Multiple Neural Operators Achieve Near-Optimal Rates for Multi-Task Learning

Adrien Weihs[1] and Hayden Schaeffer[1]

[1]Department of Mathematics,
University of California Los Angeles,
Los Angeles, CA 90095, USA.

Abstract

We study the approximation and statistical complexity of learning collections of operators in a shared multi-task setting, with a focus on the Multiple Neural Operators (MNO) architecture. For broad classes of Lipschitz multiple operator maps, we derive near-optimal upper bounds for approximation and statistical generalization. On the lower-bound side, we establish a curse of parametric complexity and prove corresponding minimax rates. Together, these results show that shared representations across tasks do not increase the overall cost: multi-task operator learning follows the same scaling laws as single operator learning. We also compare MNO with a multi-task extension of DeepONet based on concatenated task inputs and show that, from a worst-case approximation-complexity perspective, both architectures satisfy essentially the same asymptotic rates.

Keywords and phrases. Deep Neural Networks, Approximation Theory, Neural Scaling Laws, Multi-task Learning, Multi-Operator Learning, Operator Learning.
Mathematics Subject Classification. 41A99, 68T07

1 Introduction

We study the problem of learning collections of operators in a shared multi-task setting, referred to as *multi-operator learning* or *multi-task operator learning* [43,44,46,47]. The goal is to approximate multiple operator maps, that is, maps of the form

$$G : W \longrightarrow \{G[\alpha] : U \to V\}_{\alpha \in W},$$

within a single neural network model. Here, W, U, V are function spaces, $\alpha \in W$ identifies the operator or task, and for each fixed α, the corresponding map $G[\alpha]$ sends an input function $u \in U$ to an output in V. This viewpoint extends the standard operator-learning problem [23, 36], in which one seeks to approximate a single operator between function spaces. It also arises naturally in a wide variety of applications, including parameterized kernel operators, solution operators of parameterized PDEs, and task-conditioned operator families or foundation models; see [46, Section 2] for a broader discussion. Learning such maps is particularly challenging for several reasons. First, multiple operator/multi-task learning is "effectively higher-dimensional" than standard operator learning. Second, the dependence on the operator descriptor α, the input function u, and the evaluation variable x may all be highly nonlinear. Third, these variables play different roles in the problem, and this structure should be taken into account in the design of the learning mechanism.

Given their expressivity and flexibility, neural networks are a natural approximation class in this setting. However, the associated architectural search problem remains a central challenge. Specifically, one seeks network structures that are empirically effective while also admitting theoretical guarantees. A natural way to quantify such guarantees is through their scaling laws. These describe how a target error metric, for instance approximation accuracy or generalization error, depends on a notion of complexity, such as architectural size, sparsity, or the number of training samples. In this sense, scaling laws relate achievable performance to the resources available for learning.

A recent architecture addressing both empirical performance and theoretical guarantees is the Multiple Neural Operators (MNO) architecture [46, 47]. Concretely, for Lipschitz multiple operator maps G, it was shown in [47] that for every target accuracy $\varepsilon > 0$, there exists a MNO with explicit ε-dependent bounds on the depth, width, sparsity, and parameter magnitude (see Table 3) satisfying $\|\mathrm{MNO} - G\| \leq \varepsilon$. In particular, these constructive bounds imply an estimate of the form

$$N_{\#}(\mathrm{MNO}) \leq \varepsilon^{-\varepsilon^{-\varepsilon^{-1/d}}},$$

where $N_{\#}$ denotes the total number of nonzero parameters and $d > 0$ depends on the underlying function classes. These approximation-theoretic bounds were then used in [46] to derive statistical learning rates. In particular, one obtains a generalization bound scaling as

$$\left(\frac{\log\log n_\alpha}{\log\log\log n_\alpha}\right)^{-2/d},$$

where n_α denotes the number of sampled operators $G[\alpha]$ used during training. To the best of our knowledge, these rates were the first of their kind for multiple operator learning. However, they also exhibit an additional exponential blow-up relative to the corresponding rates in standard operator learning (see Table 1), raising the possibility that multiple operator learning may be intrinsically more complex and might suffer from a specific *curse of parametric complexity*.

In this paper, we demonstrate that the complexity of multi-task operator learning matches the qualitative exponential rates one expects for single-task operator learning. Specifically, our first main result is a substantial improvement of the previously known approximation and generalization scaling laws for multi-task learning using MNO. In particular, we show that the additional constructive blow-up is not intrinsic: one can derive complexity upper bounds for MNO that match the scale of the corresponding operator-learning rates. This immediately yields a stronger bound in the statistical learning rates. Our second main contribution is a lower-bound theory for multiple operator learning. We prove lower complexity bounds for broad classes of Lipschitz and differentiable multiple operator maps, showing that the upper bounds obtained in this work are close to sharp and that some form of parametric complexity barrier is indeed unavoidable. Lastly, we show that the same minimax scaling laws also apply to alternative architectural approaches to multiple operator learning. In this way, the paper clarifies which aspects of the previously observed complexity growth are artifacts of the constructive analysis and which reflect intrinsic barriers of the target class.

1.1 Main Contributions

Our main contributions are summarized as follows.

1. **Near-optimal constructive approximation rates for MNO** We derive near-optimal approximation-theoretic upper bounds for MNO on classes of Lipschitz multiple operator maps in Theorem 3.1. The rates are obtained through a refined error analysis of the constructive approximation scheme. In particular, for every target accuracy $\varepsilon > 0$, we construct a MNO approximator with uniform error at most ε and with explicit ε-dependent bounds on the depth, width, sparsity, and parameter magnitude of its subnetworks. The precise scalings are given in Table 3. The resulting total approximation complexity satisfies

$$N_{\#} \lesssim \exp\left(d\log(\varepsilon^{-1})\,\varepsilon^{-\max\{d_W,d_U\}}\right),$$

where d_W, d_U denote the dimension of the domain of functions in W, U respectively and $d > 0$ depends on U and W. Inverting the relationship, we obtain the following scaling of error ε as a function of $N_{\#}$:

$$\varepsilon \lesssim \left(\frac{\log N_{\#}}{\log\log N_{\#}}\right)^{-1/\max\{d_W,d_U\}}.$$

These rates are of iterated exponential-type, qualitatively matching the known rates for operator learning [15, 24–26, 31, 37, 38, 42].

2. **Refined statistical learning rates for MNO** We show that the refined approximation-theoretic bounds propagate directly to the statistical setting. More precisely, by combining the refined approximation construction with the generalization framework of [46], we obtain stronger learning rates for MNO on Lipschitz multiple operator maps in Theorem 3.4. In particular, the resulting generalization bound scales as

$$\left(\frac{\log n_\alpha}{\log\log n_\alpha}\right)^{-2/\max\{d_W,d_U\}},$$

thus improving the previously known rates and reducing them to the corresponding operator-learning scale.

3. **Lower complexity bounds for multiple operator learning** We extend the lower-complexity framework of [26] to the multiple operator setting. We introduce the notion of multiple operator maps of neural network type adapted to separable architectures such as MNO. As a consequence, in Theorem 3.8 and Lemma 3.10, we obtain lower complexity bounds for broad classes of Lipschitz and differentiable multiple operator maps, proving the same curse of parametric complexity as in operator learning for the multiple operator case. Specifically, we show that there exists a multiple operator map G so that for any suitable neural network NN satisfying $\|\,\mathrm{NN} - G\| \le \varepsilon$, we have

$$N_{\#}(\mathrm{NN}) \ge \exp\left(c\,\varepsilon^{-1/\eta}\right)$$

where η depends on G, W and U.

4. **Minimax approximation-complexity rates for multiple operator learning** Adapting the lower bounds to the precise setting of the improved constructive upper bounds, in Theorem 3.19, we derive minimax approximation-complexity rates for MNO on the Lipschitz class $\mathcal{H}$. In particular, for the worst-case approximation complexity $\mathfrak{C}(\varepsilon;\mathcal{H})$, we obtain bounds of the form

$$\exp\left(c\,\varepsilon^{-1/\eta}\right) \lesssim \mathfrak{C}(\varepsilon;\mathcal{H}) \lesssim \exp\left(d\log(\varepsilon^{-1})\,\varepsilon^{-\max\{d_W,d_U\}}\right),$$

showing that the upper bounds obtained in this work are close to sharp at the level of the overall exponential complexity regime.

5. **Comparison with a concatenated DeepONet baseline.** We analyze an alternative multiple operator architecture obtained by concatenating the operator descriptor and the input function, and prove corresponding upper and lower complexity bounds for this model in Theorem 3.29. In particular, we show that this concatenated DeepONet-type baseline obeys essentially the same minimax approximation-complexity scaling as MNO. Thus, from the viewpoint of worst-case approximation complexity, the present theory does not separate the two architectures, even though MNO exhibits markedly stronger empirical performance in previously reported experiments [47].

1.2 Related Works

Multi-task and multiple operator learning There are at least two broad motivations for learning operator families rather than isolated operators. In some applications, the problem itself is naturally described by a collection of related operators, for instance through variations in physical parameters, geometry, boundary conditions, or governing equations. In other settings, learning several operators jointly is primarily a modeling strategy: by sharing structure across tasks, one may improve data efficiency, robustness, and transfer. These perspectives have motivated a growing recent literature on multiple operator learning and closely related frameworks; see, for example, [2, 4, 14, 18, 33, 34, 39, 43, 45–52, 55].

At a coarse level, two modeling paradigms are common. One option is to train separate operator learners independently, one for each task or operator instance. Another is to regard the target as a parameterized family $\{G[\alpha]\}$, where a discrete or continuous descriptor α specifies the operator identity. The former viewpoint avoids introducing an explicit operator descriptor, but is correspondingly limited in its ability to exploit shared structure and to extrapolate to unseen operators. The latter instead augments operator learning with an explicit encoding of operator information [32,34,40,43,46,47,49], such as a task label, symbolic expression, governing

equation, or textual description. This conditioning mechanism has become central in recent work on PDE foundation models and task-adaptive operator learning, where it often improves transfer and enables zero-shot or out-of-distribution generalization without retraining.

Theoretical analyses of approximation and statistical generalization. A basic theoretical question in operator learning is expressivity: can a given architecture approximate large classes of operators, and under what assumptions? Early foundational work on neural network approximation of maps between spaces of scalar-valued functions was developed in [7, 8]. Since then, universal approximation and related expressivity results have been established for a range of operator-learning architectures, including DeepONet [25, 31], the Fourier Neural Operator [20], and PCA-Net [3]. Further developments concerning operator approximation, the effect of discretization, and architectural refinements may be found in [5, 6, 16, 17, 21, 53, 56].

A second line of work seeks to also quantify how approximation and learning performance scale with available resources. In this direction, scaling laws relate error to quantities such as model complexity, data size, and computational budget, and thereby provide a theoretical basis for performance prediction and generalization analysis [19]. They offer a principled way to compare the efficiency of different architectures by quantifying the amount of complexity required to attain a prescribed error level. Also, when constructive upper bounds are compared with lower bounds, they help separate inefficiencies of a particular architecture or proof strategy from barriers that are intrinsic to the target class itself. This distinction is especially important in multiple operator learning, where complexity may arise both from the structure of the multiple operator class and from the chosen architecture.

In the standard operator-learning setting, both empirical and theoretical analyses of scaling behavior have received significant attention. On the empirical side, [10] studies cost–accuracy trade-offs across neural operator architectures and highlights the role of both network size and sampling budget. On the theoretical side, approximation scaling laws and complexity estimates for deep ReLU networks, DeepONet, and related operator-learning architectures have been developed in [12, 15, 24–26, 31, 37, 38, 42]. The precise form of these scaling laws depends on several ingredients, including the target class of operators, the regularity and geometry of the input and output spaces, and the model class used for approximation. Existing theoretical results in operator learning can be organized, very broadly, into the categories shown in Table 1.

Setting	**Neural architecture**	**Complexity bound**	**Representative references**
Lipschitz operators (upper bounds)	DeepONet	$\exp\left(c\log(\varepsilon^{-1})\varepsilon^{-d}\right)$	[31]
Lipschitz/Differentiable operators (lower bounds)	DeepONet, FNO, PCANet, etc.	$\exp\left(c\varepsilon^{-d}\right)$	[26]
Lipschitz operators with non-standard architectures	DeepONet	ε^{-d}	[42]
Holomorphic operators	DeepONet	substantially improved rates	[15, 38]
PDE operators (problem-specific analyses)	DeepONet, FNO, PCANet, etc.	problem-dependent	[24–26, 37]

Table 1: Schematic overview of representative scaling-law regimes in operator learning. The precise exponents, assumptions, and architecture classes vary across settings.

Statistical guarantees have also become an active topic of study. For DeepONet-type models, [31] derives generalization bounds of the form

$$\left(\frac{\log n_u}{\log\log n_u}\right)^{-1/d},$$

for some $d > 0$, with n_u denoting the number of sampled input functions available during training. Closely related statistical rates for DeepONet and related models are obtained in [27, 30, 31]. Complementary sample-complexity analyses for operator learning are developed in [1, 13, 22]. By contrast, the corresponding theory in multiple operator learning remains much less developed. Empirical evidence on multi-task and operator-family

learning can be found in [18, 44]. For the Multiple Neural Operators architecture, universal approximation results and the first explicit approximation scaling laws were established in [47], and the corresponding statistical generalization rates were derived in [46]. The present paper builds on this line by substantially improving these complexity estimates and by complementing them with lower bounds and minimax approximation-complexity rates. The theoretical framework transfers to other neural operator frameworks that are built from [8].

