Subsection 4.3.

4.1 *Step (b) ReLU network*

We seek an approximation Ψ_t to $\Psi_t^\dagger$ in the form of an ReLU neural network (2.1), as summarized in Section 2.1.1, with input and output dimensions $D_{\mathcal{X}} = D_{\mathcal{Y}} = 2d+1$, and taking the concatenated input $x := (q_0, p_0, z_0) \in \Omega \times \mathbb{R}^d \times \mathbb{R}$ to its image in $\mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$. We use 2π-periodicity to identify the output in the first d components (the spatial variable q) with a unique element in $\Omega = [0, 2\pi]^d$. With slight abuse of notation this results in a well-defined mapping

$$\Psi_t : \Omega \times \mathbb{R}^d \times \mathbb{R} \to \Omega \times \mathbb{R}^d \times \mathbb{R}, \quad (q_0, p_0, z_0) \to \Psi_t(q_0, p_0, z_0).$$

Let μ denote a probability measure on $\Omega \times \mathbb{R}^d \times \mathbb{R}$. We would like to choose θ to minimize the loss function

$$L(\theta) = \mathbb{E}^{(q,p,z)\sim\mu} \|\Psi_t(q,p,z;\theta) - \Psi_t^\dagger(q,p,z)\|.$$

In practice, we achieve this by an empirical approximation of μ, based on N i.i.d. samples, and numerical simulation to approximate the evaluation of $\Psi_t^\dagger(\cdot)$ on these samples. The resulting approximate empirical loss can be approximately minimized by stochastic gradient descent (Robbins & Monro, 1951; Goodfellow *et al.*, 2016), for example.

4.2 *Step (c) moving least squares*

In this section we define the interpolation mapping $\mathcal{R} : \{q_t^j, z_t^j\}_{j=1}^N \mapsto u(\cdot, t)$, employing reconstruction by moving least squares (Wendland, 2004). In general, given a function $f^\dagger : \Omega \to \mathbb{R}$, here assumed to be defined on the domain $\Omega = [0, 2\pi]^d$, and given a set of (scattered) data $\{\mathfrak{q}^j, z^j\}_{j=1}^N$, where $z^j = f^\dagger(\mathfrak{q}^j)$ for $\mathfrak{Q} = \{\mathfrak{q}^1, \ldots, \mathfrak{q}^N\} \subset \Omega$, the method of moving least squares produces an approximation $f_{z,\mathfrak{Q}} \approx f^\dagger$, which is given by the following minimization (Wendland, 2004, definition 4.1):

$$f_{z,\mathfrak{Q},n}(q) = \min \left\{ \sum_{j=1}^N \left[z^j - P(\mathfrak{q}^j) \right]^2 \phi_\delta(q - \mathfrak{q}^j) \,\middle|\, P \in \pi_n \right\}. \tag{4.2}$$

Here, π_n denotes the space of polynomials of degree n, and $\phi_\delta(q) = \phi(q/\delta)$ is a compactly supported, non-negative weight function. In the following text we will choose $n := r - 1$, where r is the degree of smoothness of $f^\dagger \in C^r(\Omega)$. We will assume $\phi(\cdot)$ to be smooth, supported in the unit ball $B_1(0)$, and positive on the ball $B_{1/2}(0)$ with radius $1/2$.

The approximation error incurred by moving least squares can be estimated in the L^∞-norm, in terms of suitable measures of the 'density' of the scattered data points $\mathfrak{q}^j$, and the smoothness of the underlying function $f^\dagger$ (Wendland, 2004). The relevant notions are defined next, in which $|\cdot|$ denotes the Euclidean distance on $\Omega \subset \mathbb{R}^d$.

DEFINITION 4.1. The **fill distance** of a set of points $\mathfrak{Q} = \{\mathfrak{q}^1, \ldots, \mathfrak{q}^N\} \subset \Omega$ for a bounded domain $\Omega \subset \mathbb{R}^d$ is defined to be

$$h_{\mathfrak{Q},\Omega} := \sup_{q\in\Omega} \min_{j=1,\ldots,N} |q - \mathfrak{q}^j|.$$

The **separation distance** of $\mathfrak{Q}$ is defined by

$$\rho_{\mathfrak{Q}} := \frac{1}{2} \min_{k\neq j} |\mathfrak{q}^k - \mathfrak{q}^j|.$$

A set $\mathfrak{Q}$ of points in Ω is said to be **quasi-uniform** with respect to $\kappa \geq 1$, if

$$\rho_{\mathfrak{Q}} \leq h_{\mathfrak{Q},\Omega} \leq \kappa \rho_{\mathfrak{Q}}.$$

$\Diamond$

Combining the approximation error of moving least squares with a stability result for moving least squares, with respect to the input data, leads to the error estimate that we require to estimate the error in our proposed HJ-Net. Proof of the following may be found in Appendix C. The statement involves both the input data set Q and a set $Q^\dagger$, which Q is supposed to approximate, together with their respective fill-distances and separation distances.

PROPOSITION 4.2 (Error stability). Let $\Omega = [0, 2\pi]^d \subset \mathbb{R}^d$ and consider function $f^\dagger \in C^r(\Omega)$, for some fixed regularity parameter $r \geq 2$. Assume that $Q^\dagger = \{q^{j,\dagger}\}_{j=1}^N$ is quasi-uniform with respect to $\kappa \geq 1$. Let $\{q^j, z^j\}_{j=1}^N$ be approximate interpolation data, and define $Q := \{q^j\}_{j=1}^N$, where, for some $\rho \in (0, \frac{1}{2}\rho_{Q^\dagger})$ and $\epsilon > 0$, we have

$$|q^j - q^{j,\dagger}| < \rho, \quad |z^j - f(q^{j,\dagger})| < \epsilon.$$

Using this approximate interpolation data let $f_{z,Q} := f_{z,Q,n}$ be obtained by moving least squares (4.2) with $n := r - 1$. Then, there exist constants $h_0, \gamma, C > 0$, depending only on d, r and $\kappa \geq 1$, such that, for $h_{Q,\Omega} \leq h_0$ and moving least squares scale parameter $\delta := \gamma h_{Q,\Omega}$, we have

$$\|f^\dagger - f_{z,Q}\|_{L^\infty} \leq C \left(\|f^\dagger\|_{C^r} h^r_{Q^\dagger,\Omega} + \epsilon + \|f^\dagger\|_{C^1} \rho \right). \tag{4.3}$$

$\Diamond$

Algorithm 1 Pruning

Input: Interpolation points $Q = \{q^j\}_{j=1}^N$.

Output: Pruned interpolation points $Q' = \{q^{j_k}\}_{k=1}^m$ with fill distance $h_{Q',\Omega} \leq 3h_{Q,\Omega}$, and Q' is quasi-uniform for $\kappa = 3$, i.e.,

$$\rho_{Q'} \leq h_{Q',\Omega} \leq 3\rho_{Q'}$$

procedure
  Set $m \leftarrow 1, j_1 \leftarrow 1$ and $Q' \leftarrow \{q^1\}$.
  while $m < N$ **do**
    Given $Q' = \{q^{j_1}, \dots, q^{j_m}\} \subset Q$, does there exist $q^k \in Q$, such that

$$B_h(q^k) \cap \bigcup_{\ell=1}^{m} B_h(q^{j_\ell}) = \emptyset?$$

    if Yes **then**
      Set $j_{m+1} \leftarrow k$,
      Set $Q' \leftarrow Q' \cup \{q^k\}$,
      Set $m \leftarrow m + 1$.
    else
      Terminate the algorithm.
    end if
  end while
end procedure

REMARK 4.3. The constants C and γ in the previous proposition can be computed explicitly (Wendland, 2004, see theorem 3.14 and corollary 4.8). ◊

Proposition 4.2 reveals two sources of error in the reconstruction by moving least squares: the first term on the right-hand side of (4.3) is the error due to the discretization by a finite number of evaluation points q^j. This error persists even in a perfect data setting, i.e., when $q^j = q^{j,\dagger}$ and $z^j = f^\dagger(q^{j,\dagger})$. The last two terms in (4.3) reflect the fact that in our intended application to HJ-Net the evaluation points q^j and the function values z^j are only given approximately, via the approximate flow map $\Psi_t \approx \Psi_t^\dagger$, introducing additional error sources.

The proof of Proposition 4.2 relies crucially on the fact that the set of evaluation points $Q = \{q^j\}_{j=1}^N$ is quasi-uniform. In our application to HJ-Net these points are obtained as the image of $(q_0^j, p_0^j, z_0^j) := (q_0^j, \nabla_q u_0(q_0^j), u_0(q_0^j))$ under the approximate flow Ψ_t. In particular, they depend on u_0 and we cannot ensure any *a priori* control on the separation distance ρ_Q. Our proposed reconstruction $\mathcal{R}$ therefore involves the pruning step, stated as Algorithm 1. Lemma C.3 in Appendix C shows that Algorithm 1 produces a quasi-uniform set $Q' \subset Q$ with the desired properties asserted above.

Given the interpolation by moving least squares and the pruning algorithm we finally define the reconstruction map $\mathcal{R} : \{(q_t^j, z_t^j)\}_{j=1}^N \mapsto f_{z,Q}$. In our application the interpolation data are connected

Algorithm 2 Reconstruction $\mathcal{R}$

(1) Given approximate interpolation data $\{q_t^j, z_t^j\}_{j=1}^N$ at data points $Q = \{q_t^1, \ldots, q_t^N\}$, determine a quasi-uniform subset $Q' = \{q_t^{j_k}\}_{k=1}^m \subset Q$ by Algorithm 1.

(2) Given a degree of smoothness $r \in \mathbb{N}_{\geq 2}$, define $f_{z,Q} := f_{z,Q,n=r-1}$ as the moving least squares interpolant (4.2), based on the (pruned) interpolation data $\{(q_t^{j_k}, z_t^{j_k})\}_{k=1}^m$ with $\delta = \gamma h_{Q',\Omega}$, and γ the constant in Proposition 4.2.

with a ground truth map $f^\dagger \in C^r(\Omega)$. The following algorithm assumes knowledge of the degree of smoothness r:

4.3 *Summary of HJ-Net*

Thus, in the final definition of HJ-Net given in equation (4.1) we recall that $\mathcal{E}$ denotes the encoder mapping

$$u_0 \mapsto \mathcal{E}(u_0) := \left(q_0^j, \nabla u_0(q_0^j), u_0(q_0^j)\right).$$

The mapping

$$\left(q_0^j, p_0^j, z_0^j\right) \mapsto \left(q_t^j, p_t^j, z_t^j\right) := \Psi_t\left(q_0^j, p_0^j, z_0^j; \theta\right)$$

approximates the flow Ψ_t for each of the triples, (q_0^j, p_0^j, z_0^j), $j = 1, \ldots, N$, and θ is trained from data to minimize an approximate empirical loss. And, finally, the reconstruction $\mathcal{R}$ is obtained by applying the pruned moving least squares Algorithm 2 to the data $\{q_t^j, p_t^j, z_t^j\}_{j=1}^N$ at scattered data points $Q_t = \{q_t^1, \ldots, q_t^N\}$ and with corresponding values $z_t = (z_t^1, \ldots, z_t^N)$, to approximate the output $u(q,t) \approx f_{z_t,Q_t}(q)$.

5. Error estimates and complexity

Subsection 5.1 contains statement of the main theorem concerning the computational complexity of HJ-Net, and the high level structure of the proof. The theorem demonstrates that the curse of parametric complexity is overcome in this problem in the sense that the cost to achieve a given error, as measured by the size of the parameterization, grows only algebraically with the inverse of the error. In Subsections 5.2 and 5.3 we provide a detailed discussion of the approximation of the Hamiltonian flow and the moving last squares algorithm, which, together, lead to the proof of Theorem 5.1. Proofs of the results stated in those two subsections are given in the appendix.

5.1 *HJ-Net beats the curse of parametric complexity*

THEOREM 5.1 (HJ-Net approximation estimate). Consider equation (HJ) on periodic domain $\Omega = [0, 2\pi]^d$, with C^r_{per} initial data and Hamiltonian $H \in C^{r+1}_{\text{per}}$, where $r \geq 2$. Assume that H satisfies the no-blowup Assumption 3.1. Let $\mathcal{F} \subset C^r_{\text{per}}$ be a compact set of initial data, and let $T < T^*(\mathcal{F})$, where

$T^*(\mathcal{F})$ is given by Proposition 3.2. Then, there is constant $C = C(T, d, r, H, \mathcal{F}) > 0$, such that for any $\epsilon > 0$ and $t \in [0, T]$ there exists a set of points $Q_\epsilon = \{q^1, \dots, q^N\}$, optimal parameter θ_ϵ and neural network $\Psi_t(\cdot) = \Psi_t(\cdot\,; \theta_\epsilon)$ such that the corresponding HJ-Net approximation given by (4.1), with Steps (b) and (c) defined in Subsections 4.1 and 4.2, respectively, satisfies

$$\sup_{u_0 \in \mathcal{F}} \|\mathcal{S}_t(u_0) - \mathcal{S}_t^\dagger(u_0)\|_{L^\infty} \leq \epsilon.$$

Furthermore, we have the following complexity bounds: (i) the number N of encoding points $Q_\epsilon = \{q^j\}_{j=1}^N$ from Step (a) can be bounded by

$$N \leq C\epsilon^{-d/r}; \tag{5.1}$$

and (ii) the neural network $\Psi_t(\cdot) = \Psi_t(\cdot\,; \theta_\epsilon)$ from Step (b), Subsection 4.1, satisfies

$$\text{depth}(\Psi_t) \leq C \log(\epsilon^{-1}), \quad \text{size}(\Psi_t) \leq C\epsilon^{-(2d+1)/r} \log(\epsilon^{-1}). \tag{5.2}$$

♢

Proof. We first note that, for any $u_0 \in \mathcal{F}$ and $T < T^*(\mathcal{F})$, Proposition 3.2 shows that the solution u of (HJ) can be computed by the method of characteristics up to time T. Thus, $\mathcal{S}_t^\dagger$ is well defined for any $t \in [0, T]$. We break the proof into three steps, relying on propositions established in the following subsections, and then conclude in a final fourth step.

Step 1 (Neural network approximation). Let M be given by

$$M := 1 \vee \sup_{u_0 \in \mathcal{F}} \sup_{q \in \Omega} \|\nabla u_0(q)\|_{\ell^\infty} \vee \sup_{u_0 \in \mathcal{F}} \sup_{q \in \Omega} |u_0(q)|,$$

where $a \vee b$ denotes the maximum. By this choice of M we have $\nabla u_0(q) \in [-M, M]^d$, $u_0(q) \in [-M, M]$ for all $u_0 \in \mathcal{F}$, $q \in \Omega$. By Proposition 5.4 there exists a constant $\beta = \beta(d, L_H, t) \geq 1$, and a constant $C > 0$, depending only on M, d, t, and the norm $\|H\|_{C^{r+1}(\mathbb{T}^d \times [-\beta M, \beta M]^d)}$, such that the Hamiltonian flow map $\Psi_t^\dagger$ (3.7) can be approximated by a neural network Ψ_t with

$$\text{size}(\Psi_t) \leq C\epsilon^{-(2d+1)/r} \log(\epsilon^{-1}), \quad \text{depth}(\Psi_t) \leq C \log(\epsilon^{-1}),$$

and

$$\sup_{(q_0, p_0, z_0) \in \Omega \times [-M, M]^d \times [-M, M]} \left| \Psi_t(q_0, p_0, z_0) - \Psi_t^\dagger(q_0, p_0, z_0) \right| \leq \epsilon. \tag{5.3}$$

Step 2 (Choice of encoding points). Fix $\rho > 0$, to be determined below. Let $Q := \rho\mathbb{Z}^d \cap [0, 2\pi]^d$ denote an equidistant grid on $[0, 2\pi]^d$ with grid spacing ρ. Enumerating the elements of Q we write $Q = \{q_0^1, \dots, q_0^N\}$, where we note that there exists a constant $C_1 = C_1(d) > 0$ depending only on d, such

that $N \leq C_1 \rho^{-d}$; equivalently,

$$\frac{\rho^d}{C_1} \leq \frac{1}{N}. \tag{5.4}$$

For any $u_0 \in \mathcal{F}$ let

$$Q_{u_0}^\dagger := \left\{ q_t^{j,\dagger} \,\middle|\, q_t^{j,\dagger} = q_t(q_0^j, p_0^j),\ q_0^j \in Q_\epsilon,\ p_0^j = \nabla_q u_0(q_0^j) \right\} \tag{5.5}$$

be the set of image points under the characteristic mapping defined by u_0. Since $Q_{u_0}^\dagger = \{q_t^{j,\dagger}\}_{j=1}^N$ is a set of N points it follows from the definition of the fill distance that N balls of radius $h_{Q_{u_0}^\dagger,\Omega}$ cover $\Omega = [0, 2\pi]^d$. A simple volume counting argument then implies that there exists a constant $C_0 = C_0(d) > 0$, such that $1/N \leq C_0\, h^d_{Q_{u_0}^\dagger,\Omega}$; equivalently,

$$\frac{1}{C_0 N} \leq h^d_{Q_{u_0}^\dagger,\Omega}, \quad \forall\, u_0 \in \mathcal{F}. \tag{5.6}$$

Given ϵ from Step 1 we now choose $\rho := (C_0 C_1)^{1/d} \epsilon^{1/r}$, so that by (5.4) and (5.6),

$$\epsilon^{d/r} = \frac{1}{C_0}\frac{\rho^d}{C_1} \leq \frac{1}{C_0 N} \leq h^d_{Q_{u_0}^\dagger,\Omega}, \quad \forall\, u_0 \in \mathcal{F}.$$

We emphasize that C_0, C_1 depend only on d, and are independent of $u_0 \in \mathcal{F}$ and ϵ. We have thus shown that if $Q_\epsilon := \{q^1, \ldots, q^N\}$ is an enumeration of $\rho\mathbb{Z}^d \cap [0, 2\pi]^d$ with $\rho := (C_0 C_1)^{1/d} \epsilon^{1/r}$ then the fill distance of the image points $Q_{u_0}^\dagger$ under the characteristic mapping satisfies

$$\epsilon \leq h^r_{Q_{u_0}^\dagger,\Omega}, \quad \forall\, u_0 \in \mathcal{F}. \tag{5.7}$$

In particular, this step defines our encoding points Q_ϵ.