Organization of the paper. The remainder of the paper is organized as follows. In Section 2, we introduce the assumptions and the mathematical framework used throughout the paper. Section 3 presents the main results of the paper. Proofs of all results are collected in the Appendix.

2 Background

In this section, we review the mathematical framework underlying our analysis. We begin by introducing the general notation and collecting the assumptions on the function spaces, target multiple operator maps, and product norms used throughout the paper. We then summarize the existing approximation and generalization scaling laws for MNO, which serve as a benchmark for our improved upper bounds. Finally, we recall the lower-complexity framework from operator learning on which our lower-bound analysis is based.

2.1 General Notation

Throughout the paper, we take $\mathbb{N} = \{1, 2, 3, \dots\}$. For operators $T : U \to V$, we define the norm

$$\|T\|_{\mathrm{op}} := \sup_{u \in U} \|T(u)\|_V.$$

We write Ω_A for the domain of functions in the function set A. We denote the ball centered at x with radius δ with respect to the norm $\|\cdot\|$ as $\mathcal{B}_{\delta,\|\cdot\|}(x)$. When the norm is omitted, the norm is understood to be the ℓ^2/Euclidean norm. When considering balls of functions mapping from Ω_A into $\mathbb{R}$, we write $\mathcal{B}_{\delta,\|\cdot\|,\Omega_A}(x)$. We denote by ev the evaluation operator; for instance, $\mathrm{ev}_x(f) = f(x)$ for a function f and a point x in its domain. For a neural network Φ, we write $N_{\#}(\Phi)$ for the total number of nonzero parameters of Φ, that is, the total number of nonzero entries in all weight matrices and bias vectors. We use $\lesssim$ in scaling discussions to indicate the dominant asymptotic order, possibly suppressing multiplicative constants and lower-order terms. Lastly, we recall the definition of covering numbers.

Definition 2.1 (Covering Number)**.** *Let (X, d) be a metric space and let $\eta > 0$. A finite subset $\mathcal{C} \subset X$ is called a η-cover of X if for every $x \in X$, there exists $c \in \mathcal{C}$ such that*

$$d(x, c) \leq \eta.$$

The covering number of X at scale η with respect to the metric d is defined as

$$\mathcal{N}(\eta, X, d) := \min\{|\mathcal{C}| : \mathcal{C} \subset X \text{ is a } \eta\text{-cover of } X\}.$$

2.2 Assumptions

Assumptions 1. We make the following assumption on our spaces.

S.1 The space $U(d_U, \gamma_U, L_U, \beta_U)$ is a function set such that

(a) any function $u \in U$ is defined on $\Omega_U := [-\gamma_U, \gamma_U]^{d_U}$;

(b) for all functions $u \in U$ and $x, y \in \Omega_U$, we have

$$|u(x) - u(y)| \leq L_U |x - y|;$$

(c) for all functions $u \in U$, we have $\|u\|_{\mathrm{L}^\infty} \leq \beta_U$.

The space defined in Assumption **S.1** yields a general approximation class for approximation theory.

Assumptions 2. We make the following assumptions on our multiple operator map G.

O.1 Assume that $W(d_W, \gamma_W, L_W, \beta_W)$, $U(d_U, \gamma_U, L_U, \beta_U)$ and $V(d_V, \gamma_V, L_V, \beta_V)$ satisfy Assumption **S.1**. For $L_{\mathcal{G}} > 1$, $r_{\mathcal{G}} \geq 1$, a multiple operator map is a function G such that

$$G : \mathcal{B}_{\beta_W, \|\cdot\|_{\mathrm{L}^\infty}, \Omega_W}(0) \mapsto \mathcal{G}$$

where $\mathcal{G} = \Big\{ G[\alpha] \,|\, G[\alpha] : \mathcal{B}_{\beta_U, \|\cdot\|_{\mathrm{L}^\infty}, \Omega_U}(0) \mapsto V \text{ and } \|G[\alpha][u_1] - G[\alpha][u_2]\|_{\mathrm{L}^\infty(\Omega_V)} \leq L_{\mathcal{G}} \|u_1 - u_2\|_{\mathrm{L}^{r_{\mathcal{G}}}(\Omega_U)} \Big\}$

O.2 For $r_G \geq 1$, the multiple operator map G satisfies

$$\|G[\alpha_1] - G[\alpha_2]\|_{\mathrm{L}^\infty(\mathcal{B}_{\beta_U, \|\cdot\|_{\mathrm{L}^\infty}, \Omega_U}(0) \times \Omega_V)} \leq L_G \|\alpha_1 - \alpha_2\|_{\mathrm{L}^{r_G}(\Omega_W)}$$

for $\alpha_1, \alpha_2 \in \mathcal{B}_{\beta_W, \|\cdot\|_{\mathrm{L}^\infty}, \Omega_W}(0)$.

Assumptions **O.1** and **O.2** reflect minimal Lipschitz regularity assumptions on multiple operator maps required for our approximation-theoretical results.

Assumptions 3. We make the following assumptions on our product norm.

N.1 For Banach spaces X and Y, we equip $X \times Y$ with a norm $\|\cdot\|_{X \times Y}$ satisfying $\|(x, y)\|_{X \times Y} \geq C_{\text{prod}} \max\{\|x\|_X, \|y\|_Y\}$ for some $C_{\text{prod}} > 0$.

N.2 For Banach spaces X and Y, we equip $X \times Y$ with a norm $\|\cdot\|_{X \times Y}$ satisfying $\|(x, y)\|_{X \times Y} \leq C_{\text{prod}} \max\{\|x\|_X, \|y\|_Y\}$ for some $C_{\text{prod}} > 0$.

N.3 For Banach spaces X and Y, we equip $X \times Y$ with a norm $\|\cdot\|_{X \times Y}$ satisfying $\|(x, 0)\|_{X \times Y} = \|x\|_X$ and $\|(0, y)\|_{X \times Y} = \|y\|_Y$.

The following product Banach space norms satisfy all Assumptions **N.1**, **N.2** and **N.3**: $\|(x, y)\|_{X \times Y} = (\|x\|_X^p + \|y\|_Y^p)^{1/p}$ with $p \geq 1$ or $\|(x, y)\|_{X \times Y} = \max\{\|x\|_X, \|y\|_Y\}$.

2.3 Complexity Scaling Laws for Multiple Operator Learning

We begin by introducing the class of neural networks used throughout our analysis. This class covers a wide range of architectures, is widely used [28–31, 54], and will serve as the basic building block for all of our constructions.

Definition 2.2 (Feedforward ReLU Network Class)**.** *Let $q : \mathbb{R}^{d_1} \to \mathbb{R}$ be a feedforward ReLU network defined as*

$$q(x) = W_L \cdot \mathrm{ReLU}\left(W_{L-1} \cdots \mathrm{ReLU}(W_1 x + b_1) + \cdots + b_{L-1}\right) + b_L, \tag{1}$$

where W_ℓ are weight matrices, b_ℓ are bias vectors, and $\mathrm{ReLU}(a) = \max\{a, 0\}$ is applied element-wise. We define the class of such feedforward networks with ReLU activations:

$$\mathcal{F}_{\mathrm{NN}}(d_1, d_2, L, p, K, \kappa, R) = \left\{ [q_1, q_2, \ldots, q_{d_2}]^\top \in \mathbb{R}^{d_2} \;\middle|\; \begin{array}{l} \text{each } q_k : \mathbb{R}^{d_1} \to \mathbb{R} \text{ has the above form with} \\ L \text{ layers, width bounded by } p, \\ \|q_k\|_{\mathrm{L}^\infty} \leq R, \quad \|W_\ell\|_{\infty,\infty} \leq \kappa, \quad \|b_\ell\|_\infty \leq \kappa, \\ \sum_{\ell=1}^{L} \left(\|W_\ell\|_0 + \|b_\ell\|_0\right) \leq K \end{array} \right\},$$

where

- $\|q\|_{\mathrm{L}^\infty} = \sup_{x \in \Omega} |q(x)|$,
- $\|W_\ell\|_{\infty,\infty} = \max_{i,j} |[W_\ell]_{ij}|$,
- $\|b_\ell\|_\infty = \max_i |[b_\ell]_i|$,

- $\|\cdot\|_0$ *denotes the number of nonzero elements.*

This network class consists of vector-valued functions with input dimension d_1, output dimension d_2, depth L, width at most p, at most K nonzero parameters, all bounded in magnitude by κ, and uniformly bounded output norm by R.

To approximate multiple operator maps, the following MNO architecture was introduced in [47]. This architecture has proven effective in practice while remaining mathematically tractable.

Definition 2.3 (MNO Architecture)**.** *Let $P, H \in \mathbb{N}$. We define a* MNO *to be a map of the form*

$$\mathrm{MNO}[\alpha][u](x) = \sum_{p=1}^{P} \sum_{k=1}^{H} l_p(M_W(\alpha))\, b_{pk}(M_U(u))\, \tau_{pk}(x),$$

where $\alpha \in W$, $u \in U$, $x \in \Omega_V$, the functions l_p, b_{pk}, and τ_{pk} are chosen from suitable neural-network classes $\mathcal{F}_{\mathrm{NN}}$, and $M_W : W \mapsto \mathbb{R}^m$ and $M_U : U \mapsto \mathbb{R}^q$ are linear maps.

The architecture above fits into the broader fully separable family

$$\sum_{p=1}^{P} \sum_{k=1}^{H} \sum_{\ell=1}^{N} \theta_{pk\ell}\, l_p(M_W(\alpha))\, b_k(M_U(u))\, \tau_\ell(x), \qquad \theta_{pk\ell} \in \mathbb{R}. \tag{2}$$

This more general form is convenient for analysis, and the original MNO architecture is recovered by a direct relabelling of indices or by grouping terms into suitably chosen subnetworks.

A first fundamental question is the expressive capacity of the MNO architecture. In [47], both a universal approximation theorem for continuous multiple operator maps and quantitative scaling laws for Lipschitz multiple operator maps satisfying Assumptions **O.1** and **O.2** were established. In particular, for the latter class, [47, Theorem 3.16] shows that for every target accuracy $\varepsilon > 0$, there exists a network NN of the form (2) such that

$$\sup_{\alpha \in W} \sup_{u \in U} \sup_{x \in \Omega_V} |G[\alpha][u](x) - \mathrm{NN}[\alpha][u](x)| \le \varepsilon,$$

with architectural scalings summarized in Table 3. These scalings imply the following total parameter count and, equivalently, the following approximation rate expressed in terms of the total number of nonzero parameters $N_\#$:

$$N_\# \lesssim \varepsilon^{-\varepsilon^{-\varepsilon^{-d_W}}} \qquad \Longleftrightarrow \qquad \varepsilon \lesssim \left(\frac{\log\log N_\#}{\log\log\log N_\#} \right)^{-1/d_W}.$$

In Section 3.1, we show that a different organization of the approximation argument leads to substantially improved rates.

2.4 Generalization Scaling Laws for Multiple Operator Learning

Beyond approximation capabilities, a second fundamental question in learning concerns generalization, namely, how well an algorithm trained on a given dataset performs on previously unseen samples. We next introduce the learning setup in [46] that is used to train a network of the form (2).

Given a function $f : \mathbb{R} \to \mathbb{R}$ and a constant $a \ge 0$, we denote by

$$\mathrm{Clip}_a(f) := \min\{\max(f, -a), a\}$$

the clipping operation onto the interval $[-a, a]$. This map can be realized using only affine transformations and ReLU activations. In particular, it belongs to the class $\mathcal{F}_{\mathrm{NN}}(1, 1, 2, 1, 6, 2a, a)$.

For the study of generalization, it is convenient to work with a clipped version of the network class (2). This does not impose a significant restriction in our setting: indeed, the approximation results recalled above produce approximating networks with uniformly bounded outputs, so clipping at a sufficiently large level does not alter the constructions. Moreover, in practical implementations, network outputs are typically bounded as well, either explicitly or through the scale of the training data and targets.

Definition 2.4 (Clipped Multiple Operator Network Class)**.** *Let $\mathcal{F}_i$ for $1 \leq i \leq 3$ be network classes defined in Definition 2.2. For $a, I > 0$, $P, H, N \in \mathbb{N}$ and fixed sampling points $\{y_s\}_{s=1}^{n_{c_W}} \subset \Omega_W$ and $\{c_s\}_{s=1}^{n_{c_U}}$, we define $\mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P, H, N)$, the set of a-clipped multiple operator networks, as*

$$\begin{aligned}&\mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P, H, N)\\ &= \left\{ \mathrm{Clip}_a \left(\sum_{p=1}^{P} \sum_{k=1}^{H} \sum_{\ell=1}^{N} \theta_{pk\ell} l_p(\boldsymbol{\alpha}) b_k(\mathbf{u}) \tau_\ell(x) \right) \mid \theta_{pk\ell} \in [-I, I],\, \tau_\ell \in \mathcal{F}_1,\, b_k \in \mathcal{F}_2,\, l_p \in \mathcal{F}_3 \right\}\end{aligned}$$

where $\boldsymbol{\alpha} = (\alpha(y_1), \alpha(y_2), ..., \alpha(y_{n_{c_W}}))^\top$ and $\mathbf{u} = (u(c_1), u(c_2), ..., u(c_{n_{c_U}}))^\top$.

The following definition specifies the probabilistic framework used to model the training data. In the multiple operator setting, observations are generated through a hierarchical sampling procedure: one first draws parameter instances α, then draws input functions u, and finally samples evaluation points x where noisy measurements of the output $G[\alpha][u]$ are taken.

Definition 2.5 (Training Set)**.** *Let $G : W \mapsto \{G[\alpha] : U \to V\}$ be a multiple operator map. Let μ_α be a probability measure on W, μ_u a probability measure on U, and μ_x a probability measure on Ω_V. Given fixed sampling points $\{y_s\}_{s=1}^{n_{c_W}} \subset \Omega_W$ and $\{c_s\}_{s=1}^{n_{c_U}} \subset \Omega_U$, we define the training set:*

$$S_{G,\{y_s\},\{c_s\}} = \left\{ \boldsymbol{\alpha}_\ell, \left\{ \boldsymbol{u}_{\ell i}, \{(x_{\ell ij}, w_{\ell ij})\}_{j=1}^{n_x} \right\}_{i=1}^{n_u} \right\}_{\ell=1}^{n_\alpha}$$

where

- $\alpha_\ell \overset{\text{iid}}{\sim} \mu_\alpha$ *and* $\boldsymbol{\alpha}_\ell = \big(\alpha_\ell(y_1), \ldots, \alpha_\ell(y_{n_{c_W}})\big) \in \mathbb{R}^{n_{c_W}}$*;*
- $u_{\ell i} \overset{\text{iid}}{\sim} \mu_u$ *and* $\boldsymbol{u}_{\ell i} = \big(u_{\ell i}(c_1), \ldots, u_{\ell i}(c_{n_{c_U}})\big) \in \mathbb{R}^{n_{c_U}}$*;*
- $x_{\ell ij} \overset{\text{iid}}{\sim} \mu_x$ *drawn from* Ω_V*;*
- $w_{\ell ij} = G[\alpha_\ell][u_{\ell i}](x_{\ell ij}) + \zeta_{\ell ij}$*, where $\zeta_{\ell ij}$ are i.i.d. sub-Gaussian noise variables with mean 0 and variance proxy σ^2.*

For a given training set $S_{G,\{y_s\},\{c_s\}}$, we introduce the corresponding learned operator below. More precisely, this is a neural network selected from the class $\mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P, H, N)$ by minimizing the empirical L^2-loss over the observed data.

Definition 2.6 (Trained Operator)**.** *Let $\mathcal{F}_i$ for $1 \leq i \leq 3$ be network classes defined in Definition 2.2. Let $G : W \mapsto \{G[\alpha] : U \to V\}$ be a multiple operator map. Let μ_α be a probability measure on W, μ_u a probability measure on U, and μ_x a probability measure on Ω_V. Given fixed sampling points $\{y_s\}_{s=1}^{n_{c_W}} \subset \Omega_W$ and $\{c_s\}_{s=1}^{n_{c_U}} \subset \Omega_U$, let $S_{G,\{y_s\},\{c_s\}}$ be the training set defined in Definition 2.5. For $a, I > 0$, the trained a-clipped operator $G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}$ is defined as*

$$G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S} = \underset{\mathrm{NN} \in \mathrm{Cl}_a(I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,\{y_s\},\{c_s\},P,H,N)}{\operatorname{argmin}} \frac{1}{n_\alpha n_u n_x} \sum_{\ell=1}^{n_\alpha} \sum_{i=1}^{n_u} \sum_{j=1}^{n_x} \left(\mathrm{NN}[\boldsymbol{\alpha}_\ell][\boldsymbol{u}_{\ell i}](x_{\ell ij}) - w_{\ell ij}\right)^2.$$

Next, we define the expected generalization error associated with the learned operator. This quantity evaluates the performance of $G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}$ after averaging both over the randomness of the training set S and over previously unseen test samples (α, u, x). In this way, it captures how well a model trained on one realization of the data transfers to new observations.

Definition 2.7 (Expected Generalization Error)**.** *Let $\mathcal{F}_i$ for $1 \leq i \leq 3$ be network classes defined in Definition 2.2. Let $G : W \mapsto \{G[\alpha] : U \to V\}$ be a map. Let μ_α be a probability measure on W, μ_u a probability measure on U, and μ_x a probability measure on Ω_V. Let $\{y_s\}_{s=1}^{n_{c_W}} \subset \Omega_W$ and $\{c_s\}_{s=1}^{n_{c_U}} \subset \Omega_U$ be fixed sampling points. We define the expected generalization error as*

$$\mathbb{E}_{S_{G,\{y_s\},\{c_s\}}} \underbrace{\mathbb{E}_{\alpha\sim\mu_\alpha} \mathbb{E}_{u\sim\mu_u} \mathbb{E}_{\{x_j\}_{j=1}^{n_x}\sim\mu_x^{\otimes n_x}}}_{\text{test sampling}} \underbrace{\left[\frac{1}{n_x} \sum_{j=1}^{n_x} \left(G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}[\boldsymbol{\alpha}][\mathbf{u}](x_j) - G[\alpha][u](x_j)\right)^2 \right]}_{\text{empirical approximation of the squared } \mathrm{L}^2(\mu_x) \text{ error}},$$

where $\mathbb{E}_{S_{G,\{y_s\},\{c_s\}}}$ denotes the expectation over the full training dataset $S_{G,\{y_s\},\{c_s\}}$.

We now state the main scaling law for the expected generalization error [46, Theorem 3.5 and Corollary 3.8] for MNOs approximating multiple operator maps satisfying Assumptions **O.1** and **O.2**. To this end, let $\varepsilon > 0$ be a target approximation accuracy and consider the hypothesis class

$$\mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P, H, N),$$

where the architectural parameters of $\mathcal{F}_i$ are chosen according to the ε-dependent scalings provided by the approximation theory (i.e. [47, Theorem 3.16]), so that the class contains an ε-accurate approximation of G. Also, let $\eta > 0$ be the covering radius used to estimate the size of this class through the metric entropy

$$\log \mathcal{N}\big(\eta, \mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P, H, N), \|\cdot\|_{\mathrm{L}^\infty(W\times U\times\Omega_V)}\big).$$

Then, we obtain the following approximation–estimation tradeoff, relating the target accuracy, the architectural complexity, and the sampling budgets (n_α, n_u, n_x):

$$\begin{aligned}
&\mathbb{E}_{S_{G,\{y_s\},\{c_s\}}} \mathbb{E}_{\alpha\sim\mu_\alpha}\, \mathbb{E}_{u\sim\mu_u}\, \mathbb{E}_{\{x_j\}_{j=1}^{n_x}\sim\mu_x^{\otimes n_x}} \left[\frac{1}{n_x}\sum_{j=1}^{n_x} \left(G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}[\boldsymbol{\alpha}][\mathbf{u}](x_j) - G[\alpha][u](x_j)\right)^2\right] \\
&\le 4\varepsilon^2 + \eta(8\sigma + 6) \\
&+ \frac{8\sigma\eta}{\sqrt{n_\alpha n_u n_x}} \sqrt{\log\left(\mathcal{N}\left(\eta, \mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P, H, N), \|\cdot\|_{\mathrm{L}^\infty(W\times U\times\Omega_V)}\right)\right) + \log(2)} \\
&+ \frac{16\sigma^2}{n_\alpha n_u n_x}\left(\log\left(\mathcal{N}\left(\eta, \mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P, H, N), \|\cdot\|_{\mathrm{L}^\infty(W\times U\times\Omega_V)}\right)\right) + \log(2)\right) \\
&+ \frac{112\beta_V^2}{3n_\alpha} \log\left(\mathcal{N}\left(\eta/(4\beta_V), \mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P, H, N), \|\cdot\|_{\mathrm{L}^\infty(W\times U\times\Omega_V)}\right)\right)
\end{aligned}$$

where $\boldsymbol{\alpha} = (\alpha(y_1), \alpha(y_2), ..., \alpha(y_{n_{c_W}}))^\top$ and $\mathbf{u} = (u(c_1), u(c_2), ..., u(c_{n_{c_U}}))^\top$. Estimating the metric entropy as a function of ε, picking

$$\varepsilon \asymp \left(\frac{\log\log n_\alpha}{\log\log\log n_\alpha}\right)^{-\frac{1}{d_W}} \qquad \text{and} \qquad \eta = 4\beta_V n_\alpha^{-1},$$

the above reduces to the learning rate:

$$\begin{aligned}
&\mathbb{E}_{S_{G,\{y_s\},\{c_s\}}} \mathbb{E}_{\alpha\sim\mu_\alpha}\, \mathbb{E}_{u\sim\mu_u}\, \mathbb{E}_{\{x_j\}_{j=1}^{n_x}\sim\mu_x^{\otimes n_x}} \left[\frac{1}{n_x}\sum_{j=1}^{n_x} \left(G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}[\boldsymbol{\alpha}][\mathbf{u}](x_j) - G[\alpha][u](x_j)\right)^2\right] \\
&= \mathcal{O}\left(\left(\frac{\log\log n_\alpha}{\log\log\log n_\alpha}\right)^{-\frac{2}{d_W}}\right).
\end{aligned}$$

In Section 3.1, we show that our improved approximation-theoretic rates lead, in the same way, to a corresponding improvement in the learning rate, reducing it to the rate dictated by operator learning.

2.5 Lower Complexity Bounds in Operator Learning

We now recall the operator-learning lower-complexity framework from [26], which will serve as the starting point for our lower-bound analysis in the multiple operator setting.

We begin by introducing the notion of an infinite-dimensional cube [9, 11, 26]. These cubes provide a way to describe intrinsic geometric features of compact sets in infinite-dimensional spaces, and thereby furnish a quantitative measure of approximation-theoretic complexity. They also arise naturally in many function classes of interest: [26, Lemma 2.7] establishes their existence for suitable compact classes of differentiable functions, while Lemma 3.13 proves the same for the bounded Lipschitz classes considered in this paper.

Definition 2.8 (Infinite-dimensional Cube)**.** *Let X be a Banach space and let $e_1, e_2, \cdots \in X$ be a sequence of linearly independent elements so that $\|e_j\|_X = 1$ for all $j \in \mathbb{N}$. Given constants $A > 0$ and $\eta > 1$, we say that a set $K \subset X$ contains an infinite-dimensional cube $Q_\eta = Q_\eta\,(A; \{e_j\}_{j\in\mathbb{N}})$ if the following two conditions hold:*

1. *K contains every element of the form*

$$u = A\sum_{j=1}^{\infty} j^{-\eta} y_j e_j, \qquad y_j \in [0,1] \quad \textit{for all } j \in \mathbb{N}.$$

2. *The family $\{e_j\}_{j\in\mathbb{N}}$ admits a bounded biorthogonal sequence in the dual space X^*, that is, there exist functionals $e_1^*, e_2^*, \cdots \in X^*$ such that*

$$e_k^*(e_j) = \delta_{jk} \qquad \textit{for all } j,k \in \mathbb{N}, \quad \textit{and} \quad \|e_k^*\|_{X^*} \leq M \qquad \textit{for all } k \in \mathbb{N}$$

for some constant $M > 0$.

A cube as in Definition 2.8 may be viewed as an infinite-dimensional coordinate system inside K, with coefficients $y_j \in [0,1]$ that are damped by a decay rate $j^{-\eta}$. The associated bounded biorthogonal sequence guarantees that these coefficients can be recovered from the embedded element, so that the cube genuinely carries a coordinate structure. The exponent η measures the asymptotic size of this embedded cube, and will directly control the lower-complexity bounds stated below.

The next step is to specify the class of neural approximants to which the lower-bound framework applies. This is captured by the notions of functionals and operators of neural network type, which combine a finite-dimensional linear encoding with a ReLU network. Many operator-learning architectures encountered in practice (e.g. DeepONet [35], PCANet [3]) fall within this framework, in the sense that they can be interpreted as operators of neural network type after a suitable finite-dimensional discretization of the input. We begin by introducing functionals of neural network type.

Definition 2.9 (Functional of Neural Network Type)**.** *We say that $\mathrm{NN} : W \to \mathbb{R}$ is a functional of neural network type if*

$$\mathrm{NN}[\alpha] = \Phi(M_W(\alpha)) \qquad \textit{for all } \alpha \in W \tag{3}$$

where $M_W : W \to \mathbb{R}^m$ is linear, and Φ is a ReLU neural network. We define the complexity of NN as

$$\mathcal{C}(\mathrm{NN}) = \min_{\Phi \textit{ satisfying } (3)} N_{\#}(\Phi).$$

We now extend this notion to operators.

Definition 2.10 (Operator of Neural Network Type)**.** *We say that $\mathrm{NN} : U \to V$ is an operator of neural network type if for all $x \in \Omega_V$,*

$$\mathrm{ev}_x \circ \mathrm{NN}[u] = \Phi_x(M_U(u)) \qquad \textit{for all } u \in U \tag{4}$$

where $M_U : U \to \mathbb{R}^q$ is linear, and Φ_x is a ReLU neural network. We define the complexity of NN as

$$\mathcal{C}(\mathrm{NN}) = \sup_{x\in\Omega_V} \min_{\Phi_x \textit{ satisfying } (4)} N_{\#}(\Phi_x).$$

Equipped with the above geometric and representational notions, [26, Corollary 2.12] establishes a curse of parametric complexity for operator learning. More precisely, for every $r \in \mathbb{N}$, there exists an r-times Fréchet differentiable pathological operator G defined on any compact set containing an infinite-dimensional cube such that, for every operator of neural network type NN_ε satisfying

$$\sup_{u\in K} \|G[u] - \mathrm{NN}_\varepsilon[u]\| \leq \varepsilon,$$

the complexity of NN_ε must obey a lower bound of the form

$$\mathcal{C}(\mathrm{NN}_\varepsilon) \geq \exp\left(c\,\varepsilon^{-q}\right)$$

for suitable constants $c, q > 0$. In Section 3.2, our objectives are twofold: first, we prove an analogous result for multiple operator learning with the architecture (2); second, we deduce minimax approximation-complexity rates for the class of Lipschitz multiple operator maps.