Step 3 (Interpolation error estimate). Let Q_ϵ be the set of evaluation points determined in Step 2. Since Q_ϵ is an equidistant grid with grid spacing proportional to $\epsilon^{1/r}$ the fill distance of Q_ϵ is bounded by $h_{Q_\epsilon,\Omega} \leq C\epsilon^{1/r}$, where the constant $C = C(d) \geq 1$ depends only on d. By Proposition 5.7 there exists a (possibly larger) constant $C = C(d, t, H, \mathcal{F}) \geq 1$, such that for all $u_0 \in \mathcal{F}$ the set of image points $Q_{u_0}^\dagger$ under the characteristic mapping (5.5) has uniformly bounded fill distance

$$h_{Q_{u_0}^\dagger,\Omega} \leq C\epsilon^{1/r}, \quad \forall\, u_0 \in \mathcal{F}. \tag{5.8}$$

Furthermore, taking into account (5.7) the upper bound (5.3) implies that

$$\sup_{(q_0,p_0,z_0)\in\mathbb{T}^d\times[-M,M]^d\times[-M,M]} \left| \Psi_t(q_0, p_0, z_0) - \Psi_t^\dagger(q_0, p_0, z_0) \right| \leq h^r_{Q_{u_0}^\dagger,\Omega},$$

for any $u_0 \in \mathcal{F}$. In turn, this shows that the approximate interpolation data $(q_t^j, z_t^j) = \mathcal{P} \circ \Psi_t(q_0^j, p_0^j, z_0^j)$, $j = 1, \dots, N$, obtained from the neural network approximation $\Psi_t \approx \Psi_t^\dagger$ by (orthogonal) projection $\mathcal{P}$ onto the first and last components, satisfy

$$\left| q_t^j - q_t^{j,\dagger} \right| \leq h^r_{Q^\dagger_{u_0},\Omega}, \quad \left| z_t^j - u(q_t^{j,\dagger}, t) \right| \leq h^r_{Q^\dagger_{u_0},\Omega},$$

where $u(q,t)$ denotes the exact solution of the HJ PDE (HJ) with initial data $u_0 \in \mathcal{F}$. By Proposition 5.6 there exists a constant $C > 0$, depending only on d and r, such that the pruned moving least squares approximant $f_{z,Q_{u_0}}$ obtained by Algorithm 2 with (approximate) data points $Q_{u_0} = \{q_t^1, \dots, q_t^N\}$ and corresponding data values $z = \{z_t^1, \dots, z_t^N\}$ satisfies

$$\|u(\cdot, t) - f_{z,Q_{u_0}}\|_{L^\infty(\Omega)} \leq C \left(1 + \|u(\cdot, t)\|_{C^r(\Omega)}\right) h^r_{Q^\dagger_{u_0},\Omega}. \tag{5.9}$$

Step 4 (Conclusion). By the short-time existence result of Proposition 3.2 there exists $C = C(H, \mathcal{F}, t) > 0$, such that $\|u(\cdot, t)\|_{C^r(\Omega)} \leq C$ for any solution u of the Hamilton–Jacobi equation (HJ) with initial data $u(\cdot, t = 0) = u_0 \in \mathcal{F}$. By definition of the HJ-Net approximation we have $\mathcal{S}_t(u_0) \equiv f_{z,Q_{u_0}}$ and by definition of the solution operator $\mathcal{S}_t^\dagger(u_0) \equiv u(\cdot, t)$. We thus conclude that

$$\|\mathcal{S}_t(u_0) - \mathcal{S}_t^\dagger(u_0)\|_{L^\infty} \overset{(5.9)}{\leq} C h^r_{Q^\dagger_{u_0},\Omega} \overset{(5.8)}{\leq} C\epsilon,$$

for a constant $C = C(T, d, r, H, \mathcal{F}) > 0$, independent of ϵ. Since ϵ is arbitrary replacing ϵ by ϵ/C throughout the above argument implies the claimed error and complexity estimate of Theorem 5.1. □

Remark 5.2 (Nonuniform sampling). In practice, an input-function-dependent, *nonuniform* sampling could improve the results of the moving least squares interpolant. This is especially relevant, given the potentially rapid growth of second- and higher order derivatives along characteristics. However, this would likely not lead to faster asymptotic convergence rates, motivating us to restrict attention to the uniformly sampled setting for our analysis, for simplicity. ♢

Remark 5.3 (Overall computational complexity of HJ-Net). The error and complexity estimate of Theorem 5.1 implies that for moderate dimensions d, and for sufficiently smooth input functions $u_0 \in \mathcal{F} \subset C^r_{\text{per}}$, with $r > 3d+1$, the overall complexity of this approach scales at most linearly in ϵ^{-1}: Indeed, the mapping of the data points $(q_0^j, p_0^j, z_0^j) \mapsto (q_t^j, p_t^j, z_t^j)$ for $j = 1, \dots, N$ requires N forward-passes of the neural network Ψ_t, which has $O(\epsilon^{-(2d+1)/r} \log(\epsilon^{-1}))$ internal degrees of freedom. Since $N = O(\epsilon^{-d/r})$ the computational complexity of this mapping is thus bounded by $O(\epsilon^{-(3d+1)/r} \log(\epsilon^{-1})) = O(\epsilon^{-1})$. A naive implementation of the pruning algorithm requires at most $O(N^2) = O(\epsilon^{-2d/r}) = O(\epsilon^{-1})$ comparisons. The computational complexity of the reconstruction by the moving least squares method with $m \leq N$ (pruned) interpolation points and with $M \sim \epsilon^{-d/r}$ evaluation points can be bounded by $O(m + M) = O(\epsilon^{-d/r} + M) = O(\epsilon^{-1})$ (Wendland, 2004, last paragraph of chapter 4.2). In particular, the overall complexity to obtain an ϵ-approximation of the output function $\mathcal{S}_t(u_0) \approx \mathcal{S}_t^\dagger(u_0)$ at e.g., the encoding points Q_ϵ (with $M = |Q_\epsilon| \sim \epsilon^{-d/r}$) is at most linear in ϵ^{-1}. ♢

Theorem 5.1 shows that for fixed d and r the proposed HJ-Net architecture can overcome the general curse of parametric complexity in the operator approximation $\mathcal{S} \approx \mathcal{S}^\dagger$ implied by Theorem 2.11 even though the underlying operator does not have higher than C^r-regularity. This is possible because HJ-Net leverages additional structure inherent to the HJ PDE HJ (reflected in the method of characteristics), and therefore does not rely solely on the C^r-smoothness of the underlying operator $\mathcal{S}^\dagger$.

5.2 *Approximation of Hamiltonian flow and quadrature map*

In this subsection we quantify the complexity of ϵ-approximation by an ReLU network as defined in Subsection 4.1.

Recall that $\Psi_t^\dagger : \Omega \times \mathbb{R}^d \times \mathbb{R} \to \Omega \times \mathbb{R}^d \times \mathbb{R}$ comprises solution of the Hamiltonian equations (3.1) and quadrature (3.2), leading to (3.4). An approximation $\Psi_t : \Omega \times \mathbb{R}^d \times \mathbb{R} \to \Omega \times \mathbb{R}^d \times \mathbb{R}$ of this Hamiltonian flow map is provided the by the following proposition, proved in Appendix D.

PROPOSITION 5.4. Let $\Omega = [0, 2\pi]^d$. Let $r \geq 2$, and $t > 0$ be given, and assume that $H \in C^{r+1}_{\text{per}}(\Omega \times \mathbb{R}^d)$ satisfies the no-blowup Assumption 3.1 with constant $L_H > 0$. Then, for any $M \geq 1$, there exist constants $\beta := (1 + \sqrt{d}) \exp(L_H t)$ and $C = C\left(\|H\|_{C^{r+1}(\Omega \times [-\beta M, \beta M]^d)}, M, r, d, t\right) > 0$, such that for all $\epsilon \in (0, \frac{1}{2}]$ there is an ReLU neural network $\Psi_t(\cdot) = \Psi_t(\cdot\,; \theta_\epsilon)$ satisfying

$$\sup_{(q_0, p_0, z_0) \in \Omega \times [-M, M]^{d+1}} \left| \Psi_t(q_0, p_0, z_0) - \Psi_t^\dagger(q_0, p_0, z_0) \right| \leq \epsilon, \tag{5.10}$$

satisfying

$$\text{size}(\Psi_t) \leq C \epsilon^{-(2d+1)/r} \log\left(\epsilon^{-1}\right), \quad \text{depth}(\Psi_t) \leq C \log\left(\epsilon^{-1}\right).$$

♢

REMARK 5.5. Using the results of Yarotsky & Zhevnerchuk (2020); Kohler & Langer (2021); Lu *et al.* (2021a) one could in fact improve the size bound of Proposition 5.4 somewhat: neglecting logarithmic terms in ϵ^{-1} it can be shown that a neural network with $\text{size}(\Psi_t) \lesssim \epsilon^{-(d+1/2)/r}$ suffices. However, this comes at the expense of substantially increasing the depth from a logarithmic scaling $\text{depth}(\Psi_t) \lesssim \log(\epsilon^{-1})$, to an algebraic scaling $\text{depth}(\Psi_t) \lesssim \epsilon^{-(d+1/2)/r}$. ♢

5.3 *Moving least squares reconstruction error*

In this subsection we discuss error estimates for the reconstruction by moving least squares, based on imperfect input–output pairs, as defined in Subsection 4.2.

We recall that the reconstruction $\mathcal{R}$ in the HJ-Net approximation is obtained by applying the pruned moving least squares Algorithm 2 to the data $\{q_t^j, z_t^j\}_{j=1}^N$, where (q_t^j, p_t^j, z_t^j) are obtained as $(q_t^j, p_t^j, z_t^j) = \Psi_t(q_0^j, p_0^j, z_0^j)$ with fixed evaluation points $\{q_0^j\}_{j=1}^N \subset \Omega$, and where $p_0^j := \nabla_q u_0(q_0^j)$, $z_0^j := u_0(q_0^j)$ are determined in terms of a given input function u_0, so that $z_t^j \approx u(q_t^j, t)$ is an approximation of the corresponding solution $u(\cdot\,, t)$ at time t.

In the following text we first derive an error estimate in terms of the fill distance of $Q_t = \{q_t^j\}_{j=1}^N$, in Proposition 5.6. Subsequently, in Proposition 5.7, we provide a bound on the fill distance $h_{Q_t, \Omega}$ at time t in terms of the fill distance $h_{Q,\Omega}$ at time 0. Proof of both propositions can be found in Appendix D.

PROPOSITION 5.6. Let $\Omega = [0, 2\pi]^d \subset \mathbb{R}^d$ and fix a regularity parameter $r \geq 2$. There exist constants $h_0, C > 0$ such that the following holds: assume that $Q^\dagger = \{q^{1,\dagger}, \ldots, q^{N,\dagger}\} \subset \Omega$ is a set of N evaluation points with fill distance $h_{Q^\dagger,\Omega} \leq h_0$. Then, for any 2π-periodic function $f^\dagger \in C^r_{\text{per}}(\Omega)$, and approximate input–output data $\{(q^j, z^j)\}_{j=1}^N$, such that

$$|q^j - q^{j,\dagger}| \leq h^r_{Q^\dagger,\Omega}, \quad |z^j - f^\dagger(q^{j,\dagger})| \leq h^r_{Q^\dagger,\Omega},$$

the pruned moving least squares approximant $f_{z,Q} := f_{z,Q,n=r-1}$ of Algorithm 2 with interpolation data points $Q = \{q^1, \ldots, q^N\}$ and data values $z = \{z^1, \ldots, z^N\}$ satisfies

$$\|f^\dagger - f_{z,Q}\|_{L^\infty(\Omega)} \leq C \left(1 + \|f^\dagger\|_{C^r(\Omega)}\right) h^r_{Q^\dagger,\Omega}.$$

◊

In contrast to Proposition 4.2, Proposition 5.6 includes the pruning step in the reconstruction, and does not assume quasi-uniformity of either the underlying exact point distribution $Q^\dagger$, or the approximate point distribution Q. To obtain a bound on the reconstruction error we can combine the preservation of C^r-regularity implied by the short-time existence Proposition 3.2, with the following:

PROPOSITION 5.7. Let $\Omega = [0, 2\pi]^d$, let $r \geq 2$ and assume that the Hamiltonian $H \in C^{r+1}_{\text{per}}(\Omega \times \mathbb{R}^d)$ is periodic in q and satisfies Assumption 3.1 with constant $L_H > 0$. Let $\mathcal{F} \subset C^r_{\text{per}}(\Omega)$ be a compact subset and fix $t < T^*$. Then, there exists a constant $C = C(H, \mathcal{F}, L_H, t) > 0$, such that for any set of evaluation points $Q = \{q_0^j\}_{j=1}^N \subset \Omega$, and for any $u_0 \in \mathcal{F}$ the image points

$$Q^\dagger_{u_0} := \left\{ q_t^j \,\middle|\, q_t^j = \Phi^\dagger_{t,u_0}(q_0^j),\ j = 1, \ldots, N \right\} \subset \Omega,$$

under the spatial characteristic mapping $\Phi^\dagger_{t,u_0} : q_0 \mapsto q_t(q_0, \nabla_q u_0(q_0))$, satisfy the following uniform bound on the fill distance:

$$h_{Q^\dagger_{u_0},\Omega} \leq C h_{Q,\Omega}.$$

◊

6. Conclusions

The first contribution of this work is to study the curse of dimensionality in the context of operator learning, here interpreted in terms of the infinite-dimensional nature of the input space. We state a theorem that, for the first time, establishes a general form of the curse of parametric complexity, a natural scaling limit of the CoD in high-dimensional approximation, characterized by lower bounds that are *exponential* in the desired error. The theorem demonstrates that in general it is not possible to obtain complexity estimates, for the size of the approximating neural operator, that grow algebraically with inverse error unless specific structure in the underlying operator is leveraged; in particular, we prove that this additional structure has to go beyond C^r- or Lipschitz-regularity. This considerably generalizes and strengthens earlier work on the curse of parametric complexity in Lanthaler (2023),

where a mild form of this curse had been identified for PCA-Net. As shown in this work our result applies to many proposed operator learning architectures including PCA-Net, the FNO and DeepONet, and recent nonlinear extensions thereof. The lower complexity bound in this work is obtained for neural operator architectures based on standard feedforward ReLU neural networks, and could likely be extended to feedforward architectures employing piecewise polynomial activations. It is of note that algebraic complexity bounds, which seemingly overcome the curse of parametric complexity of this work, have recently been derived for the approximation of Lipschitz operators (Schwab *et al.*, 2023); these results build on nonstandard neural network architectures with either superexpressive activation functions, or nonstandard connectivity, and therefore do not contradict our results. In fact, a recent follow-up (Lanthaler, 2024) to this work sheds further light on this question from an information-theoretic perspective, resulting in a statement of the curse of parametric complexity, which is independent of the activation function.

The second contribution of this paper is to present an operator learning framework for HJ equations, and to provide a complexity analysis demonstrating that the methodology is able to tame the curse of parametric complexity for these PDE. We present the ideas in a simple setting, and there are numerous avenues for future investigation. For example, as pointed out in Subsection 3.3, one main limitation of the proposed approach based on characteristics is that we can only consider finite time intervals on which classical solutions of the HJ equations exist. As already pointed out, there it would be of interest to extend the methodology to viscosity solutions, after the potential formation of singularities. It would also be of interest to combine our work with the work on CoD with respect to dimension of Euclidean space, cited in Section 1. Furthermore, in practice we recommend learning the Hamiltonian flow, which underlies the method of characteristics for the HJ equation, using symplectic neural networks (Jin *et al.*, 2020). However, the analysis of these neural networks is not yet developed to the extent needed for our complexity analysis in this paper, which builds on the work in (Yarotsky, 2017). Extending the analysis of symplectic neural networks, and using this extension to analyse generalizations of HJ-Net as defined here are interesting directions for future study. Finally, we note that it is of interest to iterate the learned operator. In order to do this we would need to generalize the error estimates to the C^1 topology. This could be achieved either by interpolation between higher order C^s bounds of the proposed methodology for $s > 1$ combined with the existing error analysis, or by using the gradient variable p in the interpolation.

Funding

Swiss National Science Foundation (P500PT-206737 to S.L.); Department of Defense Vannevar Bush Faculty Fellowship.

References

Albano, P. & Cannarsa, P. (2002) Propagation of singularities for solutions of nonlinear first order partial differential equations. *Arch. Ration. Mech. Anal.*, **162**, 1–23.

Albano, P., Cannarsa, P. & Sinestrari, C. (2020) Generation of singularities from the initial datum for Hamilton–Jacobi equations. *J. Differ. Equations*, **268**, 1412–1426.

Anthony, M. & Bartlett, P. L. (1999) *Neural Network Learning: Theoretical Foundations*. Cambridge, England: Cambridge University Press.

Bach, F. (2017) Breaking the curse of dimensionality with convex neural networks. *J. Mach. Learn. Res.*, **18**, 1–53.

Barron, A. R. (1993) Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Tran. Inf. Theory*, **39**, 930–945.

Barron, E., Cannarsa, P., Jensen, R. & Sinestrari, C. (1999) Regularity of Hamilton–Jacobi equations when forward is backward. *Indiana Univ. Math. J.*, **48**, 1–409.

Bhattacharya, K., Hosseini, B., Kovachki, N. B. & Stuart, A. M. (2021) Model reduction and neural networks for parametric PDEs. *SMAI J. Comput. Math.*, **7**, 121–157.

Boullé, N., Earls, C. J. & Townsend, A. (2022) Data-driven discovery of green's functions with human-understandable deep learning. *Sci. Rep.*, **12**, 1–9.

Boullé, N. & Townsend, A. (2022) Learning elliptic partial differential equations with randomized linear algebra. *Found. Comput. Math.*, **23**, 709–739.

Chen, T. & Chen, H. (1995) Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Networks*, **6**, 911–917.

Chow, S.-N. & Hale, J. K. (2012) *Methods of Bifurcation Theory*, vol. **251**. New York: Springer Science & Business Media.

Chow, Y. T., Darbon, J., Osher, S. & Yin, W. (2017) Algorithm for overcoming the curse of dimensionality for time-dependent non-convex Hamilton–Jacobi equations arising from optimal control and differential games problems. *J. Sci. Comput.*, **73**, 617–643.

Chow, Y. T., Darbon, J., Osher, S. & Yin, W. (2019) Algorithm for overcoming the curse of dimensionality for state-dependent Hamilton–Jacobi equations. *J. Comput. Phys.*, **387**, 376–409.

Christensen, O. *et al.* (2003) *An Introduction to Frames and Riesz Bases*, vol. **7**. Science+Business Media, New York: Springer.

Dahlke, S., De Mari, Grohs, P. & Labate, D. (2015) *Harmonic and Applied Analysis. Appl. Numer. Harmon. Anal*. Cham Heidelberg New York Dordrecht London: Springer International Publishing Switzerland.

Darbon, J., Dower, P. M. & Meng, T. (2023) Neural network architectures using min-plus algebra for solving certain high-dimensional optimal control problems and Hamilton–Jacobi pdes. *Math. Control Signals Syst.*, **35**, 1–44.

Darbon, J., Langlois, G. P. & Meng, T. (2020) Overcoming the curse of dimensionality for some Hamilton–Jacobi partial differential equations via neural network architectures. *Res. Math. Sci.*, **7**, 1–50.

Darbon, J. & Meng, T. (2021) On some neural network architectures that can represent viscosity solutions of certain high dimensional Hamilton–Jacobi partial differential equations. *J. Comput. Phys.*, **425**, 109907.

Darbon, J. & Osher, S. (2016) Algorithms for overcoming the curse of dimensionality for certain Hamilton–Jacobi equations arising in control theory and elsewhere. *Res. Math. Sci.*, **3**, 1–26.

de Hoop, M., Huang, D. Z., Qian, E. & Stuart, A. M. (2022) The cost-accuracy trade-off in operator learning with neural networks. *J. Mach. Learn.*, **1**, 299–341.

de Hoop, M. V., Kovachki, N. B., Nelsen, N. H. & Stuart, A. M. (2023) Convergence rates for learning linear operators from noisy data. *SIAM/ASA J. Uncertainty Quantif.*, **11**, 480–513.

Deng, B., Shin, Y., Lu, L., Zhang, Z. & Karniadakis, G. E. (2022) Approximation rates of deeponets for learning operators arising from advection—Diffusion equations. *Neural Networks*, **153**, 411–426.

DeVore, R. A., Hanin, B. & Petrova, G. (2021) Neural network approximation. *Acta Numer.*, **30**, 327–444.

DeVore, R. A. (1998) Nonlinear approximation. *Acta Numer.*, **7**, 51–150.

Donoho, D. L. (2001) Sparse components of images and optimal atomic decompositions. *Constr. Approx.*, **17**, 353–382.

Evans, L. C. (2010) *Partial Differential Equations*. Providence, R.I: American Mathematical Society.

Goodfellow, I., Bengio, Y. & Courville, A. (2016) *Deep Learning*. Cambridge, MA: MIT press.

Gribonval, R., Kutyniok, G., Nielsen, M. & Voigtlaender, F. (2022) Approximation spaces of deep neural networks. *Constr. Approx.*, **55**, 259–367.

Heil, C. (2010) *A Basis Theory Primer: Expanded Edition*. New York: Springer Science & Business Media.

Herrmann, L., Schwab, C. & Zech, J. (2020) Deep ReLU neural network expression rates for data-to-qoi maps in Bayesian PDE inversion. *SAM Res. Rep.*, **36**, 125011.

Herrmann, L., Schwab, C. & Zech, J. (2024) Neural and GPC operator surrogates: Construction and expression rate bounds. *Advances in Computational Mathematics*, **50**, 72. https://doi.org/10.1007/s10444-024-10171-2.

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

Hua, N. & Lu, W. (2023) Basis operator network: A neural network-based model for learning nonlinear operators via neural basis. *Neural Networks*, **164**, 21–37.