3 Main Results

In this section, we present the main results of the paper (proofs are given in the Appendix). We begin by deriving near-optimal scaling laws for both approximation and generalization. We then establish a lower complexity bound for approximation, which in turn yields minimax complexity rates for approximating Lipschitz multiple operator maps with MNO. Finally, we compare these minimax approximation-complexity rates with those of the simpler concatenated DeepONet architecture, which is directly inspired by single-operator learning.

3.1 Near-Optimal Approximation Rates

The first result of this section provides the near-optimal upper bound underlying our minimax rates in Theorem 3.19. The key point is that the new construction in Theorem 3.1 reduces the upper bound on approximation complexity in multiple operator learning to the scale of single operator learning. As explained in Section 1, in other words, the multiple operator structure no longer causes an additional exponential deterioration in the number of parameters required to achieve a prescribed accuracy $0 < \varepsilon \ll 1$. The stronger bounds come from a different organization of the error analysis. Table 2 schematically compares the nested proof strategy of [47] with the new one. In both constructions, the first stage is identical: one approximates the dependence of the target multiple operator map on the variable α, which produces a representation involving on the order of $\varepsilon^{-\gamma}$ terms. The distinction arises only at the second stage, in how the resulting approximation error is aggregated. More precisely, in the nested argument, the second-stage error accumulates term by term over the first-stage representation. Hence, if the second-stage approximation error for each term is bounded by δ, the total contribution is controlled by

$$\sum_{p=1}^{\varepsilon^{-\gamma}} \delta \asymp \varepsilon^{-\gamma}\delta.$$

To ensure that this remains of order ε, one must therefore choose

$$\varepsilon^{-\gamma}\delta \asymp \varepsilon, \qquad \text{that is,} \qquad \delta \asymp \varepsilon^{\gamma+1}.$$

Thus the second-stage approximation must be carried out at a much smaller accuracy than the final target, which is precisely the source of the complexity inflation.

In the present construction, the second-stage error instead appears with coefficients θ_p and takes the form

$$\sum_{p=1}^{\varepsilon^{-\gamma}} \theta_p\, \delta = \delta \sum_{p=1}^{\varepsilon^{-\gamma}} \theta_p.$$

The key point is that the coefficients are constructed so as to form a partition of unity, that is $\sum_{p=1}^{\varepsilon^{-\gamma}} \theta_p \approx 1$ with a controllable error. Hence the total second-stage contribution is simply of order δ, rather than $\varepsilon^{-\gamma}\delta$. To keep the global error of order ε, it is therefore enough to choose

$$\delta \asymp \varepsilon.$$

This implies that the second-stage approximation no longer suffers an additional amplification through the number of first-stage terms. This is the key mechanism behind the improved rates proved below.

The terms *nested* and *parallel* describe the interaction between the two approximation stages. In the nested case, the Stage 1 representation propagates into the Stage 2 error estimate and forces a smaller Stage 2 tolerance. In the parallel case, the Stage 2 error estimate is decoupled from the number of Stage 1 terms, so its required accuracy remains at the scale of the final target error.

	Nested strategy	**Parallel strategy**
Stage 1 output	$\varepsilon^{-\gamma}$ terms	$\varepsilon^{-\gamma}$ terms with coefficients θ_p
Stage 2 error contribution	$\sum_{p=1}^{\varepsilon^{-\gamma}} \delta$	$\sum_{p=1}^{\varepsilon^{-\gamma}} \theta_p \, \delta = \delta \sum_{p=1}^{\varepsilon^{-\gamma}} \theta_p$
Coefficient structure	no partition of unity	approximate partition of unity, $\sum_{p=1}^{\varepsilon^{-\gamma}} \theta_p \approx 1$
Total Stage 2 error	$\varepsilon^{-\gamma}\delta$	δ
Condition for global error $\asymp \varepsilon$	$\varepsilon^{-\gamma}\delta \asymp \varepsilon$	$\delta \asymp \varepsilon$
Required Stage 2 accuracy	$\delta \asymp \varepsilon^{\gamma+1}$	$\delta \asymp \varepsilon$
Effect on complexity	amplification of Stage 2	no amplification through the number of terms

Table 2: Schematic comparison of how the second-stage approximation error enters in the nested construction of [47] and in the parallel construction introduced in Theorem 3.1. In the nested case, the Stage 2 error accumulates over $\varepsilon^{-\gamma}$ terms, forcing the per-term accuracy to be of order $\varepsilon^{\gamma+1}$ in order to maintain a global error of order ε. In the parallel case, the coefficients form an approximate partition of unity, so that the Stage 2 error factors as $\delta \sum_p \theta_p \approx \delta$, and it therefore suffices to choose $\delta \asymp \varepsilon$.

Theorem 3.1 (Scaling Laws for Multiple Operator Learning with the General Separable Architecture)**.** *Consider integers* $d_W, d_U, d_V > 0$*,*

$$\gamma_W, \gamma_U, \gamma_V, \beta_W, \beta_U, \beta_V, L_W, L_U, L_V, L_G, L_{\mathcal{G}} > 0 \qquad \text{and} \qquad r_G, r_{\mathcal{G}} \geq 1$$

and assume that $W(d_W, \gamma_W, L_W, \beta_W)$*,* $U(d_U, \gamma_U, L_U, \beta_U)$ *and* $V(d_V, \gamma_V, L_V, \beta_V)$ *satisfy Assumption* ***S.1****. Let* G *be a map satisfying Assumptions* ***O.1*** *and* ***O.2****. There exist constants*

- C *depending on* γ_V*,* L_V
- C' *depending on* β_U*,* $L_{\mathcal{G}}$*,* γ_U*,* $r_{\mathcal{G}}$
- C_U *depending on* $L_{\mathcal{G}}$*,* γ_U*,* $r_{\mathcal{G}}$*,* L_U
- C'' *depending on* β_W*,* L_G*,* γ_W*,* r_G
- C_ζ *depending on* L_G*,* γ_W*,* r_G*,* L_W

such that the following holds. For $\varepsilon > 0$ *sufficiently small,*

- *let* $N := 4C\sqrt{d_V}\varepsilon^{-1}$ *and consider the network class* $\mathcal{F}_1 := \mathcal{F}_{\mathrm{NN}}(d_V, 1, L_1, p_1, K_1, \kappa_1, R_1)$ *whose parameters scale as*

$$\begin{aligned} &L_1 = \mathcal{O}\left(d_V^2 \log d_V + d_V^2(\log(\varepsilon^{-1}) + 2\log(2))\right), \qquad && p_1 = \mathcal{O}(1), \\ &K_1 = \mathcal{O}\left(d_V^2 \log d_V + d_V^2(\log(\varepsilon^{-1}) + 2\log(2))\right), \qquad && \kappa_1 = \mathcal{O}(d_V^{d_V/2+1}\varepsilon^{-(d_V+1)}2^{2(d_V+1)}), \\ &R_1 = 1 \end{aligned}$$

  where the constants hidden in $\mathcal{O}$ *depend on* γ_V *and* L_V*;*

- *let* $\delta_U = C_U\varepsilon/2$ *and let* $\{c_i\}_{i=1}^{n_{c_U}} \subset \Omega_U$ *be points so that* $\{\mathcal{B}_{\delta_U}(c_i)\}_{i=1}^{n_{c_U}}$ *is a cover of* Ω_U *for some* n_{c_U}*;*

- *let* $H := 8C'\sqrt{n_{c_U}}\varepsilon^{-1}$ *and consider the network class* $\mathcal{F}_2 := \mathcal{F}_{\mathrm{NN}}(n_{c_U}, 1, L_2, p_2, K_2, \kappa_2, R_2)$ *with parameters scaling as*

$$L_2 = \mathcal{O}\left(n_{c_U}^2 \log(n_{c_U}) + (n_{c_U}^2[\log(\varepsilon^{-1}) + 3\log(2)]\right), \qquad p_2 = \mathcal{O}(1),$$

$$K_2 = \mathcal{O}\left(n_{c_U}^2 \log(n_{c_U}) + n_{c_U}^2[\log(\varepsilon^{-1}) + 3\log(2)]\right),$$
$$\kappa_2 = \mathcal{O}(n_{c_U}^{n_{c_U}/2+1}\varepsilon^{-(n_{c_U}+1)}2^{3(n_{c_U}+1)}), \qquad R_2 = 1$$

where the constants hidden in $\mathcal{O}$ depend on β_U, $L_{\mathcal{G}}$, γ_U, $r_{\mathcal{G}}$;

- *let $\zeta := C_\zeta \varepsilon/2$ and let $\{y_m\}_{m=1}^{n_{c_W}} \subset \Omega_W$ be points so that $\{\mathcal{B}_\zeta(y_m)\}_{m=1}^{n_{c_W}}$ is a cover of Ω_W for some n_{c_W}*
- *let $P = 2C''\sqrt{n_{c_W}}\varepsilon^{-1}$ and consider the network class $\mathcal{F}_3 = \mathcal{F}_{\mathrm{NN}}(n_{c_W}, 1, L_3, p_3, K_3, \kappa_3, R_3)$ whose parameters scale as*

$$L_3 = \mathcal{O}\left(n_{c_W}^2 \log(n_{c_W}) + n_{c_W}^2(\log(\varepsilon^{-1}) + 2\log(2))\right), \quad p_3 = \mathcal{O}(1),$$
$$K_3 = \mathcal{O}\left(n_{c_W}^2 \log(n_{c_W}) + n_{c_W}^2(\log(\varepsilon^{-1}) + 2\log(2))\right),$$
$$\kappa_3 = \mathcal{O}(n_{c_W}^{n_{c_W}/2+1}2^{2(n_{c_W}+1)}\varepsilon^{-n_{c_W}-1}), \qquad R_3 = 1$$

where the constants hidden in $\mathcal{O}$ depend on β_W, L_G, γ_W, r_G.

Then, there exists:

- *networks $\{\tau_\ell\}_{k=1}^{N^{d_V}} \subset \mathcal{F}_1$, $\{b_k\}_{k=1}^{H^{n_{c_U}}} \subset \mathcal{F}_2$ and $\{l_p\}_{p=1}^{P^{n_{c_W}}} \subset \mathcal{F}_3$*
- *functions $\{u_k\}_{k=1}^{H^{n_{c_U}}} \subset \mathcal{B}_{\beta_U, \|\cdot\|_{\mathrm{L}^\infty}, \Omega_U}(0)$ and $\{\alpha_p\}_{p=1}^{P^{n_{c_W}}} \subset \mathcal{B}_{\beta_W, \|\cdot\|_{\mathrm{L}^\infty}, \Omega_W}(0)$*
- *points $\{v_\ell\}_{\ell=1}^{N^{d_V}} \subset \Omega_V$*

such that

$$\sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}\left|G[\alpha][u](x) - \sum_{p=1}^{P^{n_{c_W}}}\sum_{k=1}^{H^{n_{c_U}}}\sum_{\ell=1}^{N^{d_V}} G[\alpha_p][u_k](v_\ell)l_p(\boldsymbol{\alpha})b_k(\boldsymbol{u})\tau_\ell(x)\right| \leq \varepsilon \tag{5}$$

where $\boldsymbol{u} = (u(c_1), u(c_2), \ldots, u(c_{n_{c_U}}))^\top$ and $\boldsymbol{\alpha} = (\alpha(y_1), \ldots, \alpha(y_{n_{c_W}}))^\top$.

Remark 3.2 (Total Number of Parameters for Multiple Operator Learning with the General Separable Architecture). Let NN_ε be a network that satisfies (5). We want to estimate $\|\Theta\|_0 + K_1N^{d_V} + K_2H^{n_{c_U}} + K_3P^{n_{c_W}} \geq N_\#(\mathrm{NN}_\varepsilon)$ where $\Theta_0 = \{\theta_{pk\ell}\}_{p=1,k=1,\ell=1}^{P^{n_{c_W}},H^{n_{c_U}},N^{d_V}}$.

First, we note that the computation in Remark 3.26 applies to our case (with $n_{c_W} = 0$ and $d_U = 0$ since we considered $W \times \{\emptyset\} \cong W$) to compute $K_1N^{d_V} + K_2H^{n_{c_U}}$. Indeed, the scalings of $\mathcal{F}_1$ and $\mathcal{F}_2$ only differ by constants between Theorem 3.1 and Proposition 3.25. Specifically, we therefore have

$$H^{n_{c_U}} \lesssim \varepsilon^{-d\varepsilon^{-d_U}} \qquad \text{and} \qquad K_1N^{d_V} + K_2H^{n_{c_U}} \lesssim \varepsilon^{-d\varepsilon^{-d_U}} \tag{6}$$

for some $d > 0$ depending on d_U.