Huré, C., Pham, H., Bachouch, A. & Langrené, N. (2021) Deep neural networks algorithms for stochastic control problems on finite horizon: Convergence analysis. *SIAM J. Numer. Anal.*, **59**, 525–557.

Jin, P., Zhang, Z., Zhu, A., Tang, Y. & Karniadakis, G. E. (2020) Sympnets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, **132**, 166–179.

Khoo, Y., Lu, J. & Ying, L. (2021) Solving parametric PDE problems with artificial neural networks. *European J. Appl. Math.*, **32**, 421–435.

Kohler, M. & Langer, S. (2021) On the rate of convergence of fully connected deep neural network regression estimates. *Ann. Stat.*, **49**, 2231–2249.

Korolev, Y. (2022) Two-layer neural networks with values in a banach space. *SIAM J. Math. Anal.*, **54**, 6358–6389.

Kovachki, N. B., Lanthaler, S. & Mishra, S. (2021) On universal approximation and error bounds for Fourier neural operators. *J. Mach. Learn. Res.*, **22**, 1–76.

Kovachki, N. B., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A. & Anandkumar, A. (2023) Neural operator: Learning maps between function spaces with applications to PDEs. *J. Mach. Learn. Res.*, **24**, 1–97.

Lanthaler, S. (2023) Operator learning with PCA-net: Upper and lower complexity bounds. *J. Mach. Learn. Res.*, **24**, 1–67.

Lanthaler, S. (2024) *Operator Learning of Lipschitz Operators: An Information-Theoretic Perspective*, arXiv:2406.18794.

Lanthaler, S., Li, Z. & Stuart, A. M. (2023a) The nonlocal neural operator: Universal approximation. arXiv preprint arXiv:2304.13221.

Lanthaler, S., Molinaro, R., Hadorn, P. & Mishra, S. (2023b) Nonlinear reconstruction for operator learning of PDEs with discontinuities. *Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=CrfhZAsJDsZ.

Lanthaler, S., Mishra, S. & Karniadakis, G. E. (2022) Error estimates for DeepONets: A deep learning framework in infinite dimensions. *Trans. Math. Appl.*, **6**, 1–141.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. M. & Anandkumar, A. (2021) Fourier neural operator for parametric partial differential equations. *Ninth International Conference on Learning Representations*. https://openreview.net/forum?id=c8P9NQVtmnO.

Liu, H., Yang, H., Chen, M., Zhao, T. & Liao, W. (2024) Deep nonparametric estimation of operators between infinite dimensional spaces. *J. Mach. Learn. Res.*, **25**, 1–67.

Lu, J., Shen, Z., Yang, H. & Zhang, S. (2021a) Deep network approximation for smooth functions. *SIAM J. Math. Anal.*, **53**, 5465–5506.

Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. (2021b) Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nat. Mach. Intell.*, **3**, 218–229.

Ma, C., Wu, L. & Weinan, E. (2022) The Barron space and the flow-induced function spaces for neural network models. *Constr. Approx.*, **55**, 369–406.

Marcati, C. & Schwab, C. (2023) Exponential convergence of deep operator networks for elliptic partial differential equations. *SIAM J. Numer. Anal.*, **61**, 1513–1545.

Meng, T., Zhang, Z., Darbon, J. & Karniadakis, G. (2022) Sympocnet: Solving optimal control problems with applications to high-dimensional multiagent path planning problems. *SIAM J. Sci. Comput.*, **44**, B1341–B1368.

Mhaskar, H. N., Liao, Q. & Poggio, T. (2016) Learning real and Boolean functions: When is deep better than shallow. Technical report, Center for Brains, minds and machines (CBMM).

Mhaskar, H. N. & Hahm, N. (1997) Neural networks for functional approximation and system identification. *Neural Comput.*, **9**, 143–159.

Nakamura-Zimmerer, T., Gong, Q. & Kang, W. (2020) Qrnet: Optimal regulator design with lqr-augmented neural networks. *IEEE Control Syst. Lett.*, **5**, 1303–1308.

Nelsen, N. H. & Stuart, A. M. (2021) The random feature model for input-output maps between banach spaces. *SIAM J. Sci. Comput.*, **43**, A3212–A3243.

Opschoor, J. A., Schwab, C. & Zech, J. (2022) Exponential relu dnn expression of holomorphic maps in high dimension. *Constr. Approx.*, **55**, 537–582.

Patel, D., Ray, D., Abdelmalik, M. R., Hughes, T. J. & Oberai, A. A. (2024) Variationally mimetic operator networks. *Comput. Methods Appl. Mech. Eng.*, **419**, 116536, 1–30.

Petersen, P. & Voigtlaender, F. (2018) Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, **108**, 296–330.

Robbins, H. & Monro, S. (1951) A stochastic approximation method. *Ann. Math. Stat.*, **22**, 400–407.

Ryck, T. D. & Mishra, S. (2022) Generic bounds on the approximation error for physics-informed (and) operator learning. *Advances in Neural Information Processing Systems* (A. H. Oh, A. Agarwal, D. Belgrave & K. Cho eds.). https://openreview.net/forum?id=bF4eYy3LTR9.

Schwab, C., Stein, A. & Zech, J. (2023) Deep operator network approximation rates for lipschitz operators. arXiv preprint arXiv:2307.09835.

Schwab, C. & Zech, J. (2019) Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ. *Anal. Appl.*, **17**, 19–55.

Seidman, J., Kissas, G., Perdikaris, P. & Pappas, G. J. (2022) NOMAD: Nonlinear manifold decoders for operator learning. *Advances in Neural Information Processing Systems*, **35**, 5601–5613.

Stein, E. M. & Shakarchi, R. (2009) *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*. Princeton, New Jersey: Princeton University Press.

Wendland, H. (2004) *Scattered Data Approximation*, vol. **17**. Cambridge, England: Cambridge University Press.

Yarotsky, D. (2017) Error bounds for approximations with deep ReLU networks. *Neural Networks*, **94**, 103–114 Publisher: Elsevier.

Yarotsky, D. & Zhevnerchuk, A. (2020) The phase diagram of approximation rates for deep neural networks. *Advances in neural information processing systems*, **33**, 13005–13015.

You, H., Zhang, Q., Ross, C. J., Lee, C.-H. & Yu, Y. (2022) Learning deep implicit fourier neural operators (ifnos) with applications to heterogeneous material modeling. *Comput. Methods Appl. Mech. Eng.*, **398**, 115296.

Zhang, Z., Tat, L. & Schaeffer, H. (2023) BelNet: Basis enhanced learning, a mesh-free neural operator. *Proceedings of the Royal Society A*, **479**, 20230043, 1–30.

Zhu, Y. & Zabaras, N. (2018) Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *J. Comput. Phys.*, **366**, 415–447.

Appendix A. Proof of curse of parametric complexity

A.1 *Preliminaries in finite dimensions*

Our goal in this section is to prove Proposition 2.1, which we repeat here:

To this end we recall and extend several results from Yarotsky (2017) on the ReLU neural network approximation of functions $f : \mathbb{R}^D \to \mathbb{R}$. Subsequently, these results will be used as building blocks to construct functionals in the infinite-dimensional context, leading to a curse of parametric complexity in that setting, made precise in Theorem 2.11. Here, we consider the space $C^r(\mathbb{R}^D)$ consisting of r-times continuously differentiable functions $f : \mathbb{R}^D \to \mathbb{R}$. We denote by

$$F_{D,r} := \left\{ f \in C^r(\mathbb{R}^D) \,\middle|\, \sup_{|\alpha|\le r} \sup_{x\in\mathbb{R}^D} \left|D^\alpha f(x)\right| \le 1 \right\}$$

the unit ball in $C^r(\mathbb{R}^D)$. For technical reasons it will often be more convenient to consider the subset $\mathring{F}_{D,r} \subset F_{D,r}$ consisting of functions $f \in F_{D,r}$ vanishing at the origin, $f(0) = 0$.

Remark A1. We note that previous work (Yarotsky, 2017) considers the Sobolev space $W^{r,\infty}([0,1]^D)$ and the unit ball in $W^{r,\infty}([0,1]^D)$, rather than $C^r(\mathbb{R}^D)$ and our definition of $F_{D,r}$. It turns out that the complexity bounds of Yarotsky (2017) remain true also in our setting (with essentially identical proofs). We include the necessary arguments below. ♢

Let $f : \mathbb{R}^D \to \mathbb{R}$ be a function. Following (Yarotsky, 2017, section 4.3) we will denote by $\mathcal{N}(f, \epsilon)$ the minimal number of hidden computation units that is required to approximate f to accuracy ϵ by an arbitrary ReLU feedforward network Ψ, *uniformly over the unit cube* $[0, 1]^D$, i.e., $\mathcal{N}(f, \epsilon)$ is the minimal integer M such that there exists an ReLU neural network Ψ with at most M computation units[4] and such that

$$\sup_{x\in[0,1]^D} |f(x) - \Psi(x)| \leq \epsilon.$$

REMARK A2. We emphasize that, even though the domain of a function $f \in F_{D,r}$ is by definition the whole space $\mathbb{R}^D$, the above quantity $\mathcal{N}(f, \epsilon)$ only relates to the approximation of f over the unit cube $[0, 1]^D$. The reason for this seemingly awkward choice is that it will greatly simplify our arguments later on, when we construct functionals on infinite-dimensional Banach spaces with a view towards proving an infinite-dimensional analogue of the CoD. ◊

Note that the number of (nontrivial) hidden computation units M of a neural network $\Psi : \mathbb{R}^D \to \mathbb{R}$ obeys the bounds $M \leq \text{size}(\Psi) \leq 5DM^4$: the lower bound follows from the fact that each (nontrivial) computation unit has at least one nonzero connection or a bias feeding into it. To derive the upper bound we note that any network with at most M computation units can be embedded in a fully connected enveloping network (allowing skip-connections) (Yarotsky, 2017, cf. fig. 6(a)) with depth M, width M, where each hidden node is connected to all other $M^2 - 1$ hidden nodes plus the output, and where each node in the input layer is connected to all M^2 hidden units plus the output. In addition, we take into account that each computation unit and the output layer have an additional bias. The existence of this enveloping network thus implies the size bound

$$\begin{aligned}\text{size}(\Psi) &\leq \underbrace{M^2(M^2-1)+M^2}_{\text{hidden connections}} + \underbrace{D(M^2+1)}_{\text{input conn.}} + \underbrace{M+1}_{\text{biases}} \\ &\leq M^4 + 2DM^4 + M + 1 \leq 5DM^4,\end{aligned}$$

valid for any neural network $\Psi : \mathbb{R}^D \to \mathbb{R}$ with at most M computation units.

In view of the lower size bound, $\text{size}(\Psi) \geq M$, Proposition 2.1 above is now clearly implied by the following:

PROPOSITION A3. Fix $r \in \mathbb{N}$. There is a universal constant $\gamma > 0$ and a constant $\overline{\epsilon} = \overline{\epsilon}(r) > 0$, depending only on r, such that for any $D \in \mathbb{N}$ there exists a function $f_D \in \dot{F}_{D,r}$ for which

$$\mathcal{N}(f_D, \epsilon) \geq \epsilon^{-\gamma D/r}, \quad \forall \epsilon \leq \overline{\epsilon}.$$

◊

As an immediate corollary of Proposition A3 we conclude that f_D cannot be approximated to accuracy ϵ by a family of ReLU neural networks Ψ_ϵ with

$$\text{size}(\Psi_\epsilon) = o(\epsilon^{-\gamma D/r}).$$

The latter conclusion was established by Yarotsky (Yarotsky, 2017, theorem 5) (with $\gamma = 1/9$). Proposition 2.1 improves on Yarotsky's result in two important ways: firstly, it implies that the lower

[4] The number of computation units equals the number of scalar evaluations of the activation σ in a forward-pass, cf. Anthony & Bartlett (1999).

bound size$(\Psi_\epsilon) \geq \epsilon^{-\gamma D/r}$ holds for *all* $\epsilon \leq \bar{\epsilon}$, and not only along a (unspecified) sequence $\epsilon_k \to 0$; secondly, it shows that the constant $\bar{\epsilon}$ can be chosen *independently* of D. This generalization of Yarotsky's result will be crucial for the extension to the infinite-dimensional case, which is the goal of this work.

To prove Proposition A3 we will need two intermediate results, which build on Yarotsky (2017). We start with the following lemma providing a lower bound on the required size of a fixed neural network architecture, which is able to approximate arbitrary $f \in \mathring{F}_{D,r}$ to accuracy $\epsilon \geq 0$. This result is based on (Yarotsky, 2017, theorem 4(a)), but explicitly quantifies the dependence on the dimension D. This dependence was left unspecified in earlier work.

LEMMA A4. Fix $r \in \mathbb{N}$. Let $\Psi = \Psi(\cdot;\theta)$ be an ReLU neural network architecture depending on parameters $\theta \in \mathbb{R}^W$. There exists a constant $c_0 = c_0(r) > 0$, such that if

$$\sup_{f \in \mathring{F}_{D,r}} \inf_{\theta \in \mathbb{R}^W} \|f - \Psi(\cdot;\theta)\|_{L^\infty([0,1]^D)} \leq \epsilon,$$

for some $\epsilon > 0$ then $W \geq (c_0\epsilon)^{-D/2r}$. ◊

Proof. The proof in Yarotsky (2017) is based on the Vapnik–Chervonenkis (VC) dimension. We will not repeat the entire argument here, but instead discuss only the required changes in the proof of Yarotsky, which are needed to prove our extension of his result. In fact, there are only two differences between the proof our Lemma A.4, and (Yarotsky, 2017, theorem 4(a)), which we now point out: the first difference is that we consider $\mathring{F}_{D,r}$, consisting of $C^r(\mathbb{R}^D)$ functions vanishing at the origin, whereas Yarotsky (2017) considers the unit ball in $W^{r,\infty}([0,1]^D)$. Nevertheless, the proof of Yarotsky (2017) applies to our setting in a straightforward way. To see this we recall that the construction in (Yarotsky, 2017, proof of theorem 4, eq. (35)) considers $f \in F_{D,r}$ of the form

$$f(x) = \sum_{m=1}^{N^D} y_m \phi(N(x - x_m)), \tag{A.1}$$

with coefficients y_m to be determined later, and where $\phi : \mathbb{R}^D \to \mathbb{R}$ is a C^∞ bump function, which is required to satisfy[5]

$$\phi(0) = 1 \quad \text{and} \quad \phi(x) = 0, \text{ if } \|x\|_{\ell^\infty} > 1/2. \tag{A.2}$$

In (A.1), the points $x_1, \ldots, x_{N^D} \in [0,1]^D$ are chosen such that the ℓ^∞-distance between any two of them is at least $1/N$. We can easily ensure that $f(0) = 0$, by choosing the points x_m to be of the form $(j_1, \ldots, j_D)/N$, where $j_1, \ldots, j_D \in \{1, \ldots, N\}$. Then, since for any multi-index α of order $|\alpha| \leq r$ the mixed derivative

$$\max_x |D^\alpha f(x)| \leq N^r \max_m |y_m| \max_x |D^\alpha \phi(x)|,$$

any function f of the form (A.1) belongs to $\mathring{F}_{D,r}$, provided that

$$\max_m |y_m| \leq \widetilde{c}_1 N^{-r},$$

[5] We note that choosing the ℓ^∞ to define the set where $\phi(x) = 0$, rather than the ℓ^2 Euclidean distance as in Yarotsky (2017), is immaterial. The only requirement is that the support of the functions on the right-hand side in the definition (A.1) do not overlap.

where

$$\widetilde{c}_1 = \left(\sup_{|\alpha| \le r} \sup_{x \in [0,1]^D} |D^\alpha \phi(x)| \right)^{-1}. \tag{A.3}$$

As shown in Yarotsky (2017) the above construction allows one to encode arbitrary Boolean values $z_1, \dots, z_{N^D} \in \{0,1\}$ by construction suitable $f \in \mathring{F}_{D,r}$; this in turn can be used to estimate a VC-dimension related to $\Psi(\cdot\,;\theta)$ from below, following verbatim the arguments in Yarotsky (2017). As no changes are required in this part of the proof we will not repeat the details here; Instead, following the argument leading up to (Yarotsky, 2017, eq. (38)), and under the assumptions of Lemma A.4, Yarotsky's argument immediately implies the following lower bound:

$$W \ge \widetilde{c}_0 \left(\frac{3\epsilon}{\widetilde{c}_1} \right)^{-D/2r},$$

where $\widetilde{c}_0$ is an absolute constant.

To finish our proof of Lemma A.4, it remains to determine the dependence of the constant $\widetilde{c}_1$ in (A.3), on the parameters D and r. To this end, we construct a specific bump function $\phi : \mathbb{R}^D \to \mathbb{R}$. Recall that our only requirement on ϕ in the above argument is that (A2) must hold. To construct suitable ϕ, let $\phi_0 : \mathbb{R} \to \mathbb{R}, \xi \mapsto \phi_0(\xi)$ be a smooth bump function such that $\phi_0(0) = 1$, $\|\phi_0\|_{L^\infty} \le 1$ and $\phi_0(\xi) = 0$ for $|\xi| > 1/2$. We now make the particular choice

$$\phi(x_1, \dots, x_D) := \prod_{j=1}^{D} \phi_0(x_j).$$

Let $\alpha = (\alpha_1, \dots, \alpha_D)$ be a multi-index with $|\alpha| = \sum_{j=1}^D \alpha_j \le r$. Let $\kappa := |\{\alpha_j \ne 0\}$ denote the number of nonzero components of α. We note that $\kappa \le r$. We thus have

$$|D^\alpha \phi(x)| = \prod_{j=1}^{D} |D^{\alpha_j} \phi_0(x_j)| \le \prod_{\alpha_j \ne 0} |D^{\alpha_j} \phi_0(x_j)| \le \|\phi_0\|^\kappa_{C^r(\mathbb{R})} \le \|\phi_0\|^r_{C^r(\mathbb{R})}.$$

In particular, we conclude that $\widetilde{c}_1 = [\sup_{|\alpha| \le r} \sup_{x \in [0,1]^D} |D^\alpha \phi(x)|]^{-1}$ can be bounded from below by $\widetilde{c}_1 \ge \widetilde{c}_2 := \|\phi_0\|^{-r}_{C^r(\mathbb{R})}$, where $\widetilde{c}_2 = \widetilde{c}_2(r)$ clearly only depends on r, and is *independent* of the ambient dimension D. The claimed inequality of Lemma A.4 now follows from

$$W \ge \widetilde{c}_0 \left(\frac{3\epsilon}{\widetilde{c}_1} \right)^{-D/2r} \ge \left(\frac{3\epsilon}{(\widetilde{c}_0 \wedge 1)^{2r/D} \widetilde{c}_2} \right)^{-D/2r} \ge \left(\frac{3\epsilon}{(\widetilde{c}_0 \wedge 1)^{2r} \widetilde{c}_2} \right)^{-D/2r},$$

i.e., we have $W \ge (c_0 \epsilon)^{-D/2r}$ with constant

$$c_0 = \frac{3}{(\widetilde{c}_0 \wedge 1)^{2r} \widetilde{c}_2(r)},$$

depending only on r. □

While Lemma A4 applies to a *fixed* architecture capable of approximating all $f \in \mathring{F}_{D,r}$, our goal is to instead construct a single $f \in \mathring{F}_{D,r}$ for which a similar lower complexity bound holds for *arbitrary* architectures. The construction of such f will rely on the following lemma, based on (Yarotsky, 2017, Lem. 3):

LEMMA A5. Fix $r \in \mathbb{N}$. For any (fixed) $\epsilon > 0$, there exists $f_\epsilon \in \mathring{F}_{D,r}$, such that

$$\mathcal{N}(f_\epsilon, \epsilon) \geq D^{-1/4}(c_1\epsilon)^{-D/8r},$$

where $c_1 = c_1(r) > 0$ depends only on r. ◊

Proof. As explained above, any neural network (with potentially sparse architecture), $\Psi : \mathbb{R}^D \to \mathbb{R}$, with M computation units can be embedded in the fully connected architecture $\widetilde{\Psi}(\cdot\,;\theta)$, $\theta \in \mathbb{R}^W$, with size bound $W \leq 5DM^4$. By Lemma A4, it follows that if $W \leq 5DM^4 < (c_0\epsilon)^{-D/2r}$, then there exists $f_\epsilon \in \mathring{F}_{D,r}$, such that

$$\inf_{\theta\in\mathbb{R}^W} \|f_\epsilon - \widetilde{\Psi}(\cdot\,;\theta)\|_{L^\infty} > \epsilon.$$

Expressing the above size bound in terms of M, it follows that for any network Ψ with $M < (5D)^{-1/4}(c_0\epsilon)^{-D/8r}$ computation units, we must have $\|f_\epsilon - \Psi\|_{L^\infty} > \epsilon$. Thus, approximation of f_ϵ to within accuracy ϵ requires at least $M \geq (5D)^{-1/4}(c_0\epsilon)^{-D/8r}$ computation units. From the definition of $\mathcal{N}(f_\epsilon, \epsilon)$, we conclude that

$$\mathcal{N}(f_\epsilon, \epsilon) \geq D^{-1/4}(c_1\epsilon)^{-D/8r},$$

for this choice of $f_\epsilon \in \mathring{F}_{D,r}$, with constant $c_1 = 5^{2r}c_0(r)$ depending only on r. □

Lemma A5 will be our main technical tool for the proof of Proposition A3. We also recall that $\mathcal{N}(f, \epsilon)$ satisfies the following properties, (Yarotsky, 2017, eq. (42)–(44)):

$$\mathcal{N}(af, |a|\epsilon) = \mathcal{N}(f, \epsilon), \tag{A.4a}$$

$$\mathcal{N}(f \pm g, \epsilon + \|g\|_{L^\infty}) \leq \mathcal{N}(f, \epsilon), \tag{A.4b}$$

$$\mathcal{N}(f_1 \pm f_2, \epsilon_1 + \epsilon_2) \leq \mathcal{N}(f_1, \epsilon_1) + \mathcal{N}(f_2, \epsilon_2). \tag{A.4c}$$

Proof (Proposition A3). We define a rapidly decaying sequence $\epsilon_k \to 0$, by $\epsilon_1 = 1/4$ and $\epsilon_{k+1} = \epsilon_k^2$, so that by recursion $\epsilon_k = 2^{-2^k}$. We also define $a_k := \frac{1}{2}\epsilon_k^{1/2} = \frac{1}{2}\epsilon_{k-1}$. For later reference, we note that since $\epsilon_k \leq 1/2$ for all k, we have

$$\sum_{s=k+1}^{\infty} a_s = \frac{1}{2}\left(\epsilon_k + \epsilon_k^2 + \epsilon_k^{2^2} + \dots\right) \leq \frac{1}{2}\epsilon_k \sum_{s=0}^{\infty} 2^{-s} = \epsilon_k. \tag{A.5}$$

Our goal is to construct $f \in \mathring{F}_{D,r}$ of the form

$$f = \sum_{k=1}^{\infty} a_k f_k, \quad \text{with } f_k \in \mathring{F}_{D,r}\ \forall\, k \in \mathbb{N}.$$

We note that $a_k \leq 2^{-k}$, hence if f is of the above form then,

$$\|f\|_{C^r} \leq \sum_{s=1}^{\infty} a_s \|f_s\|_{C^r} \leq 1, \quad f(0) = \sum_{s=1}^{\infty} a_s \underbrace{f_s(0)}_{=0} = 0,$$

implies that $f \in \mathring{F}_{D,r}$ irrespective of the specific choice of $f_k \in \mathring{F}_{D,r}$. In the following construction, we choose $\gamma := 1/32$ throughout. We determine f_k recursively, formally starting from the empty sum, i.e., $f = 0$. In the recursive step, given $f_1, \ldots, f_{k-1}$, we distinguish two cases:

Case 1: Assume that

$$\mathcal{N}\left(\sum_{s=1}^{k-1} a_s f_s, 2\epsilon_k\right) \geq \epsilon_k^{-\gamma D/r}.$$

In this case, we set $f_k := 0$, and trivially obtain

$$\mathcal{N}\left(\sum_{s=1}^{k} a_s f_s, 2\epsilon_k\right) \geq \epsilon_k^{-\gamma D/r}. \tag{A.6}$$

Case 2: In the other case, we have

$$\mathcal{N}\left(\sum_{s=1}^{k-1} a_s f_s, 2\epsilon_k\right) < \epsilon_k^{-\gamma D/r}.$$

Our first goal is to again choose f_k such that (A.6) holds, at least for sufficiently large k. We note that, by the general properties of $\mathcal{N}$, (A.4c), and the assumption of Case 2:

$$\begin{aligned}\mathcal{N}\left(\sum_{s=1}^{k} a_s f_s, 2\epsilon_k\right) &\geq \mathcal{N}(a_k f_k, 4\epsilon_k) - \mathcal{N}\left(\sum_{s=1}^{k-1} a_s f_s, 2\epsilon_k\right) \\ &\geq \mathcal{N}(a_k f_k, 4\epsilon_k) - \epsilon_k^{-\gamma D/r}. \end{aligned} \tag{A.7}$$

By Lemma A5, we can find $f_k \in \mathring{F}_{D,r}$, such that

$$\mathcal{N}(a_k f_k, 4\epsilon_k) \overset{(A.4a)}{=} \mathcal{N}(f_k, 4\epsilon_k/a_k) = \mathcal{N}(f_k, 4\epsilon_k^{1/2}) \geq D^{-1/4}(8c_1\epsilon_k^{1/2})^{-D/8r}. \tag{A.8}$$

This defines our recursive choice of f_k in Case 2. By (A.7) and (A.8), to obtain (A.6) it now suffices to show that

$$D^{-1/4}(8c_1\epsilon_k^{1/2})^{-D/8r} \geq 2\epsilon_k^{-\gamma D/r}.$$

It turns out that there exists $\bar{\epsilon} > 0$ depending only on r, such that

$$(8^2 c_1^2 \epsilon_k)^{-D/16r} \geq 2D^{1/4}\epsilon_k^{-\gamma D/r} = (2^{-r/\gamma D} D^{-r/(4\gamma D)} \epsilon_k)^{-\gamma D/r},$$

for all $\epsilon_k \leq \bar{\epsilon}$, and where $\gamma = 1/32$. In fact, upon raising both sides to the exponent $-32r/D$, it is straightforward to see that this inequality holds independently of $D \in \mathbb{N}$, provided that

$$\epsilon_k \leq \frac{\inf_D D^{-r/(4\gamma D)}}{[2^{8r} 8c_1(r)]^4},$$

where we note that $\inf_D D^{-r/(4\gamma D)} > 0$ on account of the fact that $\lim_{D\to\infty} D^{-1/D} = 1$. Define $\overline{\epsilon}$ by

$$\overline{\epsilon}(r) := \min\left\{ k \in \mathbb{N} \,\middle|\, \epsilon_k \le \frac{\inf_D D^{-r/(4\gamma D)}}{[2^{8r} 8 c_1(r)]^4} \right\}.$$

With this choice of $\overline{\epsilon}, \gamma > 0$, and by construction of f_k, we then have

$$\mathcal{N}\left(\sum_{s=1}^{k} a_s f_s, 2\epsilon_k \right) \ge \epsilon_k^{-\gamma D/r}, \tag{A.9}$$

for all $\epsilon_k \le \overline{\epsilon}$. This is inequality (A.6), and concludes our discussion of Case 2.

Continuing the above construction by recursion, we obtain a sequence $f_1, f_2, \cdots \in \mathring{F}_{D,r}$, such that (A.9) holds for any $k \in \mathbb{N}$ such that $\epsilon_k \le \overline{\epsilon}$. Define $f = \sum_{k=1}^{\infty} a_k f_k$. We claim that for any $\epsilon \le \overline{\epsilon}$ we have

$$\mathcal{N}(f, \epsilon) \ge \epsilon^{-\gamma D/2r}.$$

To see this, we fix $\epsilon \le \overline{\epsilon}$. Choose $k \in \mathbb{N}$, such that $\epsilon_k \le \overline{\epsilon}$ and $\epsilon_k^2 \le \epsilon \le \epsilon_k$. Then,

$$\begin{aligned}
\mathcal{N}(f, \epsilon) \ge \mathcal{N}(f, \epsilon_k) &= \mathcal{N}\left(\sum_{s=1}^{\infty} a_s f_s, \epsilon_k \right) \\
&\overset{(A.4b)}{\downarrow} \\
&\ge \mathcal{N}\left(\sum_{s=1}^{k} a_s f_s, \epsilon_k + \left\| \sum_{s=k+1}^{\infty} a_s f_s \right\|_{L^\infty} \right) \\
&\ge \mathcal{N}\left(\sum_{s=1}^{k} a_s f_s, \epsilon_k + \sum_{s=k+1}^{\infty} a_s \right) \\
&\overset{(A.5)}{\downarrow} \\
&\ge \mathcal{N}\left(\sum_{s=1}^{k} a_s f_s, 2\epsilon_k \right) \\
&\overset{(A.9)}{\downarrow} \\
&\ge \epsilon_k^{-\gamma D/r} \ge \epsilon^{-\gamma D/2r},
\end{aligned}$$

where the last inequality follows from $\epsilon_k \le \epsilon^{1/2}$. The claim of Proposition A3 thus follows for all $\epsilon \le \overline{\epsilon}$, upon redefining the universal constant as $\gamma = (1/32)/2 = 1/64$. □

A.2 *Proof of Lemma* 2.7

Proof (Lemma 2.7). Since the interior of Ω is nonempty, then upon a rescaling and shift of the domain, we may wlog assume that $[0, 2\pi]^d \subset \Omega$. Let us define $e_\kappa \propto \sin(\kappa \cdot x)$ as a suitable re-normalization of the standard Fourier sine-basis, indexed by $\kappa \in \mathbb{N}^d$, and normalized such that $\|e_\kappa\|_{C^s} = 1$. We note that for each e_κ we can define a bi-orthogonal functional e_κ^* by

$$e_\kappa^*(u) := \frac{2}{(2\pi)^d} \int_{[0,2\pi]^d} u(x) \sin(\kappa \cdot x)\, \mathrm{d}x.$$

The norm of e_κ^* is easily seen to be bounded by 2. Hence, for any enumeration $j \mapsto \kappa(j)$, the sequence $e_{\kappa(j)}$ satisfies the assumptions in the definition of a infinite-dimensional hypercube.

Furthermore, if $j \mapsto \kappa(j)$ is an enumeration of $\kappa \in \mathbb{N}^d$, such that $j \mapsto \|\kappa(j)\|_{\ell^\infty}$ is monotonically increasing (nondecreasing), we note that any series of the form

$$u = A \sum_{j=1}^{\infty} j^{-\alpha} y_j \, e_{\kappa(j)}, \quad y_j \in [0,1],$$

is absolutely convergent in $C^\rho(\Omega)$, provided that

$$\sum_{j=1}^{\infty} j^{-\alpha} \| \, e_{\kappa(j)} \|_{C^\rho(\Omega)} \sim \sum_{j=1}^{\infty} j^{-\alpha} \|\kappa(j)\|_{\ell^\infty}^{\rho-s} < \infty.$$

Inverting the enumeration $j = j(\kappa)$ for $\kappa \in \mathbb{N}^d$, and setting $K := \|\kappa\|_{\ell^\infty}$, we find that

$$\#\left\{\kappa' \in \mathbb{N}^d \;\middle|\; \|\kappa'\|_{\ell^\infty} < K\right\} \le j(\kappa) \le \#\left\{\kappa' \in \mathbb{N}^d \;\middle|\; \|\kappa'\|_{\ell^\infty} \le K\right\},$$

where the number of elements in the lower and upper bounding sets are $\sim K^d$. We thus conclude that $j(\kappa) \sim K^d$. We also note that each shell $\left\{\kappa \in \mathbb{N}^d \;\middle|\; \|\kappa\|_{\ell^\infty} = K\right\}$, with $K \in \mathbb{N}$, contains $\sim K^{d-1}$ elements. Thus, we have

$$\begin{aligned}
\sum_{j=1}^{\infty} j^{-\alpha} \|\kappa(j)\|_{\ell^\infty}^{\rho-s} &\sim \sum_{K=1}^{\infty} \sum_{\|\kappa\|_{\ell^\infty}=K} j(\kappa)^{-\alpha} \|\kappa\|_{\ell^\infty}^{\rho-s} \\
&\sim \sum_{K=1}^{\infty} K^{-\alpha d} \sum_{\|\kappa\|_{\ell^\infty}=K} \|\kappa\|_{\ell^\infty}^{\rho-s} \\
&\sim \sum_{K=1}^{\infty} K^{-\alpha d} K^{d-1} K^{\rho-s} \\
&= \sum_{K=1}^{\infty} K^{-1-d\left(\alpha-1-\frac{\rho-s}{d}\right)}.
\end{aligned}$$

The last sum converges if $\alpha > 1 - \frac{\rho-s}{d}$. Thus, choosing $A > 0$ sufficiently small then ensures that $Q_\alpha = Q_\alpha(A; e_1, e_2, \dots)$ is a subset of $K = \left\{u \in C^\rho(\Omega) \;\middle|\; \|u\|_{C^\rho} \le M\right\}$. □

A.3 *Proof of Theorem* 2.11

We now provide the detailed proof of Theorem 2.11, which builds on Proposition A.3 above.

Proof. Fix $\delta > 0$. By assumption on $K \subset \mathcal{X}$, there exists $A > 0$ and a linearly independent sequence $e_1, e_2, \cdots \in \mathcal{X}$ of normed elements, such that the set Q_α consisting of all $u \in \mathcal{X}$ of the form

$$u = A \sum_{j=1}^{\infty} j^{-\alpha} y_j \, e_j, \quad y_j \in [0,1],$$

defines a subset $Q_\alpha \subset K$.

Step 1 (Finding embedded cubes $\simeq [0,1]^D$). We note that for any $k \in \mathbb{N}$ and for any choice of $y_j \in [0,1]$, with indices $j = 2^k, \dots, 2^{k+1}-1$, we have

$$u = A2^{-(k+1)\alpha} \sum_{j=2^k}^{2^{k+1}-1} y_j e_j \in Q_\alpha. \tag{A.10}$$

Set $D = 2^k$, and let us denote the set of all such u by $\overline{Q}_D$, in the following. Note that up to the constant rescaling by $A2^{-(k+1)\alpha}$, $\overline{Q}_D$ can be identified with the D-dimensional unit cube $[0,1]^D$. In particular, since $\overline{Q}_D \subset Q_\alpha \subset K$, it follows that K contains a rescaled copy of $[0,1]^D$ for any such D. Furthermore, it will be important in our construction that all of the embedded cubes, defined by (A.10) for $k \in \mathbb{N}$, only intersect at the origin.

By Proposition A3, there exist constants $\gamma, \overline{\epsilon} > 0$, independent of D, such that given any $D = 2^k$, we can find $f_D : \mathbb{R}^D \to \mathbb{R}, f_D \in \mathring{F}_{D,r}$, for which the following lower complexity bound holds:

$$\mathcal{N}(f_D, \epsilon) \geq \epsilon^{-\gamma D/r}, \quad \forall \epsilon \leq \overline{\epsilon}. \tag{A.11}$$

Our aim is to construct $\mathcal{S}^\dagger : \mathcal{X} \to \mathbb{R}$, such that the restriction $\mathcal{S}^\dagger|_{\overline{Q}_D}$ to the cube $\overline{Q}_D \simeq [0,1]^D$ 'reproduces' this f_D. If this can be achieved, then our intuition is that $\mathcal{S}^\dagger$ embeds all f_D for $D = 2^k$, $k \in \mathbb{N}$, at once, and hence a rescaled version of the lower complexity bound $\gtrsim_D \epsilon^{-\gamma D/r}$ should hold for any D. Our next aim is to make this precise, and determine the implicit constant that arises due to the fact that $\overline{Q}_D$ is only a rescaled version of $[0,1]^D$.

Step 2 (Construction of $\mathcal{S}^\dagger$). To construct suitable $\mathcal{S}^\dagger$, we first recall that we assume the existence of 'bi-orthogonal' elements $e_1^*, e_2^*, \dots$ in the continuous dual space $\mathcal{X}^*$, such that

$$e_i^*(e_j) = \delta_{ij}, \ \forall\, i,j \in \mathbb{N},$$

and furthermore, there exists a constant $M > 0$, such that $\|e_j^*\|_{\mathcal{X}^*} \leq M$ for all $j \in \mathbb{N}$. Given the functions $f_D = f_{2^k}$ from Step 1, we now make the following ansatz for $\mathcal{S}^\dagger$:

$$\mathcal{S}^\dagger(u) = \sum_{k=1}^{\infty} 2^{-\alpha^* r k} f_{2^k}\left(A^{-1}2^{(k+1)\alpha}\left[e_{2^k}^*(u), \dots, e_{2^{k+1}-1}^*(u)\right]\right). \tag{A.12}$$

Here, $f_{2^k} = f_D$ (for $D = 2^k$) satisfies (A.11) and $\alpha^* = 1 + \alpha + \delta/2$. We note in passing that $\mathcal{S}^\dagger$ defines a r-times Fréchet differentiable functional. This will be rigorously shown in Lemma A7 below (with $c = A^{-1}2^\alpha$).

Step 3 (Relating $\mathcal{S}^\dagger$ with f_D). We next show in which way 'the restriction $\mathcal{S}^\dagger|_{\overline{Q}_D}$ to the cube $\overline{Q}_D \simeq [0,1]^D$ reproduces f_D'. Note that if $u \in \overline{Q}_D$ is of the form (A.10) with $D = 2^k$ and with coefficients

$$y := [y_{2^k}, \dots, y_{2^{k+1}-1}] \in [0,1]^{2^k} = [0,1]^D,$$

then, if $k' \neq k$,

$$[e_{2^{k'}}^*(u), \dots, e_{2^{k'+1}-1}^*(u)] = [0, \dots, 0],$$

while for $k' = k$ we find

$$[e_{2^k}^*(u), \dots, e_{2^{k+1}-1}^*(u)] = [y_{2^k}, \dots, y_{2^{k+1}-1}] = y.$$

From the fact that $f_{2^{k'}}(0) = 0$ for all k' by construction (recall that $f_{2^{k'}} \in \mathring{F}_{2^{k'},r}$), we conclude that

$$\mathcal{S}^\dagger(u) = 2^{-\alpha^* rk} f_{2^k}(y) = D^{-\alpha^* r} f_D(y), \tag{A.13}$$

for any $u \in \overline{Q}_D$. In this sense, '$\mathcal{S}^\dagger|_{\overline{Q}_D}$ reproduces f_D'.