Next, by [31, Lemma 2], we have that $n_{c_W} \lesssim \zeta^{-d_W} \lesssim \varepsilon^{-d_W}$. We estimate

$$K_3P^{n_{c_W}} \lesssim \left(\varepsilon^{-2d_W}\log(\varepsilon^{-1})\right)\left(\varepsilon^{-d_W/2+1}\right)^{\varepsilon^{-d_W}} \lesssim \varepsilon^{-d'\varepsilon^{-d_W}} \tag{7}$$

where $d' > 0$ depends on d_W. Therefore, using (6) and (7), we have

$$\|\Theta\|_0 \leq N^{d_V}H^{n_{c_U}}P^{n_{c_W}} \lesssim \varepsilon^{-d'\varepsilon^{-d_W}}\varepsilon^{-d\varepsilon^{-d_U}} \lesssim \varepsilon^{-d''\varepsilon^{-\max\{d_W,d_U\}}}. \tag{8}$$

for some d'' depending on d_U and d_W. Combining (6), (7) and (8), we obtain:

$$\|\Theta\|_0 + K_1N^{d_V} + K_2H^{n_{c_U}} + K_3P^{n_{c_W}} \lesssim \varepsilon^{-d''\varepsilon^{-\max\{d_W,d_U\}}}.$$

Repeating the computation in [47, Remark 3.14], we can also deduce that the approximation error ε scales as follows:

$$\varepsilon \lesssim \left(\frac{\log N_\#}{\log\log N_\#}\right)^{-1/\max\{d_W,d_U\}}$$

	# networks		width		depth		sparsity		parameter magnitude	
	Existing	Proposed	Existing	Proposed	Existing	Proposed	Existing	Proposed	Existing	Proposed
l_p	$\varepsilon^{-\varepsilon^{-d_W}}$	$\varepsilon^{-\varepsilon^{-d_W}}$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	ε^{-d_W}	ε^{-2d_W}	ε^{-d_W}	ε^{-2d_W}	$\varepsilon^{-\varepsilon^{-d_W}}$	$\varepsilon^{-\varepsilon^{-d_W}}$
b_k	$\varepsilon^{-\varepsilon^{-\varepsilon^{-d_W}}}$	$\varepsilon^{-\varepsilon^{-d_U}}$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\varepsilon^{-\varepsilon^{-d_W}}$	ε^{-2d_U}	$\varepsilon^{-\varepsilon^{-d_W}}$	ε^{-2d_U}	$\varepsilon^{-\varepsilon^{-\varepsilon^{-d_W}}}$	$\varepsilon^{-\varepsilon^{-d_U}}$
τ_ℓ	$\varepsilon^{-\varepsilon^{-d_W}}$	ε^{-d_V}	$\mathcal{O}(1)$	$\mathcal{O}(1)$	ε^{-d_W}	$\log(\varepsilon^{-1})$	ε^{-d_W}	$\log(\varepsilon^{-1})$	$\varepsilon^{-\varepsilon^{-d_W}}$	$\varepsilon^{-(1+d_V)}$

Table 3: Comparison of the existing MNO constructive scalings from [47] and the proposed scalings derived in this work. Constants, lower-order and poly-logarithmic terms are suppressed; entries in the columns labelled Proposed correspond to the present work.

Remark 3.3 (Distribution of Complexity across Subnetworks). We recall that the subnetworks l_p, b_k, and τ_ℓ in (2) encode, respectively, the infinite-dimensional dependence on the parameter variable α, the infinite-dimensional dependence on the input function u, and the remaining finite-dimensional dependence on the output variable x. Beyond the improvement in total parameter scaling (see Remark 3.2), Table 3 also reveals a markedly different distribution of complexity across the subnetworks.

In the existing construction of [47], the complexity is highly unbalanced: the subnetworks l_p and τ_ℓ have comparable scalings, while the subnetworks b_k are dramatically more complex and therefore carry the main burden of the approximation. In this sense, the architecture is used in a strongly hierarchical manner, with the subnetworks appearing in the second-stage approximation, namely b_k and τ_ℓ, absorbing most of the complexity. We also note that the space-approximation subnetworks τ_ℓ have the same complexity scaling as the subnetworks l_p. This appears intuitively suboptimal, since one would expect the finite-dimensional spatial approximation step to be substantially simpler than the infinite-dimensional approximation of functional dependence.

By contrast, the proposed construction leads to a much more balanced allocation of complexity. The subnetworks l_p and b_k now have comparable scalings (both of the same general order as the previous l_p and τ_ℓ) while the subnetworks τ_ℓ become significantly simpler. Thus, the improved rates are also reflected in the internal organization of the architecture: rather than concentrating complexity in a single dominant block, the new proof distributes it more evenly across the infinite- and finite-dimensional approximation components.

The next result yields substantially improved generalization scaling laws. Its proof is driven by the rates established in Theorem 3.1: once the estimate (9) is established, the remaining step is to control the metric entropy of the associated hypothesis class. Since this metric entropy is governed by the architectural scalings of the network classes, the improvement in approximation complexity propagates directly to the statistical error bound. In particular, because the new approximation theory removes the additional parametric curse specific to multiple operator learning, the resulting learning rate is likewise reduced to the scale dictated by operator learning.

Theorem 3.4 (Scaling Laws for the Expected Generalization Error)**.** *Let $d_W, d_U, d_V > 0$ be integers,*

$$\gamma_W, \gamma_U, \gamma_V, \beta_W, \beta_U, \beta_V, L_W, L_U, L_V, L_G, L_{\mathcal{G}} > 0 \qquad \text{and} \qquad r_G, r_{\mathcal{G}} \geq 1$$

*and assume that $W(d_W, \gamma_W, L_W, \beta_W)$, $U(d_U, \gamma_U, L_U, \beta_U)$ and $V(d_V, \gamma_V, L_V, \beta_V)$ satisfy Assumption **S.1**. Let G be a map satisfying Assumptions **O.1** and **O.2**.*

There exist constants C, C_δ, C', C_ζ, and C'', depending on the same quantities as in Theorem 3.1, such that the following holds. For any $\varepsilon > 0$, use the latter constants to define N, δ, H, ζ, P, the network classes $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$, and the sampling points $\{c_m\}_{m=1}^{n_{c_U}}$, $\{y_m\}_{m=1}^{n_{c_W}}$ as in Theorem 3.1, with ε replaced everywhere by $\varepsilon/2$.

Let $a = \beta_V$, $I \geq \beta_V$, $n_\alpha, n_u, n_x \in \mathbb{N}$, μ_α a probability measure on W, μ_u a probability measure on U, and μ_x a probability measure on Ω_V. Consider the clipped network class

$$\mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P^{n_{c_W}}, H^{n_{c_U}}, N^{d_V}).$$

For $\eta > 0$, the expected generalization error is bounded as follows:

$$\mathbb{E}_{S_{G,\{y_s\},\{c_s\}}} \mathbb{E}_{\alpha\sim\mu_\alpha} \mathbb{E}_{u\sim\mu_u} \mathbb{E}_{\{x_j\}_{j=1}^{n_x}\sim\mu_x^{\otimes n_x}} \left[\frac{1}{n_x} \sum_{j=1}^{n_x} \left(G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}[\boldsymbol{\alpha}][\mathbf{u}](x_j) - G[\alpha][u](x_j) \right)^2 \right]$$

$$
\begin{aligned}
&\leq 4\varepsilon^2 + \eta(8\sigma + 6) \\
&+ \frac{8\sigma\eta}{\sqrt{n_\alpha n_u n_x}} \sqrt{\log\left(\mathcal{N}\left(\eta, \mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P^{n_{c_W}}, H^{n_{c_U}}, N^{d_V}), \|\cdot\|_{\mathrm{L}^\infty(W\times U\times\Omega_V)}\right)\right) + \log(2)} \\
&+ \frac{16\sigma^2}{n_\alpha n_u n_x}\left(\log\left(\mathcal{N}\left(\eta, \mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P^{n_{c_W}}, H^{n_{c_U}}, N^{d_V}), \|\cdot\|_{\mathrm{L}^\infty(W\times U\times\Omega_V)}\right)\right) + \log(2)\right) \\
&+ \frac{112\beta_V^2}{3n_\alpha}\log\left(\mathcal{N}\left(\eta/(4\beta_V), \mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P^{n_{c_W}}, H^{n_{c_U}}, N^{d_V}), \|\cdot\|_{\mathrm{L}^\infty(W\times U\times\Omega_V)}\right)\right)
\end{aligned} \tag{9}
$$

where $\boldsymbol{\alpha} = (\alpha(y_1), \alpha(y_2), ..., \alpha(y_{n_{c_W}}))^\top$ *and* $\mathbf{u} = (u(c_1), u(c_2), ..., u(c_{n_{c_U}}))^\top$.

Furthermore, if we pick

$$
\varepsilon = \left(\frac{\max\{d_W, d_U\}}{4(1+\max\{d_W, d_U\}/2)}\frac{\log n_\alpha}{\log\log n_\alpha}\right)^{-1/\max\{d_W, d_U\}} \qquad \textit{and} \qquad \eta = 4\beta_V n_\alpha^{-1},
$$

then the expected generalization error scales as follows:

$$
\begin{aligned}
&\mathbb{E}_{S_{G,\{y_s\},\{c_s\}}}\mathbb{E}_{\alpha\sim\mu_\alpha}\,\mathbb{E}_{u\sim\mu_u}\,\mathbb{E}_{\{x_j\}_{j=1}^{n_x}\sim\mu_x^{\otimes n_x}}\left[\frac{1}{n_x}\sum_{j=1}^{n_x}\left(G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}[\boldsymbol{\alpha}][\mathbf{u}](x_j) - G[\alpha][u](x_j)\right)^2\right] \\
&= \mathcal{O}\left(\left(\frac{\log n_\alpha}{\log\log n_\alpha}\right)^{-2/\max\{d_W, d_U\}}\right)
\end{aligned}
$$

where the constants hidden in $\mathcal{O}$ *are independent of* n_α, n_u, n_x.

Remark 3.5 (Exchanging the sampling hierarchy). If one modifies the sampling procedure in Definition 2.5 by exchanging the roles of α and u, then one expects the corresponding learning rate to take the same form as above, but with the parameter-side quantities replaced by their input-side counterparts. In particular, one would expect a learning rate of the form

$$
\mathcal{O}\left(\left(\frac{\log n_u}{\log\log n_u}\right)^{-2/\max\{d_W, d_U\}}\right).
$$

3.2 Approximation Complexity Lower Bounds and Minimax Rates for MNO

To extend the lower-complexity framework of [26] to the multiple operator setting, we first introduce the natural analogue of an operator of neural network type. The key idea is to require that, after fixing an input function u and an evaluation point x, the resulting dependence on the parameter variable α is represented by a finite-dimensional linear encoding followed by a ReLU network.

Definition 3.6 (Multiple Operator Map of Neural Network Type)**.** *We say that* $\mathrm{NN} : W \to \{U \to V\}$ *is a multiple operator map of neural network type if for every* $u \in U$ *and* $x \in \Omega_V$*,*

$$
\mathrm{ev}_x \circ \mathrm{ev}_u \circ \mathrm{NN}[\alpha] = \Phi_{x, M_U(u)}(M_W(\alpha)) \qquad \textit{for all } \alpha \in W \tag{10}
$$

where $M_W : W \to \mathbb{R}^m$ *and* $M_U : U \to \mathbb{R}^q$ *are linear, and* $\Phi_{x,M_U(u)}$ *is a ReLU neural network. We define the complexity of* NN *as*

$$
\mathcal{C}(\mathrm{NN}) = \sup_{x\in\Omega_V}\sup_{u\in U}\min_{\Phi_{x,M_U(u)}\ \textit{satisfying (10)}} N_{\#}\left(\Phi_{x,M_U(u)}\right).
$$

Remark 3.7 (Limitations of the Classical Neural-network-type Framework). The notion of operator of neural network type from Definitions 2.10 is formulated for maps that can be written in the form $u \mapsto \Phi(M(u))$, where M is linear and Φ is a ReLU network. As recalled in Section 2.5, this framework is well suited to a range of classical operator-learning architectures, as well as to concatenation-based models, where all inputs are first assembled into a single finite-dimensional vector and then processed jointly by one network (see Section 3.3).

By contrast, the separable architecture (2) has the form

$$\sum_{p=1}^{P}\sum_{k=1}^{H}\sum_{\ell=1}^{N}\theta_{pk\ell}\, l_p(M_W(\alpha))\, b_k(M_U(u))\, \tau_\ell(x),$$

and is therefore built from products of subnetworks. To cast such an expression into the form $\Phi(M(\alpha,u))$, one would need to absorb terms such as

$$l_p(M_W(\alpha))\, b_k(M_U(u))$$

into a single ReLU network acting on a linear encoding of (α,u). Such closure under multiplication fails already for simple non-constant subnetworks. Indeed, the identity map $x\mapsto x$ is exactly representable by a finite ReLU network, for instance via $x=\mathrm{ReLU}(x)-\mathrm{ReLU}(-x)$, whereas multiplying this representation by itself yields the quadratic map $x\mapsto x^2$, which is not piecewise affine. This is precisely why Definition 3.6 is needed: it is tailored to the separable structure of MNO-type architectures.

The abstract lower-bound argument from [26] carries over directly to the multiple operator setting and yields the following curse of parametric complexity. The proof is based on lifting a pathological functional $F: W\to\mathbb{R}$, known to exhibit the curse of parametric complexity, to a rank-one multiple operator map of the form

$$G[\alpha][u](x)=F(\alpha)\phi(x),$$

where $\phi\in V$ is fixed and nontrivial. Evaluating G at (u_0,x_0) recovers $F(\alpha)$, so any low-complexity neural approximation of G would in particular yield a low-complexity neural approximation of F. The desired lower bound therefore follows from the corresponding functional lower bound of [26, Theorem 2.11].