Step 4 (Lower complexity bound, uniform in D). Let $\mathcal{S}_\epsilon : \mathcal{X} \to \mathbb{R}$ be a family of operators of neural network-type, such that

$$\sup_{u \in K} |\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u)| \le \epsilon, \quad \forall \epsilon > 0.$$

By assumption on $\mathcal{S}_\epsilon$ being of neural network-type, there exists $\ell = \ell_\epsilon \in \mathbb{N}$, a linear mapping $\mathcal{L}_\epsilon : \mathcal{X} \to \mathbb{R}^\ell$, and $\Phi_\epsilon : \mathbb{R}^\ell \to \mathbb{R}$ a neural network representing $\mathcal{S}_\epsilon$:

$$\mathcal{S}_\epsilon(u) = \Phi_\epsilon(\mathcal{L}_\epsilon u), \quad \forall u \in \mathcal{X}.$$

For $D = 2^k$, $k \in \mathbb{N}$, define $\iota_D : \mathbb{R}^D \to \mathcal{X}$ by

$$\iota_D(y) = A 2^{-\alpha k} \sum_{j=1}^{2^k} y_j e_{2^k + j - 1}.$$

Then, since $\mathcal{L}_\epsilon \circ \iota_D : \mathbb{R}^D \to \mathbb{R}^\ell$ is a linear mapping, there exists a matrix $W_D \in \mathbb{R}^{\ell \times D}$, such that $\mathcal{L}_\epsilon \circ \iota_D(y) = W_D y$ for all $y \in \mathbb{R}^D$. In particular, it follows that

$$\mathcal{S}_\epsilon(\iota_D(y)) = \Phi_\epsilon(W_d y),$$

By (A.10), we clearly have $\iota_D([0,1]^D) = \overline{Q}_D$. Let $\widehat{\Phi}_\epsilon(y) := D^{\alpha^* r} \Phi_\epsilon(W_D y)$. It now follows that

$$\begin{aligned}
\sup_{y \in [0,1]^D} \left| f_D(y) - \widehat{\Phi}_\epsilon(y) \right| &= D^{\alpha^* r} \sup_{y \in [0,1]^D} \left| D^{-\alpha^* r} f_D(y) - \Phi_\epsilon(W_D y) \right| \\
&= D^{\alpha^* r} \sup_{u \in \overline{Q}_D} \left| \mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u) \right| \\
&\le D^{\alpha^* r} \sup_{u \in K} \left| \mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u) \right| \\
&\le D^{\alpha^* r} \epsilon.
\end{aligned}$$

Let M denote the number of hidden computation units of $\widehat{\Phi}_\epsilon$. By construction of f_D (cp. Proposition A3), we have

$$\operatorname{size}(\widehat{\Phi}_\epsilon) \ge M \ge \mathcal{N}(f_D, D^{\alpha^* r} \epsilon) \ge (D^{\alpha^* r} \epsilon)^{-\gamma D / r},$$

whenever $D^{\alpha^* r} \epsilon \le \bar{\epsilon}$. On the other hand, we can also estimate (cp. equation (2.2) in Section 2.1.2),

$$\operatorname{size}(\widehat{\Phi}_\epsilon) \le 2 \|W_D\|_0 + 2 \operatorname{size}(\Phi_\epsilon) \le 2D \operatorname{size}(\Phi_\epsilon) + 2 \operatorname{size}(\Phi_\epsilon) \le 4D \operatorname{size}(\Phi_\epsilon).$$

Combining these bounds, we conclude that

$$\operatorname{size}(\Phi_\epsilon) \ge \frac{1}{2D} (D^{\alpha^* r} \epsilon)^{-\gamma D / r} = \frac{1}{2} (D^{\alpha^* r + r/\gamma D} \epsilon)^{-\gamma D / r},$$

holds for any neural network representation of $\mathcal{S}_\epsilon$, whenever $D^{\alpha^* r}\epsilon \leq \overline{\epsilon}$. And hence

$$\mathrm{cmplx}(\mathcal{S}_\epsilon) \geq \frac{1}{2}(D^{\alpha^* r + r/\gamma D}\epsilon)^{-\gamma D/r}, \tag{A.14}$$

whenever $D^{\alpha^* r}\epsilon \leq \overline{\epsilon}$. By Lemma A6 below, taking the supremum on the right over all admissible D implies the lower bound

$$\mathrm{cmplx}(\mathcal{S}_\epsilon) \geq \exp(c\epsilon^{-1/(\alpha^*+\delta/2)r}), \quad \forall \epsilon \leq \widetilde{\epsilon},$$

where $\widetilde{\epsilon}, c > 0$ depend only on $\alpha^*, r, \overline{\epsilon}(r), \delta$ and γ. Recalling our choice of $\alpha^* = 1 + \alpha + \delta/2$, and the fact that the constant $\overline{\epsilon} = \overline{\epsilon}(r)$ depends only on r, while γ is universal, it follows that

$$\mathrm{cmplx}(\mathcal{S}_\epsilon) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r}), \quad \forall \epsilon \leq \widetilde{\epsilon},$$

with $\widetilde{\epsilon}, c > 0$ depending only on α, r, δ. Up to a change in notation, this is the claimed complexity bound. □

The following lemma addresses the optimization of the lower bound in (A.14):

LEMMA A6. Let $r \in \mathbb{N}$, and $\alpha^*, \overline{\epsilon}, \gamma > 0$ be given. Assume that

$$\mathrm{cmplx}(\mathcal{S}_\epsilon) \geq \frac{1}{2}\left(D^{\alpha^* r + r/\gamma D}\epsilon\right)^{-\gamma D/r},$$

for any D of the form $D = 2^k$, $k \in \mathbb{N}$, and whenever $D^{\alpha^* r}\epsilon \leq \overline{\epsilon}$. Fix a small parameter $\delta > 0$. There exist $\widetilde{\epsilon}, c > 0$, depending only on $r, \alpha^*, \gamma, \overline{\epsilon}, \delta$, such that

$$\mathrm{cmplx}(\mathcal{S}_\epsilon) \geq \exp\left(c\epsilon^{-1/(\alpha^*+\delta)r}\right), \quad \forall \epsilon \leq \widetilde{\epsilon}.$$

◊

Proof. Write $\overline{\epsilon} = e^{-\beta}$ for $\beta \in \mathbb{R}$. Fix a small parameter $\delta > 0$. Since we restrict attention to $\epsilon \leq \widetilde{\epsilon}$, we have

$$\left(e^{\beta}\epsilon\right)^{-1/(\alpha^*+\delta)r} \geq \left(e^{\beta}\widetilde{\epsilon}\right)^{-1/(\alpha^*+\delta)r} = 1,$$

provided that $\widetilde{\epsilon} \leq \overline{\epsilon}$. Given $\epsilon \leq \widetilde{\epsilon}$, choose $k \in \mathbb{N}$, such that

$$2^{k-1} < \left(e^{\beta}\epsilon\right)^{-1/(\alpha^*+\delta)r} \leq 2^k.$$

Let $D = 2^k$. Note that this defines a function $D = D(\epsilon)$. For any ϵ, we can write

$$D(\epsilon) = \xi \ \left(e^{\beta}\epsilon\right)^{-1/(\alpha^*+\delta)r}, \tag{A.15}$$

for some $\xi \in (1/2, 1]$. We now note that for $\epsilon \leq \widetilde{\epsilon}$,

$$\frac{1}{\gamma D} = \frac{\left(e^{\beta}\epsilon\right)^{1/(\alpha^*+\delta)r}}{\gamma\xi} \leq \frac{2\left(e^{\beta}\widetilde{\epsilon}\right)^{1/(\alpha^*+\delta)r}}{\gamma}.$$

Decreasing the size of $\widetilde{\epsilon} = \widetilde{\epsilon}(r, \gamma, \alpha^*, \overline{\epsilon}, \delta)$ further, we can ensure that for $\epsilon \leq \widetilde{\epsilon}$,

$$\frac{1}{\gamma D(\epsilon)} \leq \frac{2\left(e^{\beta}\widetilde{\epsilon}\right)^{1/(\alpha^*+\delta)r}}{\gamma} \leq \frac{\delta}{2}.$$

Note also that $2^{r/\gamma D} \leq 2^{\delta r/2} \leq D^{\delta r/2}$ for any D of the form $D = 2^k$, $k \in \mathbb{N}$. It thus follows that for any given $\epsilon \leq \widetilde{\epsilon}$, and with our particular choice of $D = D(\epsilon)$ satisfying (A.15), we have

$$2^{r/\gamma D} D^{\alpha^* r + r/\gamma D} \epsilon \leq D^{(\alpha^*+\delta)r} \epsilon = e^{-\beta} \xi^{(\alpha^*+\delta)r} \leq e^{-\beta}.$$

Note that this in particular implies that $D^{\alpha^* r} \epsilon \leq e^{-\beta} = \overline{\epsilon}$. We conclude that

$$\begin{aligned}
\mathrm{cmplx}(\mathcal{S}_\epsilon) &\geq \frac{1}{2} (D^{\alpha^* r + r/\gamma D} \epsilon)^{-\gamma D/r} \\
&= (2^{r/\gamma D} D^{\alpha^* r + r/\gamma D} \epsilon)^{-\gamma D/r} \\
&\geq (D^{(\alpha^*+\delta)r} \epsilon)^{-\gamma D/r} \\
&\geq \exp\left(\frac{\beta \gamma D}{r}\right) \\
&= \exp\left(\frac{\beta \gamma \xi e^{-1/(\alpha^*+\delta)r}}{r} \epsilon^{-1/(\alpha^*+\delta)r}\right) \\
&\geq \exp\left(\frac{\beta \gamma e^{-1/(\alpha^*+\delta)r}}{2r} \epsilon^{-1/(\alpha^*+\delta)r}\right).
\end{aligned}$$

Upon defining $c = c(r, \gamma, \alpha^*, \overline{\epsilon}, \delta)$ as

$$c := \frac{\beta \gamma e^{-1/(\alpha^*+\delta)r}}{2r},$$

we obtain the lower bound

$$\mathrm{cmplx}(\mathcal{S}_\epsilon) \geq \exp\left(c \epsilon^{-1/(\alpha^*+\delta)r}\right), \quad \forall \epsilon \leq \overline{\epsilon}.$$

This concludes the proof of Lemma A6. □

In the lemma below, we provide a simple result on Fréchet differentiability which was used in our proof of Proposition 2.11:

LEMMA A7 (Fréchet differentiability of a series). Assume that we are given a bounded family of functions $f_D \in C^r(\mathbb{R}^D)$ indexed by integers $D = 2^k$, $k \in \mathbb{N}$, such that $\|f_D\|_{C^r(\mathbb{R}^D)} \leq 1$ for all D. Let $e_1^*, e_2^*, \cdots : \mathcal{X} \to \mathbb{R}$ be a sequence of linear functionals, such that $\|e_j^*\|_{\mathcal{X}^*} \leq M$ for all $j \in \mathbb{N}$. Let $c, \alpha > 0$ be given, and assume that $\alpha^* > 1 + \alpha$. Then, the functional $\mathcal{S}^\dagger : \mathcal{X} \to \mathbb{R}$ defined by the series

$$\mathcal{S}^\dagger(u) := \sum_{k=1}^{\infty} 2^{-\alpha^* r k} f_{2^k}\left(c 2^{\alpha k} (e_{2^k}^*(u), \ldots, e_{2^{k+1}-1}^*(u))\right),$$

is r-times Fréchet differentiable. ◊

Proof. By assumption on f_{2^k} and the linearity of the functionals e_j^*, each nonlinear functional $\mathcal{F}_k(u) := f_{2^k}\left(c 2^{\alpha k} (e_{2^k}^*(u), \ldots, e_{2^{k+1}-1}^*(u))\right)$ in the series defining $\mathcal{S}^\dagger$ is r-times continuously differentiable. Fixing $u \in \mathcal{X}$, let us denote $x = c 2^{\alpha k} (e_{2^k}^*(u), \ldots, e_{2^{k+1}-1}^*(u))$. The ℓth total derivative $d^\ell \mathcal{F}_k$ of $\mathcal{F}_k$ ($\ell \leq r$) is

given by

$$d^\ell \mathcal{F}_k(u)[v_1, \dots, v_\ell] = c^\ell 2^{\alpha \ell k} \sum_{j_1, \dots, j_\ell = 1}^{2^k} \frac{\partial^\ell f_{2^k}(x)}{\partial x_{j_1} \dots \partial x_{j_\ell}} \prod_{s=1}^{\ell} e^*_{2^k + j_s - 1}(v_s).$$

By assumption, we have

$$\left| \frac{\partial^\ell f_{2^k}(x)}{\partial x_{j_1} \dots \partial x_{j_\ell}} \right| \leq 1.$$

Since the sum over $j_1, \dots, j_\ell$ has $2^{k\ell}$ terms, and since the functionals are bounded $\|e_j^*\| \leq M$ by assumption, we can now readily estimate the operator norm $\|d^\ell \mathcal{F}_k(u)\|$ for $\ell \leq r$ by

$$\|d^\ell \mathcal{F}_k(u)\| \leq c^\ell 2^{\alpha \ell k} 2^{k\ell} M^\ell \leq (cM)^r 2^{(\alpha+1)rk}.$$

In particular, for any $\ell \leq r$, the series

$$\sum_{k=1}^{\infty} 2^{-\alpha^* rk} \|d^\ell \mathcal{F}_k(u)\| \leq \sum_{k=1}^{\infty} 2^{-[\alpha^* - (\alpha+1)]rk} (cM)^r < \infty,$$

is uniformly convergent. Thus, $\mathcal{S}^\dagger$ is a uniform limit of r-times continuously differentiable functionals, all of whose derivatives of order $\ell \leq r$ are also uniformly convergent. From this, it follows that $\mathcal{S}^\dagger$ is itself r-times continuously Fréchet differentiable. □

A.4 *Proof of Corollary* 2.12

Let $\mathcal{Y} = \mathcal{Y}(\Omega; \mathbb{R}^p)$ be a function space with continuous embedding in $C(\Omega; \mathbb{R}^p)$. We will only consider the case $p = 1$, the case $p > 1$ following by an almost identical argument; Let $\phi \neq 0 \in \mathcal{Y}$ be a nontrivial function. Since $\mathcal{Y} = \mathcal{Y}(\Omega)$ is continuously embedded in $C(\Omega)$, it follows that point evaluation $\mathrm{ev}_y(\phi) = \phi(y)$ is continuous. Given that ϕ is nontrivial, there exists $y \in \Omega$, such that $\mathrm{ev}_y(\phi) \neq 0$. We may wlog assume that $\mathrm{ev}_y(\phi) = \phi(y) = 1$. Let $\mathcal{F}^\dagger : \mathcal{X} \to \mathbb{R}$ be a functional exposing the curse of parametric complexity, as constructed in Theorem 2.11. We define an r-times Fréchet differentiable operator $\mathcal{S}^\dagger : \mathcal{X} \to \mathcal{Y}$ by $\mathcal{S}^\dagger(u) := \mathcal{F}^\dagger(u)\phi$.

The claim now follows immediately by observing that

$$\mathrm{ev}_y \circ \mathcal{S}^\dagger(u) = \mathcal{F}^\dagger(u), \quad \forall u \in \mathcal{X},$$

and by noting that if $\mathcal{S}_\epsilon : \mathcal{X} \to \mathcal{Y}$ is an operator of neural network-type, then $\mathrm{ev}_y \circ \mathcal{S}_\epsilon : \mathcal{X} \to \mathbb{R}$ is a functional of neural network-type, and by assumption, with $C := \|\mathrm{ev}_y\|_{\mathcal{Y} \to \mathbb{R}}$,

$$\begin{aligned} \sup_{u \in K} |\mathcal{F}^\dagger(u) - \mathrm{ev}_y \circ \mathcal{S}_\epsilon(u)| &= \sup_{u \in K} |\mathrm{ev}_y \circ \mathcal{S}^\dagger(u) - \mathrm{ev}_y \circ \mathcal{S}_\epsilon(u)| \\ &\leq \sup_{u \in K} C \|\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon(u)\|_{\mathcal{Y}} \\ &\leq C\epsilon. \end{aligned}$$

By our choice of $\mathcal{F}^\dagger : \mathcal{X} \to \mathbb{R}$, this implies that the complexity of $\mathrm{ev}_y \circ \mathcal{S}_\epsilon$ is lower bounded by an exponential bound $\geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r})$ for some constant $c = c(\alpha, \delta, r)$. This in turn implies that

$$\mathrm{cmplx}(\mathcal{S}_\epsilon) = \sup_{y\in\Omega} \mathrm{cmplx}(\mathrm{ev}_y \circ \mathcal{S}_\epsilon) \geq \exp(c\epsilon^{-1/(\alpha+1+\delta)r}).$$

This lower bound implies the exponential lower bound of Corollary 2.12.

A.5 *Proofs of Lemmas* 2.18–2.7

A.5.1 *Proof of Lemma* 2.18

Proof (Lemma 2.18). We want to show that a PCA-Net neural operator $\mathcal{S} = \mathcal{R} \circ \Psi \circ \mathcal{E}$ is of neural network-type, and aim to estimate $\mathrm{size}(\Psi)$ in terms of $\mathrm{cmplx}(\mathcal{S})$. To this end, we assume that $\mathcal{Y} = \mathcal{Y}(\Omega; \mathbb{R}^p)$ is a Hilbert space of functions. Since $\mathcal{R}$ is by definition linear, then given an evaluation point $y \in \Omega$, the mapping $\beta \mapsto \mathcal{R}(\beta)(y) \equiv \mathrm{ev}_y \circ \mathcal{R}(\beta)$ defines a linear map $\mathrm{ev}_y \circ \mathcal{R} : \mathbb{R}^{D_\mathcal{Y}} \to \mathbb{R}^p$. We can represent $\mathrm{ev}_y \circ \mathcal{R}$ by matrix multiplication: $\mathrm{ev}_y \circ \mathcal{R}(\beta) = V_y\beta$, with $V_y \in \mathbb{R}^{p\times D_\mathcal{Y}}$. The encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^{D_\mathcal{X}}$ is linear by definition, thus we can take $\mathcal{L} := \mathcal{E}$ for the linear map in the definition of 'operator of neural network-type'. Define a neural network $\Phi_y : \mathbb{R}^{D_\mathcal{X}} \to \mathbb{R}^p$ by $\Phi_y(\alpha) = V_y\Psi(\alpha)$. Then, we have the identity

$$\mathrm{ev}_y \circ \mathcal{S}(u) = (\mathrm{ev}_y \circ \mathcal{R}) \circ \Psi(\mathcal{E}u) = \Phi_y(\mathcal{L}u),$$

for all $u \in \mathcal{X}$. This shows that $\mathcal{S}$ is of neural network-type. We now aim to estimate $\mathrm{cmplx}(\mathcal{S})$ in terms of $\mathrm{size}(\Psi)$. To this end, write $\Psi(\alpha) = [\Psi_1(\alpha), \ldots, \Psi_{D_\mathcal{Y}}(\alpha)]$ with component mappings $\Psi_j : \mathbb{R}^{D_\mathcal{X}} \to \mathbb{R}$. Let $\mathcal{J} = \left\{ j \,\middle|\, j \in \{1, \ldots, D_\mathcal{Y}\},\ \Psi_j \neq 0 \right\}$ be the subset of indices for which $\Psi_j : \mathbb{R}^{D_\mathcal{X}} \to \mathbb{R}$ is not the zero function. Define a (sparsified) matrix $\widehat{V}_y \in \mathbb{R}^{p\times D_\mathcal{Y}}$ with jth column $[\widehat{V}_y]_{:j}$ defined by

$$[\widehat{V}_y]_{:j} := \begin{cases} [V_y]_{:j}, & j \in \mathcal{J}, \\ 0 & j \notin \mathcal{J}. \end{cases}$$

Then, we have $\|\widehat{V}_y\|_0 \leq p|\mathcal{J}| \leq p\,\mathrm{size}(\Psi)$, and identity $\Phi_y(\alpha) = \widehat{V}_y\Psi(\alpha)$ for all $\alpha \in \mathbb{R}^{D_\mathcal{X}}$. Thus, using the concept of sparse concatenation (2.3), we can upper bound the complexity, $\mathrm{cmplx}(\mathcal{S})$, in terms of the $\mathrm{size}(\Psi)$ of the neural network Ψ as follows:

$$\mathrm{cmplx}(\mathcal{S}) \leq \sup_y \mathrm{size}(\Phi_y) \leq \sup_y 2\|V_y\|_0 + 2\mathrm{size}(\Psi) \leq 2(p+1)\mathrm{size}(\Psi).$$

This is the claimed lower bound on $\mathrm{size}(\Psi)$. □

A.5.2 *Proof of Lemma* 2.20

Proof (Lemma 2.20). We observe that with $D := D_\mathcal{X}$, for any $y \in \Omega$ the encoder $\mathcal{L} := \mathcal{E} : \mathcal{X} \to \mathbb{R}^D$ is linear, and

$$\mathrm{ev}_y \circ \mathcal{S}(u) = \Phi_y(\mathcal{L}u), \quad \forall u \in \mathcal{X},$$

where $\Phi_y(\alpha) = \sum_{j=1}^{D_\mathcal{Y}} \Psi_j(\alpha)\phi_j(y)$ defines a neural network, $\Phi_y : \mathbb{R}^D \to \mathbb{R}^p$. Thus, DeepONet $\mathcal{S} = \mathcal{R} \circ \Psi \circ \mathcal{E}$ is of neural network-type. To estimate the complexity, $\mathrm{cmplx}(\mathcal{S})$, we let $\mathcal{J}^2$ be the set of indices $(i,j) \in \{1, \ldots, D_\mathcal{Y}\}^2$, such that the ith component, $[\phi_j(y)]_i$, of the vector $\phi_j(y) \in \mathbb{R}^p$ is nonzero.