Theorem 3.8 (Curse of Parametric Complexity for Multiple Operator Learning)**.** *Let K be a compact subset of the Banach space W containing an infinite-dimensional cube Q_η for some $\eta>1$. Let V be a Banach space continuously embedded in $C(\Omega_V)$, and let U be a set of functions.*

Then, for any $r\in\mathbb{N}$ and $\delta>0$, there exists an r-times Frechet differentiable map $G: W\to\{U\to V\}$ and $\overline{\varepsilon}:=\overline{\varepsilon}(\eta,\delta,r)>0$ such that for any $\varepsilon\le\overline{\varepsilon}$ and multiple operator map of neural network type $\mathrm{NN}_\varepsilon: W\to\{U\to V\}$ satisfying

$$\sup_{\alpha\in K}\|\,\mathrm{NN}_\varepsilon[\alpha]-G[\alpha]\|_{\mathrm{op}}\le\varepsilon, \tag{11}$$

we have $\mathcal{C}(\mathrm{NN}_\varepsilon)\ge\exp\left(c\varepsilon^{-1/[(\eta+1+\delta)r]}\right)$ for some $c:=c(\eta,\delta,r)>0$.

Theorem 3.8 gives a lower-complexity principle for multiple operator learning in a fairly general setting. However, the separable architecture (2) has an additional structural feature: the parameter variable α and the input function u play analogous roles (from the analytical viewpoint). It is therefore natural to strengthen Definition 3.6 by requiring a neural-network-type representation in both variables. This leads to the following symmetric variant.

Definition 3.9 (Symmetric Multiple Operator Map of Neural Network Type)**.** *We say that $\mathrm{NN}: W\times U\to V$ is a symmetric multiple operator map of neural network type if there exists linear maps $M_W: W\to\mathbb{R}^m$ and $M_U: U\to\mathbb{R}^q$ such that the following holds for every $x\in\Omega_V$:*

- *for every $u\in U$, we have the representation*

$$\mathrm{ev}_x\circ\mathrm{ev}_u\circ\mathrm{NN}_W[\alpha]=\Phi_{x,M_U(u)}(M_W(\alpha))\qquad\text{for all }\alpha\in W \tag{12}$$

  where $\Phi_{x,M_U(u)}$ is a ReLU neural network and $\mathrm{NN}_W: W\to\{U\to V\}$ is defined as $\mathrm{NN}_W[\alpha]=\mathrm{NN}[\alpha][\cdot](\cdot)$.

- *for every $\alpha\in W$, we have the representation*

$$\mathrm{ev}_x\circ\mathrm{ev}_\alpha\circ\mathrm{NN}_U[u]=\Psi_{x,M_W(\alpha)}(M_U(u))\qquad\text{for all }u\in U \tag{13}$$

  where $\Psi_{x,M_W(\alpha)}$ is a ReLU neural network and $\mathrm{NN}_U: U\to\{W\to V\}$ is defined as $\mathrm{NN}_U[u]=\mathrm{NN}[\cdot][u](\cdot)$.

We define the complexity of NN *as*

$$\mathcal{C}(\mathrm{NN}) = \max\left\{ \sup_{x\in\Omega_V}\sup_{u\in U} \min_{\substack{\Phi_{x,M_U(u)} \\ \textit{satisfying (12)}}} N_{\#}\left(\Phi_{x,M_U(u)}\right), \sup_{x\in\Omega_V}\sup_{\alpha\in W} \min_{\substack{\Psi_{x,M_W(\alpha)} \\ \textit{satisfying (13)}}} N_{\#}\left(\Psi_{x,M_W(\alpha)}\right) \right\}.$$

This directly leads to the following curse of parametric complexity for symmetric multiple operator map of neural network type.

Lemma 3.10 (Symmetric Curse of Parametric Complexity for Multiple Operator Learning)**.** *Let W, U be Banach spaces and $K_W \subset W$, $K_U \subset U$ be compact subsets. Assume that K_W and K_U contain infinite-dimensional cubes Q_{η_W} and Q_{η_U} for some $\eta_W > 1$ and $\eta_U > 1$, respectively. Let V be a Banach space continuously embedded in $C(\Omega_V)$. We equip the product Banach space $W \times U$ with a norm $\|\cdot\|_{W\times U}$ satisfying Assumption **N.1**.*

Then, for any $r \in \mathbb{N}$ and $\delta > 0$, there exists an r-times Frechet differentiable map $G : W \times U \to V$ and $\overline{\varepsilon} := \overline{\varepsilon}(\eta_W, \eta_U, \delta, r) > 0$ such that for any $\varepsilon \leq \overline{\varepsilon}$ and symmetric multiple operator map of neural network type $\mathrm{NN}_\varepsilon : W \times U \to V$ satisfying

$$\sup_{\alpha\in K_W}\sup_{u\in K_U} \|\,\mathrm{NN}_\varepsilon[\alpha][u] - G[\alpha][u]\|_V = \sup_{u\in K_U}\sup_{\alpha\in K_W} \|\,\mathrm{NN}_\varepsilon[\alpha][u] - G[\alpha][u]\|_V \leq \varepsilon, \tag{14}$$

we have $\mathcal{C}(\mathrm{NN}_\varepsilon) \geq \exp\left(c\varepsilon^{-1/[(\min\{\eta_W,\eta_U\}+1+\delta)r]}\right)$ for some $c := c(\eta_W, \eta_U, \delta, r) > 0$.

Remark 3.11 (Choice of Operator Norm and Banach Space Assumption for V). The proof of Theorem 3.8 does not rely on the specific choice

$$\|T\|_{\mathrm{op}} = \sup_{u\in U} \|T(u)\|_V$$

except through the estimate $|T(u_0)(x_0)| \leq C\|T\|_{\mathrm{op}}$ for some fixed $u_0 \in U$, $x_0 \in \Omega_V$, and some constant $C > 0$; see (40). Hence, the same argument applies to any norm on the operator space for which the slice-evaluation map $T \mapsto T(u_0)(x_0)$ is continuous.

In particular, one may replace $\|\cdot\|_{\mathrm{op}}$ by the uniform pointwise norm

$$\|T\|_{\mathrm{op},\infty} := \sup_{u\in U}\sup_{x\in\Omega_V} |T(u)(x)|.$$

With this choice, the proof becomes slightly simpler, since $|T(u_0)(x_0)| \leq \|T\|_{\mathrm{op},\infty}$ holds with constant 1, and the continuous embedding $V \hookrightarrow C(\Omega_V)$ is no longer needed. Accordingly, in the proof of Theorem 3.8, the constant C_V disappears and one may take $\overline{\varepsilon} = \varepsilon_0$. Moreover, the approximation condition (11) becomes $\sup_{\alpha\in K} \|\,\mathrm{NN}_\varepsilon[\alpha] - G[\alpha]\|_{\mathrm{op},\infty} \leq \varepsilon$, and one obtains $\sup_{\alpha\in K} |F(\alpha) - \mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \mathrm{NN}_\varepsilon[\alpha]| \leq \varepsilon$ directly.

A further advantage of this formulation is that the proof no longer requires V to be a Banach space. Indeed, in the proof based on $\|\cdot\|_{\mathrm{op}}$, the estimate (40) requires that

$$\big(G[\alpha] - \mathrm{NN}_\varepsilon[\alpha]\big)(u_0) \in V$$

so that its V-norm is well-defined. By contrast, for the norm $\|\cdot\|_{\mathrm{op},\infty}$ it suffices that the outputs are actual bounded functions on Ω_V, since the proof only uses pointwise differences. Therefore, the same argument applies if V satisfies Assumption **S.1**, rather than being Banach space.

The same remark applies to Lemma 3.10 with the norm choice of $\sup_{\alpha\in K_W}\sup_{u\in K_U} \|\cdot\|_{\mathrm{L}^\infty}$.

Remark 3.12 (Role of the Symmetry Assumptions). The symmetric approximation norm

$$\sup_{\alpha\in K_W}\sup_{u\in K_U} \|\,\mathrm{NN}_\varepsilon[\alpha][u] - G[\alpha][u]\|_V$$

and the two symmetric neural network representations in (12) and (13) are essential in Lemma 3.10, since the proof may be carried out starting either from the parameter side W or from the input side U, depending on which cube exponent is smaller.

Now that we have established the curse of parametric complexity for multiple operator learning in an abstract setting, we turn to the specific function classes used in our approximation-theoretical result, Theorem 3.1. To apply the lower-bound framework in that setting, the first step is to verify that the bounded Lipschitz classes from Assumption **S.1** contain an infinite-dimensional hypercube. The next result addresses precisely this point.

Lemma 3.13 (Infinite-dimensional Cube in Bounded Lipschitz Classes)**.** *Let $d_U \in \mathbb{N}$, $\gamma_U, L_U, \beta_U > 0$ and $U = U(d_U, \gamma_U, L_U, \beta_U)$ satisfy Assumption **S.1**. Then, viewed as a subset of the Banach space $\mathrm{L}^r(\Omega_U)$ with $r \geq 1$, U contains an infinite-dimensional hypercube Q_η for every $\eta > 1 + \frac{1}{d_U}$.*

Combining the latter result and Lemma 3.10, we obtain the following.

Corollary 3.14 (Curse of Parametric Complexity for Multiple Operator Learning on Bounded Lipschitz Classes)**.** *Let $d_W, d_U, d_V > 0$ be integers,*

$$\gamma_W, \gamma_U, \gamma_V, \beta_W, \beta_U, \beta_V, L_W, L_U, L_V > 0 \quad \textit{and} \quad r_G \geq 1$$

*and assume that $W(d_W, \gamma_W, L_W, \beta_W)$, $U(d_U, \gamma_U, L_U, \beta_U)$ and $V(d_V, \gamma_V, L_V, \beta_V)$ satisfy Assumption **S.1**. We equip the product Banach space $\mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U)$ with a norm $\|\cdot\|_{\mathrm{L}^{r_G}(\Omega_W)\times\mathrm{L}^{r_G}(\Omega_U)}$ that satisfies Assumption **N.1**.*

Then, for any $\eta > \min\left\{1 + \frac{1}{d_W}, 1 + \frac{1}{d_U}\right\}$, $r \in \mathbb{N}$ and $\delta > 0$, there exists an r-times Frechet differentiable map $G : \mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U) \to V$ and $\overline{\varepsilon} := \overline{\varepsilon}(\eta, \delta, r) > 0$ such that for any $\varepsilon \leq \overline{\varepsilon}$ and symmetric multiple operator map of neural network type NN_ε satisfying

$$\sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V} |\,\mathrm{NN}_\varepsilon[\alpha][u](x) - G[\alpha][u](x)| \leq \varepsilon,$$

we have $\mathcal{C}(\mathrm{NN}_\varepsilon) \geq \exp\left(c\varepsilon^{-1/[(\eta+1+\delta)r]}\right)$ for some $c := c(\eta, \delta, r) > 0$.

Remark 3.15 (Asymmetric Structure). If one were to rely only on Theorem 3.8, rather than on Lemma 3.10, in the proof of Corollary 3.14, then one would not fully exploit the symmetric role played by the parameter space and the input-function space in the bounded Lipschitz setting. Specifically, one would obtain an analogous result for (asymmetric) multiple operator maps of neural network type under the more restrictive condition $\eta > 1 + \frac{1}{d_W}$.

Corollary 3.14 provides a lower bound on approximation complexity for multiple operator maps satisfying Assumptions **O.1** and **O.2** using general symmetric multiple operator maps of neural network type defined on bounded Lipschitz classes. It only remains to connect this result directly to the architecture of interest (2). To do so, we estimate the complexity of the corresponding network class in the next result.

Lemma 3.16 (Complexity of the General Separable Architecture)**.** *The network given by*

$$\mathrm{NN}[\alpha][u](x) = \sum_{p=1}^{P}\sum_{k=1}^{H}\sum_{\ell=1}^{N} \theta_{pk\ell}\, l_p(M_W(\alpha))\, b_k(M_U(u))\, \tau_\ell(x) \qquad \theta_{pk\ell} \in \mathbb{R}$$

with $l_p \in \mathcal{F}_{\mathrm{NN}}(d_3, 1, L_3, p_3, K_3, \kappa_3, 1)$, $b_k \in \mathcal{F}_{\mathrm{NN}}(d_2, 1, L_2, p_2, K_2, \kappa_2, 1)$ and $\tau_\ell \in \mathcal{F}_{\mathrm{NN}}(d_1, 1, L_1, p_1, K_1, \kappa_1, 1)$ is a symmetric multiple operator map of neural network type. Furthermore, we have

$$\mathcal{C}(\mathrm{NN}) \leq 2(\|\Theta\|_0 + HK_2 + NK_1 + PK_3)$$

where $\Theta = \{\theta_{pk\ell}\}$.

Remark 3.17 (Asymmetric Complexity of the General Separable Architecture). From Definitions 3.6 and 3.9, we note that any symmetric multiple operator map of neural network type is also an asymmetric one. Also, its complexity measured under the symmetric setting is an upper bound to its complexity in the asymmetric case. This observation specifically applies to the general separable architecture (2) and, given Lemma 3.16, it is unnecessary to compute the complexity separately under the asymmetric notion.

Remark 3.18 (Complexity of the MNO architecture). The same argument as in Lemma 3.16 shows that the MNO architecture

$$\mathrm{NN}[\alpha][u](x) = \sum_{p=1}^{P} \sum_{k=1}^{H} l_p(M_W(\alpha))\, b_{pk}(M_U(u))\, \tau_{pk}(x)$$

is a symmetric multiple operator map of neural network type. Moreover,

$$\mathcal{C}(\mathrm{NN}) \leq 2(HPK_2 + HPK_1 + PK_3) = 2P\big(H(K_1 + K_2) + K_3\big).$$

We conclude this section with minimax rates for Lipschitz multiple operator maps using the architecture (2), as a combination of Corollary 3.14, Lemma 3.16 and Theorem 3.1.

Theorem 3.19 (Minimax Bounds for Lipschitz Multiple Operator Maps)**.** *Let $d_W, d_U, d_V > 0$ be integers,*

$$\gamma_W, \gamma_U, \gamma_V, \beta_W, \beta_U, \beta_V, L_W, L_U, L_V > 0 \quad \textit{and} \quad r_G \geq 1$$

*and assume that $W(d_W, \gamma_W, L_W, \beta_W)$, $U(d_U, \gamma_U, L_U, \beta_U)$ and $V(d_V, \gamma_V, L_V, \beta_V)$ satisfy Assumption **S.1**. We equip the product Banach space $\mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U)$ with a norm $\|\cdot\|_{\mathrm{L}^{r_G}(\Omega_W)\times\mathrm{L}^{r_G}(\Omega_U)}$ that satisfies Assumption **N.1**.*

1. *For any $\eta > \min\Big\{1 + \frac{1}{d_W}, 1 + \frac{1}{d_U}\Big\}$, $r \in \mathbb{N}$ and $\delta > 0$, there exists an r-times Frechet differentiable map $G : \mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U) \to V$ and $\overline{\varepsilon} := \overline{\varepsilon}(\eta, \delta, r) > 0$ such that the following holds: for any $\varepsilon \leq \overline{\varepsilon}$ and NN_ε of the form* (2) *satisfying*

   $$\sup_{\alpha\in W} \sup_{u\in U} \sup_{x\in\Omega_V} |\,\mathrm{NN}_\varepsilon[\alpha][u](x) - G[\alpha][u](x)| \leq \varepsilon, \tag{15}$$

   we have

   $$\|\Theta\|_0 + HK_2 + NK_1 + PK_3 \gtrsim \exp\Big(c\varepsilon^{-1/[(\eta+1+\delta)r]}\Big)$$

   for some $c := c(\eta, \delta, r) > 0$ and where $\Theta = \{\theta_{pk\ell}\}$.

2. *With $r = 1$, the map G in part 1. can be chosen so that it satisfies Assumptions **O.1** and **O.2**.*

3. *Let $\mathcal{H}$ denote the class of all maps G satisfying Assumptions **O.1** and **O.2**. Define the worst-case/minimax approximation complexity*

   $$\mathfrak{C}(\varepsilon; \mathcal{H}) := \inf\Big\{ M \in \mathbb{N} \,\Big|\, \forall G \in \mathcal{H},\ \exists\, \mathrm{NN}_\varepsilon \textit{ of the form } (2) \textit{ satisfying } (15) \textit{ and } \|\Theta\|_0 + HK_2 + NK_1 + PK_3 \leq M \Big\}.$$

   Let $r = 1$, $\eta > \min\Big\{1 + \frac{1}{d_W}, 1 + \frac{1}{d_U}\Big\}$, and $\delta > 0$. Then, for all $0 < \varepsilon \leq \overline{\varepsilon}(\eta, \delta, 1)$,

   $$\exp\Big(c\,\varepsilon^{-1/[(\eta+1+\delta)]}\Big) \lesssim \mathfrak{C}(\varepsilon; \mathcal{H}) \lesssim \exp\Big(d\log(\varepsilon^{-1})\varepsilon^{-\max\{d_W, d_U\}}\Big) \tag{16}$$

   for some $d > 0$ depending on d_W and d_U.

Remark 3.20 (Upper Bounds under Higher Regularity). The lower bound above is obtained from a general C^r-differentiable construction, and therefore applies more broadly than the Lipschitz class $\mathcal{H}$ considered in the present minimax formulation. This suggests that, for classes of C^r multiple operator maps, one should likewise expect constructive upper bounds of a similar double-exponential type.

The lower and upper bounds in (16) are both of exponential type in a diverging function of ε^{-1}, and therefore capture the same qualitative curse of parametric complexity on the class $\mathcal{H}$. The upper bound involves a faster-growing exponent, reflecting both a larger power of ε^{-1} and an additional logarithmic factor. Thus, although the two estimates are not tight at the level of the exponent, they are consistent in identifying the same overall double-exponential complexity regime.

This comparison also shows that, as in operator learning [26], Lipschitz or, more generally, C^r-regularity alone is not sufficient to overcome the curse of parametric complexity in multiple operator learning. Rather, one must exploit additional structure in the target multiple operator maps. Two important mechanisms in this direction are holomorphic dependence on the inputs or parameters, and architectures designed to emulate underlying numerical schemes [26, 37]. In operator learning, both have already been explored as ways of overcoming the curse of parametric complexity (see Section 1.2). In the multiple operator setting, however, they appear to remain largely unexplored, and investigating their potential to improve approximation complexity would therefore be of clear interest.

3.3 An Extension of DeepONet to Multi-Task Learning

In this final section, we examine an alternative approach to multiple operator learning. A particularly direct strategy is to start from a standard single-operator learning architecture and incorporate the parametric descriptor α simply by concatenating it with the input function u. It is therefore natural to ask how this concatenation affects approximation complexity. To address this question, we derive bounds for the concatenated DeepONet architecture (17) introduced below.

As in our analysis of the MNO architecture, instead of working directly with the concatenated DeepONet architecture

$$\mathrm{DeepONet_C}[\alpha][u](x) = \sum_{k=1}^{H} b_k(M_W(\alpha), M_U(u))\tau_k(x), \tag{17}$$

we consider the more general separable form

$$\sum_{k=1}^{H}\sum_{\ell=1}^{N} \theta_{k\ell}\, b_k(M_W(\alpha), M_U(u))\, \tau_\ell(x), \qquad \theta_{k\ell} \in \mathbb{R}. \tag{18}$$

We begin by showing that the concatenated DeepONet architecture (17) is an operator of neural network type, thereby placing it directly within the lower-complexity framework of [26]. This is to be expected, since the concatenated model can be interpreted as an ordinary operator-learning problem posed on the product space $W \times U$. We state the result, however, for the more general separable architecture (18). The proof therefore is a combination of the strategy used in Lemma 3.16 and the argument underlying [26, Lemma 2.20].

Lemma 3.21 (Complexity of the General Separable Architecture with Concatenated Inputs)**.** *The network given by*

$$\mathrm{NN}[\alpha][u](x) = \sum_{k=1}^{H}\sum_{\ell=1}^{N} \theta_{k\ell}\, b_k(M_W(\alpha), M_U(u))\, \tau_\ell(x), \qquad \theta_{k\ell} \in \mathbb{R}$$

with $b_k \in \mathcal{F}_{\mathrm{NN}}(d_2, 1, L_2, p_2, K_2, \kappa_2, 1)$ and $\tau_\ell \in \mathcal{F}_{\mathrm{NN}}(d_1, 1, L_1, p_1, K_1, \kappa_1, 1)$ is an operator map of neural network type from $W \times U$ to V. Furthermore, we have

$$\mathcal{C}(\mathrm{NN}) \le 2\big(\|\Theta\|_0 + NK_1 + HK_2\big).$$

Remark 3.22 (Symmetric Multiple Operator Map Interpretation of the Concatenated Separable Architecture). One may also view the architecture

$$\mathrm{NN}[\alpha][u](x) = \sum_{k=1}^{H}\sum_{\ell=1}^{N} \theta_{k\ell}\, b_k(M_W(\alpha), M_U(u))\, \tau_\ell(x)$$

as a symmetric multiple operator map of neural network type, rather than as an operator map of neural network type on the product space $W \times U$. In that case, however, the resulting complexity estimate is slightly less sharp. More precisely, one obtains the upper bound

$$\mathcal{C}(\mathrm{NN}) \le 2(\|\Theta\|_0 + NK_1 + H(K_2 + p_2)).$$

The additional $2Hp_2$ term arises from the fact that, in order to verify the symmetric definition, one must show that if $b(x, y)$ is a ReLU network in the concatenated variables (x, y), then freezing one block of variables still yields a ReLU network in the remaining variables. This operation modifies the first affine layer and increases the corresponding complexity estimate by at most the width D of that layer. Since $D \leq p_2$, we obtain the above.

As in Section 3.2, our goal is to obtain minimax rates in the same setting as our approximation theory, and in particular for input classes given by bounded Lipschitz spaces as in Assumption **S.1**. For the architecture (18), however, the relevant input space is a product of two such classes. The next result explains how to construct an infinite-dimensional cube in this product space by embedding a single cube into one factor.

Lemma 3.23 (Embedding a Single Cube into a Product Space)**.** *Let W and U be Banach spaces, and equip $W \times U$ with a norm $\|\cdot\|_{W\times U}$ satisfying Assumptions **N.1** and **N.3**. Let $K_W \subset W$ and suppose that K_W contains an infinite-dimensional cube $Q_\eta(A; \{e_j\}_{j\in\mathbb{N}})$ for some $\eta > 1$. Then the subset $K_W \times \{0\} \subset W \times U$ contains an infinite-dimensional cube with the same exponent η.*

Remark 3.24 (Optimality of infinite-dimensional product cube). Although the construction of the infinite-dimensional product cube in Lemma 3.23 is straightforward, it also raises a natural question of optimality. Namely, one may ask whether there exists a genuinely product-space cube construction that would lead to a better lower-bound exponent than the one obtained here from a single-factor embedding.

We now turn to the upper bound in the corresponding minimax approximation-complexity rates. In particular, the following result shows that the concatenated architecture is sufficiently expressive to approximate operator-learning problems posed on product spaces.

Proposition 3.25 (Scaling Laws for Multiple Operator Learning through Product Spaces)**.** *Let $d_W, d_U, d_V > 0$ be integers,*

$$\gamma_W, \gamma_U, \gamma_V, \beta_W, \beta_U, \beta_V, L_W, L_U, L_V, L_G, L_{\mathcal{G}} > 0 \qquad \text{and} \qquad r_G, r_{\mathcal{G}} \geq 1$$

*and assume that $W(d_W, \gamma_W, L_W, \beta_W)$, $U(d_U, \gamma_U, L_U, \beta_U)$ and $V(d_V, \gamma_V, L_V, \beta_V)$ satisfy Assumption **S.1**. Let G be a map satisfying Assumptions **O.1** and **O.2**. There exist constants*

- *C depending on γ_V, L_V*
- *C' depending on β_W, β_U, $L_{\mathcal{G}}$, γ_U, $r_{\mathcal{G}}$, L_G, γ_W, r_G*
- *C_W depending on $L_{\mathcal{G}}$, γ_U, $r_{\mathcal{G}}$, L_G, γ_W, r_G, L_W*
- *C_U depending on $L_{\mathcal{G}}$, γ_U, $r_{\mathcal{G}}$, L_G, γ_W, r_G, L_U*

such that the following holds. For $0 < \varepsilon$ sufficiently small,

- *let $N := 2C\sqrt{d_V}\varepsilon^{-1}$ and consider the network class $\mathcal{F}_1 := \mathcal{F}_{\mathrm{NN}}(d_V, 1, L_1, p_1, K_1, \kappa_1, R_1)$ whose parameters scale as*

$$\begin{aligned} &L_1 = \mathcal{O}\left(d_V^2 \log d_V + d_V^2(\log(\varepsilon^{-1}) + \log(2))\right), \qquad && p_1 = \mathcal{O}(1), \\ &K_1 = \mathcal{O}\left(d_V^2 \log d_V + d_V^2(\log(\varepsilon^{-1}) + \log(2))\right), \qquad && \kappa_1 = \mathcal{O}(d_V^{d_V/2+1}\varepsilon^{-(d_V+1)}2^{d_V+1}), \\ &R_1 = 1 \end{aligned}$$

  where the constants hidden in $\mathcal{O}$ depend on γ_V and L_V

- *let $\delta_W = C_W\varepsilon/2$ and let $\{a_i\}_{i=1}^{n_{c_W}} \subset \Omega_W$ be points so that $\{\mathcal{B}_{\delta_W}(a_i)\}_{i=1}^{n_{c_W}}$ is a cover of Ω_W for some n_{c_W};*
- *let $\delta_U = C_U\varepsilon/2$ and let $\{c_i\}_{i=1}^{n_{c_U}} \subset \Omega_U$ be points so that $\{\mathcal{B}_{\delta_U}(c_i)\}_{i=1}^{n_{c_U}}$ is a cover of Ω_U for some n_{c_U};*
- *let $H := 4C'\sqrt{n_{c_W} + n_{c_U}}\varepsilon^{-1}$ and consider the network class $\mathcal{F}_2 := \mathcal{F}_{\mathrm{NN}}(n_{c_W}+n_{c_U}, 1, L_2, p_2, K_2, \kappa_2, R_2)$ with parameters scaling as*

$$L_2 = \mathcal{O}\left((n_{c_W} + n_{c_U})^2 \log(n_{c_W} + n_{c_U}) + (n_{c_W} + n_{c_U})^2[\log(\varepsilon^{-1}) + 2\log(2)]\right), \qquad p_2 = \mathcal{O}(1),$$

$$K_2 = \mathcal{O}\left((n_{c_W}+n_{c_U})^2\log(n_{c_W}+n_{c_U}) + (n_{c_W}+n_{c_U})^2[\log(\varepsilon^{-1})+2\log(2)]\right),$$
$$\kappa_2 = \mathcal{O}((n_{c_W}+n_{c_U})^{(n_{c_W}+n_{c_U})/2+1}\varepsilon^{-(n_{c_W}+n_{c_U}+1)}2^{2(n_{c_W}+n_{c_U}+1)}), \qquad R_2 = 1$$

where the constants hidden in $\mathcal{O}$ depend on β_W, β_U, $L_{\mathcal{G}}$, γ_U, $r_{\mathcal{G}}$, L_G, γ_W, r_G.