Let $\widehat{V}_y \in \mathbb{R}^{p\times D_\mathcal{Y}}$ be the matrix with entries $[\widehat{V}_y]_{i,j} = [\phi_j(y)]_i$, so that $\Phi_y(\alpha) = \widehat{V}_y\Psi(\alpha)$ for all α. Note that $\widehat{V}_y$ has precisely $|\mathcal{J}^2|$ nonzero entries, and that $|\mathcal{J}^2| \le \text{size}(\phi)$, since $[\phi_j(y)]_i \neq 0$ is nonzero for all $(i,j) \in \mathcal{J}^2$. Thus, it follows that

$$\text{cmplx}(\mathcal{S}) \le \sup_y \text{size}(\Phi_y) \le 2|\mathcal{J}^2| + 2\,\text{size}(\Psi) \le 2(\text{size}(\phi) + \text{size}(\Psi)).$$

□

A.5.3 *Proof of Lemma* 2.22

Proof (Lemma 2.22). To see the complexity bound, we recall that for any $y \in \Omega$, we can choose $\Phi_y(\cdot) := Q(\Psi(\cdot), y)$ to obtain the representation

$$\text{ev}_y \circ \mathcal{S}(u) = \Phi_y(\mathcal{L}u),$$

where $\mathcal{L}u \equiv \mathcal{E}(u)$ is given by the linear encoder. The composition of two neural networks $Q(\cdot, y)$ and $\Psi(\cdot)$ can be represented by a neural network of size at most $2\,\text{size}(\Psi) + 2\,\text{size}(Q(\cdot, y)) \le 2\,\text{size}(\Psi) + 2\,\text{size}(Q)$. We thus have the lower bound,

$$\text{cmplx}(\mathcal{S}) \le \sup_y \text{size}(\Phi_y) \le 2\text{size}(Q) + 2\text{size}(\Psi).$$

This shows the claim. □

A.6 *Proof of Lemma* 2.26

Proof. Our aim is to show that $\mathcal{S} : L^2(\mathbb{T};\mathbb{R}) \to \mathbb{R}$,

$$\mathcal{S}(u) := \int_\mathbb{T} \sigma(u(x))\,\mathrm{d}x,$$

is not of neural network-type. We argue by contradiction. Suppose that $\mathcal{S}$ was of neural network type. By definition, there exists a linear mapping $\mathcal{L} : \mathcal{L}^2(\mathbb{T};\mathbb{R}) \to \mathbb{R}^\ell$, and a neural network $\Phi : \mathbb{R}^\ell \to \mathbb{R}$, for some $\ell \in \mathbb{N}$ such that

$$\mathcal{S}(u) = \Phi(\mathcal{L}u). \tag{A.16}$$

In the following we will consider $\varphi_j(x) := \sin(jx)$ for $j \in \mathbb{N}$. Since $\sigma(t) \ge 0$ for all $t \in \mathbb{R}$, and $\sigma(t) > 0$ for $t > 0$, we have

$$\mathcal{S}(u) = \int_\Omega \sigma(u(x))\,\mathrm{d}x = 0 \quad \Longleftrightarrow \quad u(x) \le 0, \quad \forall x \in [0, 2\pi]. \tag{A.17}$$

Now, fix any $D > \ell$, and consider $\iota : \mathbb{R}^D \to L^2(\mathbb{T};\mathbb{R})$, $\iota\beta := \sum_{j=1}^D \beta_j \sin(jx)$. Since ι and $\mathcal{L}$ are linear mappings, it follows that

$$\mathcal{L}\circ\iota : \mathbb{R}^D \to \mathbb{R}^\ell, \quad \beta \mapsto \mathcal{L}\iota\beta,$$

is a linear mapping. Represent this linear mapping by a matrix $W \in \mathbb{R}^{\ell\times D}$. In particular, by (A.16), we have

$$\mathcal{S}(\iota\beta) = \Phi(W\beta), \quad \forall\beta \in \mathbb{R}^D. \tag{A.18}$$

Since $D > \ell$, it follows that $\ker(W) \neq \{0\}$ is nontrivial. Let $\beta \neq 0$ be an element in the kernel, and consider $u_\beta(x) := \iota\beta(x) = \sum_{j=1}^D \beta_j \sin(jx)$. Since $u_\beta(x)$ is not identically equal to 0, either $u_\beta(x)$ or

$-u_\beta(x) = u_{-\beta}(x)$ must be positive somewhere in the domain $\mathbb{T}$. Upon replacing $\beta \to -\beta$ if necessary, we may wlog assume that $u_\beta(x) > 0$ for some $x \in \mathbb{T}$. From (A.17) and (A.18), it now follows that

$$0 \neq \mathcal{S}(u_\beta) = \mathcal{S}(\iota\beta) = \Phi(W\beta) = \Phi(0) = \mathcal{S}(0) = 0.$$

A contradiction. Thus, $\mathcal{S}$ cannot be of neural network-type. □

A.7 *Proof of curse of parametric complexity for FNO, Theorem* 2.27

Building on the curse of parametric complexity for operators of neural network-type, we next show that FNOs also suffer from a similar curse, as stated in Theorem 2.27.

Proof (Theorem 2.27). Let $\mathcal{S}^\dagger : \mathcal{X} \to \mathbb{R}$ be an r-times Fréchet differentiable functional satisfying the conclusion of Theorem 2.11 (CoD for functionals of neural network-type). In the following, we show that $\mathcal{S}^\dagger$ also satisfies the conclusions of Theorem 2.27. Our proof argues by contradiction: we assume that $\mathcal{S}^\dagger$ can be approximated by a family of discrete FNOs satisfying the error and complexity bounds of Theorem 2.27, i.e.,

1. *Complexity bound:* There exist constant $c > 0$, such that the discretization parameter $N_\epsilon \in \mathbb{N}$, and the total number of nonzero parameters $\text{size}(\mathcal{S}_\epsilon^{N_\epsilon})$ are bounded by $N_\epsilon, \text{size}(\mathcal{S}_\epsilon^{N_\epsilon}) \leq \exp(c\epsilon^{-1/(1+\alpha+\delta)r})$, for all $\epsilon \leq \overline{\epsilon}$.
2. *Accuracy:* We have
$$\sup_{u\in K} \left|\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon^{N_\epsilon}(u)\right| \leq \epsilon, \quad \forall\, \epsilon > 0.$$

Then we show that this implies the existence of a family of operators of *neural network-type* $\widetilde{\mathcal{S}}_\epsilon$, which satisfy for some $\delta' > 0$, and for all sufficiently small $\epsilon > 0$,

- complexity bound $\text{cmplx}(\widetilde{\mathcal{S}}_\epsilon) \leq \exp(c\epsilon^{-1/(1+\alpha+\delta')r})$,
- and error estimate $\max_{u\in K} \left|\mathcal{S}^\dagger(u) - \widetilde{\mathcal{S}}_\epsilon(u)\right| \leq \epsilon$,

with $c > 0$ a potentially different constant. By choice of $\mathcal{S}^\dagger$, the existence of $\widetilde{\mathcal{S}}_\epsilon$ is ruled out by Theorem 2.11, providing the desired contradiction.

In the following, we discuss the construction of $\widetilde{\mathcal{S}}_\epsilon$: Let $\mathcal{S}_\epsilon^{N_\epsilon}$ be a family of FNOs satisfying (1) and (2) above. Fix $\epsilon > 0$ for the moment. To simplify notation, we will write $N = N_\epsilon$, in the following. We recall that, by definition, the discretized FNO $\mathcal{S}_\epsilon^N$ can be written as a composition $\mathcal{S}_\epsilon^N = \widetilde{\mathcal{Q}} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{R}$, where

$$\mathcal{R} : u(x) \mapsto \chi(x_{j_1,\ldots,j_d}, u(x_{j_1,\ldots,j_d})),$$

and

$$\widetilde{\mathcal{Q}} : \left(v_{j_1,\ldots,j_d}\right) \mapsto \frac{1}{N^d} \sum_{j_1,\ldots,j_d} q(y_{j_1,\ldots,j_d}, v_{j_1,\ldots,j_d}),$$

are defined by neural networks χ and q, respectively, $\mathcal{L}_\ell$ is of the form

$$\mathcal{L}_\ell : v \mapsto \sigma\left(W_\ell v + \mathcal{F}_N^{-1}\left(\widehat{T}_\ell \mathcal{F}_N v\right) + b_\ell\right), \quad v = \left(v_{j_1,\ldots,j_d}\right)_{j_1,\ldots,j_d=1}^N,$$

with $W_\ell \in \mathbb{R}^{d_v \times d_v}$, Fourier multiplier $\widehat{T}_\ell = \{[\widehat{T}_\ell]_k\}_{\|k\|_{\ell^\infty} \le k_{\max}}$, where $[\widehat{T}_\ell]_k \in \mathbb{C}^{d_v \times d_v}$, and $\mathcal{F}_N$ ($\mathcal{F}_N^{-1}$) denote discrete (inverse) Fourier transform, and where the bias b is determined by its Fourier coefficients $\widehat{b}_k$, $\|k\|_{\ell^\infty} \le k_{\max}$. We also recall that the size of $\mathcal{S}_\epsilon^N$ is given by the total number of nonzero components, i.e.,

$$\mathrm{size}(\mathcal{S}_\epsilon^N) = \mathrm{size}(\chi) + \mathrm{size}(q) + \sum_{\ell=1}^{L} \left\{ \|W_\ell\|_0 + \|\widehat{T}_\ell\|_0 + \sum_{\|k\|_{\ell^\infty} \le k_{\max}} \|\widehat{b}_k\|_0 \right\}.$$

We now observe that, after flattening the tensor

$$\left(v_{j_1,\dots,j_d}\right) \in \mathbb{R}^{N \times \cdots \times N \times d_v} \simeq \mathbb{R}^{d_v N^d},$$

the (linear) mapping $v \mapsto W_\ell v$ can be represented by multiplication against a sparse matrix with at most $\|W_\ell\|_0 N^d$ nonzero entries. For the nonlocal operator $\mathcal{F}_N^{-1}\widehat{T}_\ell \mathcal{F}_N$, we note that for $v \in \mathbb{R}^{d_v N^d}$, the number κ of components (channels) that need to be considered is bounded by $\kappa \le \min(d_v, \|\widehat{T}_\ell\|_0) \le \|\widehat{T}_\ell\|_0$. Discarding any channels that are zeroed out by $\widehat{T}_\ell$, a naive implementation of $\mathcal{F}_N^{-1}\widehat{T}_\ell \mathcal{F}_N$ thus amounts to a matrix representation of a linear mapping $\mathbb{R}^{\kappa N^d} \to \mathbb{R}^{\kappa N^d}$, requiring at most $\kappa^2 N^{2d} \le \|\widehat{T}_\ell\|_0^2 N^{2d}$ nonzero components. Thus, each discretized hidden layer $\mathcal{L}_\ell$ can be represented *exactly* by an ordinary neural network layer $L_\ell = \sigma(A_\ell v + c_\ell)$ with matrix $A_\ell \in \mathbb{R}^{d_v N^d \times d_v N^d}$ and bias $c_\ell \in \mathbb{R}^{d_v N^d}$, satisfying the following bounds on the number of nonzero components:

$$\|A_\ell\|_0 \le \|W_\ell\|_0 N^d + \|\widehat{T}_\ell\|_0^2 N^{2d}, \quad \|c_\ell\|_0 \le \sum_{|k| \le k_{\max}} \|[\widehat{b}_\ell]_k\|_0 N^d.$$

Similarly, the input and output layers $\mathcal{R}$ and $\widetilde{\mathcal{Q}}$ can be represented *exactly* by ordinary neural networks $R : \mathbb{R}^{kN^d} \to \mathbb{R}^{d_v N^d}$ and $Q : \mathbb{R}^{d_v N^d} \to \mathbb{R}$, with

$$\mathrm{size}(R) \le N^d \mathrm{size}(\chi), \quad \mathrm{size}(Q) \le N^d \mathrm{size}(q),$$

obtained by parallelization of χ, resp. q, at each grid point. Given the above observations, we conclude that, with canonical identification $\mathbb{R}^{N \times \cdots \times N \times k} \simeq \mathbb{R}^{kN^d}$ and $\mathbb{R}^{N \times \cdots \times N \times p} \simeq \mathbb{R}^{pN^d}$, the discretized FNO $\mathcal{S}_\epsilon^N$ can be represented by an ordinary neural network $\Phi : \mathbb{R}^{kN^d} \to \mathbb{R}^{pN^d}$, $\Phi = Q \circ L_L \circ \cdots \circ L_1 \circ R$, with

$$\begin{aligned} \mathrm{size}(\Phi) &\le \sum_{\ell=1}^{L} \left\{ \|W_\ell\|_0 N^d + \|\widehat{T}_\ell\|_0^2 N^{2d} + \sum_{|k| \le k_{\max}} \|\widehat{b}_k\|_0 N^d \right\} \\ &\quad + N^d \mathrm{size}(\chi) + N^d \mathrm{size}(q) \\ &\le N^{2d} \mathrm{size}(\mathcal{S}_\epsilon^N)^2. \end{aligned}$$

By assumption on $\mathcal{S}_\epsilon^N$ (for which we aim to show that it leads to a contradiction), we have $\mathrm{size}(\mathcal{S}_\epsilon^N) \le \exp(c\epsilon^{-1/(1+\alpha+\delta)r})$, and $N_\epsilon \le \exp(c\epsilon^{-1/(1+\alpha+\delta)r})$. It follows that

$$\mathrm{size}(\Phi) \le N^{2d} \mathrm{size}(\mathcal{S}_\epsilon^N)^2 \le \exp((2d+2)c\epsilon^{-1/(1+\alpha+\delta)r}).$$

In addition, Φ trivially defines an operator of neural network-type, $\widetilde{\mathcal{S}}_\epsilon : C^s(\Omega; \mathbb{R}^k) \to \mathbb{R}$, by

$$\widetilde{\mathcal{S}}_\epsilon(u) := \Phi\left((u(x_{j_1,\dots,j_d}))_{j_1,\dots,j_d=1}^N\right).$$

To see this, we simply note that the point-evaluation mapping $\mathcal{L} : u \mapsto u(x_{j_1,\ldots,j_d})_{j_1,\ldots,j_d=1}^{N}$ is linear, and hence we have the representation

$$\widetilde{\mathcal{S}}_\epsilon(u) = \Phi(\mathcal{L}u).$$

By the above construction, we have $\widetilde{\mathcal{S}}_\epsilon(u) \equiv \mathcal{S}_\epsilon^{N_\epsilon}(u)$ for all input functions u.

To summarize, assuming that a family of FNOs $\mathcal{S}_\epsilon^{N_\epsilon}$ exists, satisfying (1) and (2) above, we have constructed a family $\widetilde{\mathcal{S}}_\epsilon$ of operators of neural network type, with

$$\sup_{u\in K}\left|\mathcal{S}^\dagger(u) - \widetilde{\mathcal{S}}_\epsilon(u)\right| = \sup_{u\in K}\left|\mathcal{S}^\dagger(u) - \mathcal{S}_\epsilon^{N_\epsilon}(u)\right| \le \epsilon,$$

and

$$\mathrm{cmplx}(\widetilde{\mathcal{S}}_\epsilon) \le \mathrm{size}(\Phi) \le \exp(c''\epsilon^{-1/(1+\alpha+\delta)r}),$$

where $c'' = (2d+2)c > 0$ is a constant independent of ϵ.

By Theorem 2.11, and fixing any $0 < \delta' < \delta$, we also have the following lower bound for all sufficiently small ϵ:

$$\exp(c'\epsilon^{-1/(1+\alpha+\delta')r}) \le \mathrm{cmplx}(\widetilde{\mathcal{S}}_\epsilon),$$

where $c' > 0$ is independent of ϵ. From the above two-sided bounds, it thus follows that

$$\exp(c'\epsilon^{-1/(1+\alpha+\delta')r}) \le \mathrm{cmplx}(\widetilde{\mathcal{S}}_\epsilon) \le \exp(c''\epsilon^{-1/(1+\alpha+\delta)r}),$$

for all ϵ sufficiently small, and where by our choice of δ', δ: $0 < 1+\alpha+\delta' < 1+\alpha+\delta$ and $c', c'' > 0$ are constants. Since $\epsilon^{-1/(1+\alpha+\delta')r}$ grows faster than $\epsilon^{-1/(1+\alpha+\delta)r}$ as $\epsilon \to 0$, this leads to the desired contradiction. We thus conclude that a family $\mathcal{S}_\epsilon^{N_\epsilon}$ of discretized FNOs as assumed above cannot exist. This concludes the proof. □

Appendix B. Short-time existence of C^r-solutions

The proof of short-time existence of solutions of the Hamilton–Jacobi equation (HJ) is based on the following Banach space implicit function theorem:

THEOREM B1 (Implicit Function Theorem, see e.g., (Chow & Hale, 2012, section 2.2)). Let $U \subset X$, $V \subset Y$ be open subsets of Banach spaces X and Y. Let

$$F : U \times V \to Z, \quad (u, v) \mapsto F(u, v),$$

be a C^p-mapping (p-times continuously Fréchet differentiable). Assume that there exist $(u_0, v_0) \in U \times V$ such that $F(u_0, v_0) = 0$, and such that the Jacobian with respect to v, evaluated at the point (u_0, v_0),

$$D_v F(u_0, v_0) : X \to Z,$$

is a linear isomorphism. Then there exists a neighbourhood $U_0 \subset U$ of u_0, and a C^p-mapping $\psi : U_0 \to V$, such that

$$F(u, \psi(u)) = 0, \quad \forall\, u \in U_0.$$

Furthermore, ψ is unique, in the sense that for any $u \in U_0$, $F(u, v) = 0$ implies $v = \psi(u)$. ◊

As a first step towards proving short-time existence for (HJ), we prove that under the no-blowup Assumption 3.1, the semigroup $\Psi_t^\dagger$ (3.7) exists for any $t \geq 0$.

Proof (Lemma 3.3). By classical ODE theory, it is immediate that for any initial data $(q_0, p_0, z_0) \in \Omega \times \mathbb{R}^d \times \mathbb{R}$ a maximal solution of the ODE system (3.4) exists, and is unique, over a short time-interval. It thus remains to prove that this solution in fact exists globally, i.e., that the solution does not escape to infinity in finite time. Since z solves $\dot{z} = \mathcal{L}(q,p)$, with a right-hand side that only depends on q and p, it will suffice to prove that the Hamiltonian trajectory $\dot{q} = \nabla_p H(q,p)$, $\dot{p} = -\nabla_q H(q,p)$ with initial data (q_0, p_0) exists globally in time. To this end, we note that, by Assumption 3.1, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}(1+|p|^2) = 2p \cdot \dot{p} = -2p \cdot \nabla_q H(q,p) \leq 2L_H(1+|p|^2).$$

Gronwall's lemma thus implies that $|p| \leq \sqrt{1+|p_0|^2}\exp(L_H t)$ remains bounded for all $t \geq 0$. This shows that blow-up cannot occur in the p-variable in a finite time. On the other hand, the right-hand side of the ODE system is periodic in q, and hence (by the bound on p) blow-up is also ruled out for q. In particular, Assumption 3.1 ensures that the Hamiltonian trajectory $t \mapsto (q(t), p(t))$ exists globally in time. As argued above, this in turn implies the global existence of the flow map $t \mapsto \Psi_t^\dagger$. □

We now apply the Implicit Function Theorem B.1 to prove Lemma 3.4.

Proof (Lemma 3.4). Let $X = C^r_{\mathrm{per}}(\Omega) \times \mathbb{R}^d \times \mathbb{R}$ with $r \geq 2$. Define

$$F : X \times \mathbb{R}^d \to \mathbb{R}^d, \quad (u_0, q, t; q_0) \mapsto q - q_t(q_0, \nabla_q u_0(q_0)),$$

i.e., we set

$$F(u_0, q, t; q_0) := q - q_t(q_0, \nabla_q u_0(q_0)),$$

with $q_t : \Omega \times \mathbb{R}^d \to \Omega$, $(q_0, p_0) \mapsto q_t(q_0, p_0)$ the spatial characteristic mapping of the Hamiltonian system (3.4) at time $t \geq 0$, where, recall, we have assumed that $u_0 \in \mathcal{F}$. Under Assumption 3.1, the spatial characteristic mapping, and hence F, is well-defined for any $t \geq 0$ (see Lemma 3.3). Since $H \in C^{r+1}(\Omega \times \mathbb{R}^d)$, the mapping $\Omega \times \mathbb{R}^d \times \mathbb{R} \to \Omega$, $(q_0, p_0, t) \mapsto q_t(q_0, p_0)$ is C^r. The mapping $F_1 : \Omega \times C^r_{\mathrm{per}}(\Omega) \to \mathbb{R}^d$, $(q_0, u_0) \mapsto F_1(q_0, u_0) := \nabla_q u_0(q_0)$ is C^{r-1} in the first argument, and is a continuous linear mapping in the second argument (and hence infinitely Frechet differentiable in the second argument). Hence, the composition

$$(q_0, u_0) \mapsto (q_0, \nabla_q u_0(q_0)) \mapsto q_t(q_0, \nabla_q u_0(q_0)),$$

is a C^{r-1} mapping. And, as a consequence, the mapping $F : X \times \mathbb{R}^d \to \mathbb{R}^d$ is a C^{r-1} mapping.

Since

$$F(u_0, q, t; q_0) = q - q_t(q_0, \nabla_q u_0(q_0)),$$

we clearly have

$$F(u_0, q_0, 0; q_0) = 0,$$

for any $q_0 \in \Omega$ and $u_0 \in C^r_{\mathrm{per}}(\Omega)$, and the derivative with respect to the last argument

$$D_{q_0} F(u_0, q_0, 0; q_0) : \mathbb{R}^d \to \mathbb{R}^d,$$

is given by

$$D_{q_0}F(u_0, q_0, 0; q_0) = D_{q_0}\left[q - q_0\right] = -\mathbf{1}_{d\times d},$$

which defines an isomorphism $\mathbb{R}^d \to \mathbb{R}^d$.

By the implicit function theorem, for any $\overline{q}_0 \in \Omega$ and $\overline{u}_0 \in C^r_{\text{per}}(\Omega)$, there exist $\epsilon = \epsilon(\overline{q}_0, \overline{u}_0), r = r(\overline{q}_0, \overline{u}_0), t^* = t^*(\overline{q}_0, \overline{u}_0) > 0$, and a mapping

$$\psi_{\overline{q}_0,\overline{u}_0} : B_\epsilon(\overline{q}_0) \times B_r(\overline{u}_0) \times [0, t^*) \to \mathbb{R}^d,$$

such that $F(q, u_0, t; q_0) = 0$ for

$$(q, u_0, t) \in B_\epsilon(\overline{q}_0) \times B_r(\overline{u}_0) \times [0, t^*),$$

if, and only if,

$$q_0 = \psi_{\overline{q}_0,\overline{u}_0}(q, u_0, t).$$

Fix $\overline{u}_0 \in C^r_{\text{per}}(\Omega)$ for the moment. Since Ω is compact, we can choose a finite number of points $\overline{q}_0^{(1)}, \ldots, \overline{q}_0^{(m)}$, such that

$$\Omega \subset \bigcup_{j=1}^m B_{\epsilon(\overline{q}_0^j,\overline{u}_0)}(\overline{q}_0^j).$$

Let

$$t^*(\overline{u}_0) := \min_{j=1,\ldots,m} t^*(\overline{q}_0^j, \overline{u}_0),$$

$$r(\overline{u}_0) := \min_{j=1,\ldots,m} r(\overline{q}_0^j, \overline{u}_0).$$

Due to the uniqueness property of each $\psi_{\overline{q}_0^j,\overline{u}_0}, j = 1, \ldots, m$, all of these mappings have the same values on overlapping domains. Hence, we can combine them into a single map

$$\psi_{\overline{u}_0} : \Omega \times B_{r(\overline{u}_0)}(\overline{u}_0) \times [0, t^*(\overline{u}_0)) \to \mathbb{R}^d.$$

Furthermore, since $\psi_{\overline{q}_0^j,\overline{u}_0} \in C^{r-1}$, we also have $\psi_{\overline{u}_0} \in C^{r-1}$. Similarly, we can cover the compact set $\mathcal{F} \subset C^r_{\text{per}}(\Omega)$ by a finite number of open balls $B_{r(\overline{u}_0^{(k)})}(\overline{u}_0^{(k)}), k = 1, \ldots, K$,

$$\mathcal{F} \subset \bigcup_{k=1}^K B_{r(\overline{u}_0^{(k)})}(\overline{u}_0^{(k)}).$$

Setting $T^* := \min_{k=1,\ldots,K} t^*(\overline{u}_0^{(k)}) > 0$, the uniqueness property of $\psi_{\overline{u}_0^{(k)}}$ again implies that these mappings agree on overlapping domains. Hence, we can combine them into a global map, and obtain a map

$$\psi : \Omega \times \mathcal{F} \times [0, T^*) \to \mathbb{R}^d,$$

which satisfies $F(q,u_0,t;\psi(q,u_0,t))=0$ for all $q\in\mathcal{Q}$, $u_0\in\mathcal{F}$ and $t<T^*$. Furthermore, this ψ is still a C^{r-1} map and it is unique, in the sense that

$$F(q,u_0,t;q_0)=0 \quad \Leftrightarrow \quad q_0=\psi(q,u_0,t), \quad \forall\, q\in\mathcal{Q},\ u_0\in C^r_{\mathrm{per}}(\mathcal{Q}),\ t<T^*,$$

i.e., for any $u_0\in\mathcal{F}$ and $t\in[0,T^*)$, we have $q=q_t(q_0,\nabla_q u_0(q_0))$ if and only if $q_0=\psi(q,u_0,t)$. In particular, this shows that for any $u_0\in\mathcal{F}$, $t\in[0,T^*)$, the C^{r-1}-mapping

$$\Phi_t^\dagger(\,\cdot\,;u_0):\mathcal{Q}\to\mathcal{Q}, \quad q_0\mapsto\Phi_t^\dagger(q_0;u_0):=q_t(q_0,\nabla_q u_0(q_0)),$$

is invertible, with inverse $q\mapsto\psi(q,u_0,t)\in C^{r-1}$. This implies the claim. □

Next, we apply Lemma 3.4 to prove short-time existence of solutions for (HJ).

Proof (Proposition 3.2). Let T^* be the maximal time such that the spatial characteristic mapping $\Phi^\dagger_{t,u_0}:\mathcal{Q}\to\mathcal{Q}$, $q_0\mapsto q_t(q_0,\nabla_q u_0(q_0))$ is invertible, for any $t\in[0,T^*)$ and for all $u_0\in\mathcal{F}$. We have $T^*>0$, by Lemma 3.4. We denote by $\Phi^\dagger_{-t,u_0}:\mathcal{Q}\to\mathcal{Q}$ the inverse $\Phi^\dagger_{-t,u_0}:=[\Phi^\dagger_{t,u_0}]^{-1}$. By the method of characteristics, a solution $u(q,t)$ of the Hamilton–Jacobi equations must satisfy

$$u(q,t)=u_0(q_0)+\int_0^t \mathcal{L}(q_\tau,p_\tau)\,\mathrm{d}\tau, \quad \mathcal{L}(q,p):=p\cdot\nabla_p H(q,p)-H(q,p), \tag{B.1}$$

where $q_0=\Phi^\dagger_{-t,u_0}(q)$, q_τ,p_τ are the trajectories of the Hamiltonian ODE (3.1) with initial data $q_0,\nabla_q u_0(q_0)$. Given a fixed $u_0\in\mathcal{F}$, we use the above expression to *define* $u:\mathcal{Q}\times[0,T^*)\to\mathbb{R}$.

We first observe that $u\in C^{r-1}(\mathcal{Q}\times[0,T^*))$, as it is the composition of C^{r-1} mappings,

$$u(q,t)=u_0(q_0)+\int_0^t \mathcal{L}(q_\tau(q_0,p_0),p_\tau(q_0,p_0))\,\mathrm{d}\tau,$$

where $q_0=\Phi_{-t,u_0}(q)$, $p_0=\nabla_q u_0(\Phi_{-t,u_0}(q))$ are C^{r-1}-functions of q. In particular, since $r\geq 2$, this implies that u is at least C^1. Evaluating $\mathrm{d}u(q_t,t)/\mathrm{d}t$ along a fixed trajectory, we find that

$$\partial_t u(q_t,t)+H(q_t,p_t)=\Big(p_t-\nabla_q u(q_t,t)\Big)\cdot D_p H(q_t,p_t). \tag{B.2}$$

Thus, to show that u is a classical solution of (HJ), it remains to show that $p_t=\nabla_q u(q_t,t)$ for all $t\in[0,T^*)$. Assume that $u\in C^2_{\mathrm{per}}$ for the moment. We first note that for the jth component of $p_t-\nabla_q u(q_t,t)$, we have (with implicit summation over repeated indices, following the 'Einstein summation rule')

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\Big[p_t^j-\partial_{q^j}u(q_t,t)\Big]=&-\partial_{q^j}H(q_t,p_t)-\partial^2_{q^j,q^k}u(q_t,t)\partial_{p^k}H(q_t,p_t)\\&-\partial_{q^j}\partial_t u(q_t,t).\end{aligned}$$

Next, we note that by the invertibility of the spatial characteristic mapping $q_0\mapsto q_t$, we can write (B.2) in the form

$$\partial_t u(q,t)=-H(q,P(q,t))+\Big(P(q,t)-\nabla_q u(q,t)\Big)\cdot D_p H(q,P(q,t)),$$

where $P(q,t) := p_t\left(\Phi^{\dagger}_{-t,u_0}(q), \nabla_q u_0(\Phi^{\dagger}_{-t,u_0}(q))\right)$. This implies that

$$\begin{aligned}\partial_{q^j}\partial_t u(q,t) &= -\partial_{q^j}H(q,P(q,t)) - \partial_{p^k}H(q,P(q,t))\partial_{q^j}P^k(q,t)\\ &\quad + \left(\partial_{q^j}P^k(q,t) - \partial^2_{q^j,q^k}u(q,t)\right)\cdot \partial_{p^k}H(q,P(q,t))\\ &\quad + \left(P^k(q,t) - \partial_{q^k}u(q,t)\right)\cdot \partial_{q^j}\left[\partial_{p^k}H(q,P(q,t))\right]\\ &= -\partial_{q^j}H(q,P(q,t)) - \partial^2_{q^j,q^k}u(q,t)\cdot \partial_{p^k}H(q,P(q,t))\\ &\quad + \left(P^k(q,t) - \partial_{q^k}u(q,t)\right)\cdot A_{j,k}(q,P(q,t)).\end{aligned}$$

We point out that on the last line, we have introduced $A_{j,k}(q,t) := \partial_{q^j}\left[\partial_{p^k}H(q,P(q,t))\right]$ which is a continuous function of q and t. Choosing now $q = q_t$, so that $P(q,t) = p_t$, we thus obtain

$$\begin{aligned}\partial_{q^j}\partial_t u(q_t,t) &= -\partial_{q^j}H(q_t,p_t) - \partial^2_{q^j,q^k}u(q_t,t)\cdot \partial_{p^k}H(q_t,p_t)\\ &\quad + \left[p^k_t - \partial_{q^k}u(q_t,t)\right]\cdot A_{j,k}(q,t).\end{aligned}$$

Substitution in the ODE for $p_t - \nabla_q u(q_t,t)$ yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[p_t - \nabla_q u(q_t,t)\right] = -A(q_t,t)\cdot\left[p_t - \nabla_q u(q_t,t)\right],$$

where $A(q_t,t)$ is the matrix with components $(A_{j,k}(q_t,t))$. Since $[p_t - \nabla_q u(q_t,t)]|_{t=0} = 0$, this implies that

$$p_t = \nabla_q u(q_t,t), \quad \forall\, t\in[0,T^*), \tag{B.3}$$

along the trajectory. At this point, the conclusion (B.3) has been obtained under the assumption that $u \in C^2_{\mathrm{per}}$, which is only ensured *a priori* if $r \geq 3$.

To prove the result also for the case $r = 2$, we can apply the above argument to smooth H^ϵ, u^ϵ_0, which approximate the given H and u_0, and for u^ϵ defined by the method of characteristics (B.1) with u_0, H replaced by the smooth approximations u^ϵ_0, H^ϵ. Then, by the above argument, for any ϵ-regularized trajectory $(q^\epsilon_t, p^\epsilon_t)$, we have

$$p^\epsilon_t = \nabla_q u^\epsilon(q^\epsilon_t, t).$$

Choosing a sequence such that $H^\epsilon \overset{C^{r+1}}{\to} H$, $u^\epsilon_0 \overset{C^r}{\to} u_0$ as $\epsilon\to 0$, the corresponding characteristic solution u^ϵ defined by (B.1) (with H^ϵ and u^ϵ_0 in place of H and u_0) converges in C^{r-1}. Since $r-1 = 1$, this implies that $p_t = \lim_{\epsilon\to 0} p^\epsilon_t = \lim_{\epsilon\to 0}\nabla_q u^\epsilon(q^\epsilon_t,t) = \nabla_q u(q_t,t)$.

Thus, we conclude that $u \in C^{r-1}(\Omega\times[0,T^*))$ defined by (B.1) is a classical solution of the Hamilton–Jacobi equations (HJ). We finally have to show that, in fact, $u \in C^r_{\mathrm{per}}(\Omega\times[0,T^*))$. C^r-differentiability in space follows readily from the fact that, by (B3), we have $\nabla_q u(q_t,t) = p_t(q_0, \nabla_q u_0(q_0))$. By the invertibility of the spatial characteristic map, this can be equivalently written in the form

$$\nabla_q u(q,t) = p_t(\Phi^{\dagger}_{-t,u_0}(q), \nabla_q u_0(\Phi^{\dagger}_{-t,u_0}(q))), \tag{B.4}$$

where on the right-hand side, $(q_0, p_0) \mapsto p_t(q_0, p_0)$, $q_0 \mapsto \nabla_q u_0(q_0)$ and $(q, t) \mapsto \Phi^{\dagger}_{-t,u_0}(q, t)$ are all C^{r-1} mappings. Thus, $\nabla_q u$ is a C^{r-1}-function. Furthermore, by (HJ), this implies that $\partial_t u = -H(q, \nabla_q u)$ is also a C^{r-1} function. This allows us to conclude that $u \in C^r_{\text{per}}(\Omega \times [0, T^*))$. The additional bound on $\|u(\cdot, t)\|_{C^r}$ follows from the trivial estimate

$$\|u(\cdot, t)\|_{C^r} \leq C \left(\|u\|_{L^\infty} + \|\nabla_q u(\cdot, t)\|_{C^{r-1}} \right),$$

combined with (B.1) and (B.4), and the fact that $q \mapsto \Phi^{\dagger}_{-t,u_0}(q)$ is a C^{r-1} mapping with continuous dependence on $u_0 \in \mathcal{F}$ and $t \in [0, T]$, and $(q_0, p_0) \mapsto p_t(q_0, p_0)$ is a C^r-mapping, so that

$$\|\nabla_q u(\cdot, t)\|_{C^{r-1}} \leq \sup_{t \in [0,T]} \sup_{u_0 \in \mathcal{F}} \|p_t(\Phi^{\dagger}_{-t,u_0}(\cdot), \nabla_q u_0(\Phi^{\dagger}_{-t,u_0}(\cdot)))\|_{C^{r-1}} < \infty,$$

for any initial data $u_0 \in \mathcal{F}$. □

Appendix C. Reconstruction from scattered data

The purpose of this appendix is to prove Proposition 4.2. This is achieved through three lemmas, followed by the proof of the proposition itself, employing these lemmas.

The first lemma is the following special case of the Vitali covering lemma, a well-known geometric result from measure theory (see e.g. (Stein & Shakarchi, 2009, Lemma 1.2)):

LEMMA C1. (Vitali covering). If $h > 0$ and $Q = \{q^1, \ldots, q^N\}$ is a set for which the domain

$$\Omega \subset \bigcup_{j=1}^{N} B_h(q^j),$$

is contained in the union of balls of radius h around the q^j, then there exists a subset $Q' = \{q^{i_1}, \ldots, q^{i_m}\}$ such that $B_h(q^{i_k}) \cap B_h(q^{i_\ell}) = \emptyset$ for all k, ℓ, and

$$\Omega \subset \bigcup_{k=1}^{m} B_{3h}(q^{i_k}).$$

◊

REMARK C2. Given $Q = \{q^1, \ldots, q^N\}$, the proof of Lemma C1 (see e.g. (Stein & Shakarchi, 2009, Lemma 1.2)) shows that the subset $Q' \subset Q$ of Lemma C1 can be found by the following greedy algorithm, which proceeds by iteratively adding elements to Q' (the following is in fact the basis for Algorithm 1 in the main text):

1. Start with $j_1 = 1$ and $Q'_1 = \{q^1\}$.
2. Iteration step: given $Q'_m = \{q^{j_1}, \ldots, q^{j_m}\} \subset Q$, check whether there exists $q^k \in Q$, such that

$$B_h(q^k) \cap \bigcup_{\ell=1}^{m} B_h(q^{j_\ell}) = \emptyset.$$

- If yes: define $j_{n+1} := k$ and $Q'_{m+1} := \{q^{j_1}, \dots, q^{j_{m+1}}\}$.
- If not: terminate the algorithm and set $Q' := Q'_m = \{q^{j_1}, \dots, q^{j_m}\}$.

$\Diamond$

Based on Lemma C.1, we can now state the following basic result:

LEMMA C3. Given a set $Q = \{q^1, \dots, q^N\} \subset \Omega$ with fill distance $h_{Q,\Omega}$, the subset $Q' \subset Q$ determined by the pruning Algorithm 1 has fill distance $h_{Q',\Omega} \leq 3h_{Q,\Omega}$ and separation distance $\rho_{Q'} \geq h_{Q,\Omega}$, and Q' is quasi-uniform with distortion constant $\kappa = 3$, i.e.,

$$\rho_{Q'} \leq h_{Q',\Omega} \leq 3\rho_{Q'}.$$

$\Diamond$

Proof. By definition of $h_{Q,\Omega}$ (cf. Definition 4.1), we have

$$\Omega \subset \bigcup_{j=1}^{N} B_{h_{Q,\Omega}}(q^j).$$

Let $Q' = \{q^{j_1}, \dots, q^{j_m}\} \subset Q$ be the subset determined by Algorithm 1 (reproduced in Remark C2). Q' satisfies the conclusion of the Vitali covering lemma C1 with $h = h_{Q,\Omega}$; thus,

$$\Omega \subset \bigcup_{k=1}^{m} B_{3h}(q^{j_k}), \tag{C.1}$$

and $B_h(q^{j_k}) \cap B_h(q^{j_\ell}) = \emptyset$ for all $k \neq \ell$. The inclusion (C.1) implies that

$$h_{Q',\Omega} = \sup_{q\in\Omega} \min_{k=1,\dots,m} \left|q - q^{j_k}\right| \leq 3h = 3h_{Q,\Omega}.$$

On the other hand, the fact that $B_h(q^{j_k}) \cap B_h(q^{j_\ell}) = \emptyset$ implies that $\frac{1}{2}\left|q^{j_\ell} - q^{j_k}\right| \geq h = h_{Q,\Omega}$ for all $k \neq \ell$, and hence $\rho_{Q'} \geq h_{Q,\Omega}$. Thus, we have

$$h_{Q',\Omega} \leq 3h_{Q,\Omega} \leq 3\rho_{Q'}.$$

The bound $\rho_{Q'} \leq h_{Q',\Omega}$ always holds (for convex sets); to see this, choose q^{j_k}, q^{j_ℓ} such that $\rho_{Q'} = \frac{1}{2}\left|q^{j_k} - q^{j_\ell}\right|$ realizes the minimal separation distance. Let $\overline{q}$ be the mid-point $\frac{1}{2}\left(q^{j_k} + q^{j_\ell}\right)$, so that $\left|\overline{q} - q^{j_k}\right| = \left|\overline{q} - q^{j_\ell}\right| = \frac{1}{2}\left|q^{j_k} - q^{j_\ell}\right| = \rho_{Q'}$. We note that $\left|\overline{q} - q^{j_r}\right| \geq \rho_{Q'}$ for any $q^{j_r} \in Q'$, since

$$2\rho_{Q'} \leq \left|q^{j_r} - q^{j_k}\right| \leq \left|q^{j_r} - \overline{q}\right| + \left|\overline{q} - q^{j_k}\right| = \left|\overline{q} - q^{j_r}\right| + \rho_{Q'},$$

i.e., $\left|\overline{q} - q^{j_r}\right| \geq \rho_{Q'}$ for $r = 1, \dots, m$. We conclude that

$$\rho_{Q'} \leq \min_{r=1,\dots,m} \left|\overline{q} - q^{j_r}\right| \leq \sup_{q\in Q} \min_{r=1,\dots,m} \left|q - q^{j_r}\right| = h_{Q',\Omega}.$$

$\square$

LEMMA C.4. Let $\Omega = [0, 2\pi]^d \subset \mathbb{R}^d$, let $r \geq 2$ and let $\kappa \geq 1$. There exists $C = C(d, r, \kappa) > 0$ and $\gamma = \gamma(d, r, \kappa) > 0$, such that for all $f^\dagger \in C^r(\Omega)$ and all $Q \subset \Omega$, quasi-uniform with respect to κ, and with fill distance $h_{Q,\Omega}$, the approximation error of the moving least squares interpolation (4.2) with exact

data $z^j = f^\dagger(q^j)$, and parameter $\delta := \gamma h_{z,Q}$, is bounded as follows:

$$\|f^\dagger - f_{z,Q}\|_{L^\infty} \leq C h_{Q,\Omega}^r \|f^\dagger\|_{C^r}.$$

$\Diamond$

Proof. This is immediate from (Wendland, 2004, Corollary 4.8). $\square$

Based on Lemma C4, we can also derive a reconstruction estimate, when the interpolation data $\{q^{j,\dagger}, f^\dagger(q^{j,\dagger})\}_{j=1}^N$ is only known up to small errors in both the position $q^j \approx q^{j,\dagger}$ and the values $z^j \approx f^\dagger(q^{j,\dagger})$, and this is Proposition 4.2 stated in the main body of the text. We now prove this proposition.

Proof (Proposition 4.2). Let $\psi : \mathbb{R}^d \to [0,1]$ be a compactly supported C^∞ function, such that $\psi(0) = 1$, $\|\psi\|_{L^\infty} \leq 1$ and $\psi(q) = 0$ for $|q| \geq 1/2$. Define

$$\widetilde{f}(q) := f^\dagger(q) + \sum_{j=1}^N \left(z^j - f^\dagger(q^j)\right) \psi\left(\frac{q - q^j}{\rho_Q}\right).$$

Note that, by assumption, we have

$$|q^j - q^k| \geq |q^{j,\dagger} - q^{k,\dagger}| - 2\delta \geq 2(\rho_Q - \rho) \geq \rho_Q,$$

for all $j \neq k$. In particular, this implies that the functions

$$q \mapsto \psi\left(\frac{q - q^j}{\rho_Q}\right),$$

have disjoint support for different values of j. Thus, for any $j = 1, \ldots, N$, we have $\widetilde{f}(q^j) = z^j$, and $f_{z,Q}$ is the (exact) moving least squares approximation of $\widetilde{f}$. From Proposition C.4 it follows that

$$\|\widetilde{f} - f_{z,Q}\|_{L^\infty} \leq C \|\widetilde{f}\|_{C^r} h_{Q,\Omega}^r.$$

We now note that

$$|z^j - f^\dagger(q^j)| \leq |z^j - f^\dagger(q^{j,\dagger})| + |f^\dagger(q^{j,\dagger}) - f^\dagger(q^j)| \leq \epsilon + \|f^\dagger\|_{C^r} \rho.$$

On the one hand (due to the disjoint supports of the bump functions), we can now make the estimate

$$\|\widetilde{f} - f^\dagger\|_{L^\infty} \leq \max_{j=1,\ldots,N} |z^j - f^\dagger(q^j)| \leq \epsilon + \|f\|_{C^r} \rho.$$

On the other hand (again due to the disjoint supports of the bump functions), we also have

$$\begin{aligned} \|\widetilde{f}\|_{C^r} &\leq \|f^\dagger\|_{C^r} + \max_{j=1,\ldots,N} |z^j - f^\dagger(q^j)| \rho_Q^{-r} \|\psi\|_{C^r} \\ &\leq \|f^\dagger\|_{C^r} + \frac{\epsilon + \|f^\dagger\|_{C^r} \rho}{\rho_Q^r} \|\psi\|_{C^r}. \end{aligned}$$

These two estimates imply that

$$\begin{aligned}\|f^\dagger - f_{z,Q}\|_{L^\infty} &\leq \|f^\dagger - \widetilde{f}\|_{L^\infty} + \|\widetilde{f} - f_{z,Q}\|_{L^\infty} \\ &\leq \left(\epsilon + \|f^\dagger\|_{C^1}\rho\right) + C\left(\|f^\dagger\|_{C^r} + \left(\epsilon + \|f^\dagger\|_{C^1}\rho\right)\frac{\|\psi\|_{C^r}}{\rho_Q^r}\right)h_{Q,\Omega}^r.\end{aligned}$$

Taking into account that $h_{Q,\Omega}/\rho_Q \leq \kappa$ is bounded, that $\|\psi\|_{C^r}$ is independent of $f^\dagger$, and introducing a new constant $C = C(d, r, \kappa, \|\psi\|_{C^r}) > 0$, we obtain

$$\|f^\dagger - f_{z,Q}\|_{L^\infty} \leq C\left(\|f^\dagger\|_{C^r}h_{Q,\Omega}^r + \epsilon + \|f^\dagger\|_{C^1}\rho\right),$$

as claimed. □

Appendix D. Complexity estimates for HJ-Net approximation

In the last section, we have shown that given a set of initial data $\mathcal{F} \subset C^r_{\text{per}}(\Omega)$, with $r \geq 2$ and $\Omega = [0, 2\pi]^d$, the method of characteristics is applicable for a positive time $T^* > 0$. In the present section, we will combine this result with the proposed HJ-Net framework to derive quantitative error and complexity estimates for the approximation of the solution operator of (HJ). This will require estimates for the approximation of the Hamiltonian flow $\Psi_t \approx \Psi_t^\dagger$ by neural networks, as well as an estimate for the reconstruction error resulting from the pruned moving least squares operator $\mathcal{R}$.

D.1 *Approximation of Hamiltonian flow*

Proof (Proposition 5.4). It follows from (Yarotsky, 2017, Theorem 1) (and a simple scaling argument), that there exists a constant $C = C(r, d) > 0$, such that for any $\epsilon > 0$, there exists a neural network $\Psi_t \approx \Psi_t^\dagger$ satisfying the bound,

$$\sup_{(q_0,p_0,z_0)\in\Omega\times[-M,M]^d\times[-M,M]} \left|\Psi_t(q_0, p_0, z_0) - \Psi_t^\dagger(q_0, p_0, z_0)\right| \leq \epsilon, \tag{D.1}$$

and such that

$$\begin{aligned}\text{depth}(\Psi_t) &\leq C\log\left(\frac{M^r\|\Psi_t^\dagger\|_{C^r}}{\epsilon}\right), \\ \text{size}(\Psi_t) &\leq C\left(\frac{M^r\|\Psi_t^\dagger\|_{C^r}}{\epsilon}\right)^{(2d+1)/r}\log\left(\frac{M^r\|\Psi_t^\dagger\|_{C^r}}{\epsilon}\right),\end{aligned}$$

where $\|\Psi_t^\dagger\|_{C^r} = \|\Psi_t^\dagger\|_{C^r(\Omega\times[-M,M]^d\times[-M,M])}$ denotes the C^r norm on the relevant domain. To prove the claim, we note that for any trajectory (q_t, p_t) satisfying the Hamiltonian ODE system

$$\dot{q}_t = \nabla_p H(q_t, p_t), \quad \dot{p}_t = -\nabla_q H(q_t, p_t), \tag{D.2}$$

with initial data $(q_0, p_0) \in \Omega \times [-M, M]^d$, we have by assumption 3.1:

$$\frac{\mathrm{d}}{\mathrm{d}t}(1 + |p_t|^2) = p_t \cdot \dot{p}_t = -2p_t \cdot \nabla_q H(q_t, p_t) \leq 2L_H(1 + |p_t|^2).$$

Integration of this inequality implies $|p_t|^2 \leq (1+|p_0|^2)\exp(2L_H t) \leq (1+dM^2)\exp(2L_H t)$. Taking also into account that $M \geq 1$, this implies that $p_t \in [-\beta M, \beta M]^d$, where $\beta = (1+\sqrt{d})\exp(L_H t)$ depends only on d, L_H and t. Since p_t remains uniformly bounded and since $q \mapsto H(q,p)$ is 2π-periodic, it follows that for any $(q_0,p_0) \in \Omega \times [-M,M]^d$ the Hamiltonian trajectory (q_t,p_t) starting at (q_0,p_0) stays in $\Omega \times [-\beta M, \beta M]$.

Recall that $\Psi_t^\dagger(q_0,p_0,z_0) = (q_t,p_t,z_t)$ is the flow map of the Hamiltonian ODE (D.2) combined with the action integral

$$z_t = z_0 + \int_0^t \Big[p_t \cdot \nabla_p H(q_t,p_t) - H(q_t,p_t)\Big]\,\mathrm{d}\tau. \tag{D.3}$$

Since the Hamiltonian trajectories starting at $(q_0,p_0) \in \Omega \times [-M,M]^d$ are confined to $\Omega \times [-\beta M, \beta M]^d$ and since the right-hand side of (D.2) and (D.3) involve only first-order derivatives of H, it follows from basic ODE theory that there exists $C = C(\|H\|_{C^{r+1}(\Omega\times[-\beta M,\beta M]^d)}, M, t, r) > 0$, such that the C^r-norm of the flow can be bounded by

$$\|\Psi_t\|_{C^r(\Omega\times[-M,M]^d\times[-M,M])} \leq C(\|H\|_{C^{r+1}(\Omega\times[-\beta M,\beta M]^d)}, M, t, r).$$

In particular, we finally conclude that there exist constants $\beta = \beta(L_H,d,t)$ and $C = C(\|H\|_{C^{r+1}(\Omega\times[-\beta M,\beta M]^d)}, M, t, r) > 0$, such that for any $\epsilon > 0$, there exists a neural network $\Psi_t \approx \Psi_t^\dagger$ satisfying the bound (D.1), and such that

$$\mathrm{depth}(\Psi_t) \leq C\log\left(\epsilon^{-1}\right), \quad \mathrm{size}(\Psi_t) \leq C\epsilon^{-(2d+1)/r}\log\left(\epsilon^{-1}\right).$$

□

D.2 *Reconstruction error*

Proof. (Proposition 5.6) Let $f^\dagger \in C^r_{\mathrm{per}}(\Omega)$ be given with $r \geq 2$, and let $\{q^j, z^j\}_{j=1}^N$ be approximate interpolation data, with

$$|q^j - q^{j,\dagger}| \leq h^r_{Q,\Omega}, \quad |z^j - f^\dagger(q^{j,\dagger})| \leq h^r_{Q,\Omega}.$$

The assertion of this proposition is restricted to $Q^\dagger = \{q^{j,\dagger}\}_{j=1}^N$ satisfying $h_{Q^\dagger,\Omega} \leq h_0$ for a constant h_0 (to be determined below). We may wlog assume that $h_0 \leq 1/16$ in the following. Denote $Q := \{q^j\}_{j=1}^N$. We recall that the first step in the reconstruction Algorithm 2 consists in an application of the pruning Algorithm 1 to determine pruned interpolation points $Q' = \{q^{j_1},\dots,q^{j_m}\} \subset Q$, such that (cf. Lemma C3)

$$\rho_{Q'} \leq h_{Q',\Omega} \leq 3\rho_{Q'}, \quad h_{Q',\Omega} \leq 3h_{Q,\Omega}, \quad h_{Q,\Omega} \leq \rho_{Q'}.$$

Step 1: Write $Q'^{,\dagger} = \{q^{j_1,\dagger},\dots,q^{j_m,\dagger}\}$. Our first goal is to show that $Q'^\dagger$ is **quasi-uniform**: To this end, we note that by definition of the separation distance and the upper bound on the distance of $q^{j,\dagger}$ and q^j:

$$\rho_{Q'^,\dagger} \geq \rho_{Q'} - 2\max_k |q^{j_k,\dagger} - q^{j_k}| \geq \rho_{Q'} - 2h^r_{Q^\dagger,\Omega}.$$

By the definition of the fill distance, and the assumption that $h_{Q^\dagger,\Omega} \leq h_0 \leq 1/2$ and $r \geq 2$, we also have

$$h_{Q^\dagger,\Omega} \leq h_{Q,\Omega} + \sup_j |q^j - q^{j,\dagger}| \leq h_{Q,\Omega} + h^r_{Q^\dagger,\Omega} \leq h_{Q,\Omega} + \frac{1}{2}h_{Q^\dagger,\Omega},$$

implying the upper bound $h_{Q^\dagger,\Omega} \leq 2h_{Q,\Omega} \leq 2\rho_{Q'}$. Similarly, we can show that $h_{Q'^\dagger,\Omega} \leq 2h_{Q',\Omega}$. Substitution in the lower bound on $\rho_{Q'^\dagger}$ above and using that $h_0 \leq 1/16$, yields

$$\rho_{Q'^\dagger} \geq \rho_{Q'} - 2\rho_{\widetilde{Q}'}(2h_{Q,\Omega})^{r-1} \geq \rho_{Q'}(1 - 2(2h_0)^{r-1}) \geq \frac{3}{4}\rho_{Q'}.$$

Thus, we conclude that

$$h_{Q'^\dagger,\Omega} \leq 2h_{Q',\Omega} \leq 6\rho_{Q'} \leq 12\rho_{Q'^\dagger}.$$

Since we always have $\rho_{Q'^\dagger} \leq h_{Q'^\dagger,\Omega}$, we conclude that $Q'^\dagger$ is quasi-uniform with $\kappa = 12$.

Step 2: Next, we intend to apply Proposition 4.2 with $Q'^\dagger$ in place of $Q^\dagger$ and with $\rho = \epsilon = h^r_{Q^\dagger,\Omega}$: To this end, it remains to show that $\rho \leq \frac{1}{2}\rho_{Q'^\dagger}$ is bounded by half the separation distance. To see this, we note that by the above bounds (recall also that $h_0 \leq 1/16$),

$$\rho \equiv h^r_{Q^\dagger,\Omega} \leq h_0^{r-1}\, h_{Q^\dagger,\Omega} \leq \frac{1}{16}h_{Q^\dagger,\Omega} \leq \frac{1}{16}\rho_{Q'} \leq \frac{1}{8}\rho_{Q'^\dagger} < \frac{1}{2}\rho_{Q'^\dagger},$$

showing that $\rho < \frac{1}{2}\rho_{Q'^\dagger}$. We can thus apply Proposition 4.2 to conclude that there exist constants $C, h_0 > 0$ (with $h_0 \leq 1/16$), such that if $h_{Q^\dagger,\Omega} \leq h_0/9$, we have

$$h_{Q'^\dagger,\Omega} \leq 2h_{Q',\Omega} \leq 6h_{Q,\Omega} \leq 9h_{Q^\dagger,\Omega} \leq h_0,$$

and hence

$$\begin{aligned}\|f^\dagger - f_{z,Q}\|_{L^\infty} &\leq C\left(\|f^\dagger\|_{C^r(\Omega)} + \epsilon + \|f^\dagger\|_{C^1(\Omega)}\rho\right)\\ &\leq 2C\left(1 + \|f^\dagger\|_{C^r(\Omega)}\right) h^r_{Q^\dagger,\Omega}.\end{aligned}$$

Replacing $h_0 \to h_0/9$ and $C \to 2C$ now yields the claimed result of Proposition 5.6. □

Proof. (Proposition 5.7). Let $q_0, \widetilde{q}_0 \in \Omega$ and $u_0 \in \mathcal{F}$ be given, and denote by (q_τ, p_τ) the solution of the Hamiltonian ODE, with initial value $(q_0, p_0) = (q_0, \nabla_q u_0(q_0))$. Define $(\widetilde{q}_\tau, \widetilde{p}_\tau)$ similarly.

By compactness of $\mathcal{F} \subset C^r_{\text{per}}(\Omega) \subset C^2_{\text{per}}(\Omega)$, there exists a constant $M > 0$, such that $|p_0|, |\widetilde{p}_0| \leq \|u_0\|_{C^2(\Omega)} \leq M$. By continuity of the flow map, there exists $\overline{M} > 0$, such that

$$|p_\tau|, |\widetilde{p}_\tau| \leq \overline{M}, \quad \forall\, \tau \in [0, t].$$

Then, we have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}|(q_\tau, p_\tau) - (\widetilde{q}_\tau, \widetilde{p}_\tau)| &\leq \left|\nabla H(q_\tau, p_\tau) - \nabla H(\widetilde{q}_\tau, \widetilde{p}_\tau)\right|\\ &\leq \left(\sup_{q\in\Omega, |p|\leq\overline{M}} \|D^2 H(q,p)\|\right)|(q_\tau, p_\tau) - (\widetilde{q}_\tau, \widetilde{p}_\tau)|,\end{aligned}$$

where D^2H denotes the Hessian of H and $\|D^2H\|$ is the matrix norm induced by the Euclidean vector norm. Further denoting

$$\|D^2H\|_\infty := \sup_{q\in\Omega, |p|\leq\overline{M}} \|D^2H(q,p)\|,$$

then by Gronwall's inequality, it follows that

$$|(q_t, p_t) - (\widetilde{q}_t, \widetilde{p}_t)| \leq e^{\|D^2 H\|_\infty t} |(q_0, p_0) - (\widetilde{q}_0, \widetilde{p}_0)|.$$

Furthermore, since $p_0 = \nabla_q u_0(q_0)$, and $\widetilde{p}_0 = \nabla_q u_0(\widetilde{q}_0)$, we have $|p_0 - \widetilde{p}_0| \leq \|u_0\|_{C^2} |q_0 - \widetilde{q}_0| \leq M|q_0 - \widetilde{q}_0|$, which implies that

$$\begin{aligned} |(q_t, p_t) - (\widetilde{q}_t, \widetilde{p}_t)| &\leq e^{\|D^2 H\|_\infty t} |(q_0, p_0) - (\widetilde{q}_0, \widetilde{p}_0)| \\ &\leq (1 + M) e^{\|D^2 H\|_\infty t} |q_0 - \widetilde{q}_0|. \end{aligned}$$

Therefore, $\Phi^\dagger_{t,u_0}(q_0) = q_t$, $\Phi^\dagger_{t,u_0}(\widetilde{q}_0) = \widetilde{q}_t$ satisfy the estimate

$$\left| \Phi^\dagger_{t,u_0}(q_0) - \Phi^\dagger_{t,u_0}(\widetilde{q}_0) \right| \leq C|q_0 - \widetilde{q}_0|,$$

with constant

$$C = (1 + M) \exp\left(t \sup_{q \in \Omega, |p| \leq \overline{M}} \|D^2 H(q, p)\| \right),$$

which depends only on t, H and on $\mathcal{F}$ (via M and $\overline{M}$), but is *independent* of the particular choice of u_0. Thus, if

$$\Omega \subset \bigcup_{j=1}^N B_h(q^j),$$

then

$$\Omega \subset \bigcup_{j=1}^N B_{Ch}(\Phi^\dagger_{t,u_0}(q^j)),$$

for any $u_0 \in \mathcal{F}$. This implies the claim. □