Then, there exists

- *networks* $\{\tau_\ell\}_{k=1}^{N^{d_V}} \subset \mathcal{F}_1$ *and* $\{b_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{F}_2$
- *functions* $\{\alpha_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0)$ *and* $\{u_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{B}_{\beta_U,\|\cdot\|_{\mathrm{L}^\infty},\Omega_U}(0)$
- *points* $\{v_\ell\}_{\ell=1}^{N^{d_V}} \subset \Omega_V$

such that

$$\sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}\left|G[\alpha][u](x) - \sum_{\ell=1}^{N^{d_V}}\sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} G[\alpha_k][u_k](v_\ell)b_k(\boldsymbol{\alpha},\boldsymbol{u})\tau_\ell(x)\right| \le \varepsilon \tag{19}$$

and where $\boldsymbol{\alpha} = \frac{\max\{\beta_W,\beta_U\}}{\beta_W}(\alpha(a_1),\alpha(a_2),\dots,\alpha(a_{n_{c_W}}))^\top$, $\boldsymbol{u} = \frac{\max\{\beta_W,\beta_U\}}{\beta_U}(u(c_1),u(c_2),\dots,u(c_{n_{c_U}}))^\top$.

Remark 3.26 (Total Number of Parameters for Multiple Operator Learning through Product Spaces). Let NN_ε be a network that satisfies (19). We want to estimate $\|\Theta\|_0 + K_1N^{d_V} + K_2H^{n_{c_W}+n_{c_U}} \ge N_\#(\mathrm{NN}_\varepsilon)$ where $\Theta_0 = \{\theta_{k\ell}\}_{k=1,\ell=1}^{H^{n_{c_W}+n_{c_U}},N^{d_V}}$.

First, by [31, Lemma 2], we have that

$$n_{c_W} \lesssim \delta_W^{-d_W} \lesssim \varepsilon^{-d_W} \qquad \text{and} \qquad n_{c_U} \lesssim \delta_U^{-d_U} \lesssim \varepsilon^{-d_U},$$

implying that $n_{c_W}+n_{c_U} \lesssim \varepsilon^{-\max\{d_W,d_U\}}$. Next, we consider

$$K_1N^{d_V} \lesssim \log(\varepsilon^{-1})\varepsilon^{-d_V}, \tag{20}$$

$$\begin{aligned} K_2H^{n_{c_W}+n_{c_U}} &\lesssim \left(\varepsilon^{-2\max\{d_W,d_U\}}\log(\varepsilon^{-1})\right)\left(\varepsilon^{-\max\{d_W,d_U\}/2+1}\right)^{\varepsilon^{-\max\{d_W,d_U\}}} \\ &\lesssim \left(\varepsilon^{-2\max\{d_W,d_U\}}\log(\varepsilon^{-1})\right)\varepsilon^{-(\max\{d_W,d_U\}/2-1)\varepsilon^{-\max\{d_W,d_U\}}} \\ &\lesssim \varepsilon^{-d\varepsilon^{-\max\{d_W,d_U\}}} \end{aligned} \tag{21}$$

for some $d>0$ and therefore

$$\|\Theta\|_0 \le N^{d_V}H^{n_{c_W}+n_{c_U}} \lesssim \varepsilon^{-d\varepsilon^{-\max\{d_W,d_U\}}}. \tag{22}$$

Combining (20), (21) and (22), we obtain:

$$\|\Theta\|_0 + K_1N^{d_V} + K_2H^{n_{c_W}+n_{c_U}} \lesssim \varepsilon^{-\gamma\varepsilon^{-\max\{d_W,d_U\}}}.$$

Repeating the computation in [47, Remark 3.14], we can also deduce that the approximation error ε scales as follows:

$$\varepsilon \lesssim \left(\frac{\log N_\#}{\log\log N_\#}\right)^{-1/\max\{d_W,d_U\}}$$

Remark 3.27 (Uniform Multiple Operator Approximation in Product Spaces). Similarly to Remark C.2, one can show that Proposition C.1 admits a uniform version for families $\{G_j : W\times U\to V\}_{j\in J}$ of multiple operator maps, even when the family is uncountable. Indeed, provided that the quantities controlling the construction (in particular the Lipschitz constants, the L^∞-bounds) are bounded uniformly over the family, the proof yields a single collection of sampling points and a single family of neural networks that work simultaneously for all maps in the family. The dependence on the particular target map then appears only through the coefficients

of the resulting approximation. In particular, with the same ε-dependent classes of networks as in Proposition 3.25, one obtains

$$\sup_{j\in J}\sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}\left|G_j[\alpha][u](x)-\sum_{k=1}^{H^{n_{c_u}+n_{c_W}}}\sum_{\ell=1}^{N^{d_V}}G_j[\alpha_k][u_k](v_\ell)b_k(\boldsymbol{\alpha},\boldsymbol{u})\tau_\ell(x)\right|\leq\varepsilon$$

This conclusion always applies for finitely many multiple operator maps. We also note that, in the special case of families $\{G_j : U \to V\}_{j\in J}$, the present statement reduces to [47, Remark 3.9].

Remark 3.28 (Generalization Bounds for the Concatenated DeepONet Architecture). Similarly to the derivation of Theorem 3.4, the approximation-complexity bounds of Proposition 3.25 and Remark 3.26 can be combined with the statistical-learning framework of [31] to obtain generalization rates for the concatenated DeepONet architecture.

Indeed, in this setting the natural training data take the form

$$\{(\alpha_j, u_j, \{x_i\}_{i=1}^{n_x})\}_{j=1}^{n},$$

that is, one observes n samples of pairs (α_j, u_j), together with evaluations of the corresponding output at sampling points $\{x_i\}_{i=1}^{n_x}$. Repeating the proof of [31, Theorem 2] for the corresponding clipped hypothesis class yields a generalization bound analogous to (9), expressed in terms of its covering numbers. Using Remark 3.26, one obtains the corresponding complexity estimate, and the approximation and estimation terms can then be balanced by choosing $\varepsilon = \varepsilon(n)$ according to the approximation complexity. This leads to a generalization error of the form

$$\mathcal{O}\left(\left(\frac{\log n}{\log\log n}\right)^{-2/\max\{d_W,d_U\}}\right).$$

Combining Lemmata 3.21 and 3.23, Proposition 3.25 and [26, Theorem 2.11], we obtain the following.

Theorem 3.29 (Minimax Bounds for Lipschitz Multiple Operator Maps using the Concatenated DeepOnet Architecture)**.** *Let $d_W, d_U, d_V > 0$ be integers,*

$$\gamma_W,\gamma_U,\gamma_V,\beta_W,\beta_U,\beta_V,L_W,L_U,L_V>0 \quad \textit{and} \quad r_G\geq 1$$

*and assume that $W(d_W,\gamma_W,L_W,\beta_W)$, $U(d_U,\gamma_U,L_U,\beta_U)$ and $V(d_V,\gamma_V,L_V,\beta_V)$ satisfy Assumption **S.1**. We equip the product Banach space $\mathrm{L}^{r_G}(\Omega_W)\times\mathrm{L}^{r_G}(\Omega_U)$ with a norm $\|\cdot\|_{\mathrm{L}^{r_G}(\Omega_W)\times\mathrm{L}^{r_G}(\Omega_U)}$ that satisfies Assumptions **N.1**, **N.2** and **N.3**.*

1. *For any $\eta > \min\left\{1+\frac{1}{d_W}, 1+\frac{1}{d_U}\right\}$, $r\in\mathbb{N}$ and $\delta>0$, there exists an r-times Frechet differentiable map $G : \mathrm{L}^{r_G}(\Omega_W)\times\mathrm{L}^{r_G}(\Omega_U)\to V$ and $\overline{\varepsilon} := \overline{\varepsilon}(\eta,\delta,r) > 0$ such that the following holds: for any $\varepsilon\leq\overline{\varepsilon}$ and NN_ε of the form* (18) *satisfying*

   $$\sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}|\,\mathrm{NN}_\varepsilon[\alpha][u](x)-G[\alpha][u](x)|\leq\varepsilon, \tag{23}$$

   we have

   $$\|\Theta\|_0+HK_2+NK_1\gtrsim\exp\left(c\varepsilon^{-1/[(\eta+1+\delta)r]}\right)$$

   for some $c := c(\eta,\delta,r) > 0$ and where $\Theta = \{\theta_{k\ell}\}$.

2. *With $r = 1$, the map G in part 1. can be chosen so that it satisfies Assumptions **O.1** and **O.2**.*

3. *Let $\mathcal{H}$ denote the class of all maps G satisfying Assumptions **O.1** and **O.2**. Define the worst-case/minimax approximation complexity*

   $$\mathfrak{C}(\varepsilon;\mathcal{H}) := \inf\Big\{M\in\mathbb{N}\,\Big|\,\forall G\in\mathcal{H},\ \exists\,\mathrm{NN}_\varepsilon \textit{ of the form } (18) \textit{ satisfying } (23) \textit{ and } \|\Theta\|_0+HK_2+NK_1\leq M\Big\}.$$

Let $r = 1$, $\eta > \min\left\{1 + \frac{1}{d_W}, 1 + \frac{1}{d_U}\right\}$, *and* $\delta > 0$. *Then, for all* $0 < \varepsilon \leq \overline{\varepsilon}(\eta, \delta, 1)$ *sufficiently small,*

$$\exp\left(c\,\varepsilon^{-1/[(\eta+1+\delta)]}\right) \lesssim \mathfrak{C}(\varepsilon; \mathcal{H}) \lesssim \exp\left(d\log(\varepsilon^{-1})\varepsilon^{-\max\{d_W, d_U\}}\right)$$

for some $d > 0$ *only depending on* d_W *and* d_U.

The minimax rates in Theorems 3.19 and 3.29, as well as the generalization rates in Theorem 3.4 and Remark 3.28, show that MNO and concatenated DeepONet share essentially the same scaling laws. Consequently, the present analysis does not distinguish the two architectures from the viewpoint of approximation complexity or statistical generalization.

This shifts the emphasis to empirical performance. In [47, Section 5], MNO was observed to outperform concatenated DeepONet by orders of magnitude on parametric PDE tasks. This likely reflects the fact that MNO explicitly separates the roles of the parameter variable α and the input function u, a distinction that is fundamental in many multiple operator learning problems [46]. Consequently, the absence of a clear minimax separation provides an additional argument for designing architectures specifically adapted to the multiple operator setting.

4 Conclusion

We studied approximation and statistical generalization in multiple operator learning, with a particular focus on the general separable architecture, of which MNO is a special case. While prior bounds exhibited an additional exponential blow-up relative to standard operator learning, our results showed that multiple operator learning obeys a qualitatively similar scaling to single task operator learning. In particular, our first main result showed that, by refining the approximation argument, MNO has near-optimal approximation rates and that the corresponding approximation complexity can be reduced to the same overall scale as in operator learning. Through the generalization framework of [46], this also yields improved statistical learning rates, again matching the operator-learning scale. Our second main result establishes a lower-bound theory for multiple operator learning. Extending the framework of [26], we proved a curse of parametric complexity for broad classes of Lipschitz and differentiable multiple operator maps, and then specialized this abstract result to the bounded Lipschitz classes considered in our approximation theory. In this way, we obtained minimax approximation-complexity bounds showing that, although the previous constructive blow-up is not intrinsic, a genuine exponential complexity barrier remains in the worst case.

Theoretically, we compared MNO with a concatenated DeepONet-type extension to multi-task learning. From the viewpoint of minimax approximation complexity, both architectures have essentially the same scaling laws on the broad Lipschitz classes studied here. This indicates that worst-case complexity alone does not explain the empirical advantage of MNO observed in [47], and suggests that the practical gains of multiple-operator-specific architectures may lie in more refined structural, geometric, or data-dependent properties that are not captured by minimax rates on generic Lipschitz classes.

Several directions remain open. On the upper-bound side, it would be natural to investigate whether stronger assumptions on the target multiple operator maps, such as holomorphic structure or PDE-specific regularity, can lead to substantially better rates. On the lower-bound side, it would be interesting to obtain sharper minimax characterizations, in particular at the level of the exponent. More broadly, one would like to establish minimax approximation-complexity rates for other classes of multiple operator maps beyond the Lipschitz/differentiable setting considered here, with the longer-term goal of identifying and characterizing a meaningful class of *well-approximable* multiple operator maps.

Acknowledgment

This work was supported by NSF 2427558.

References

[1] Ben Adcock, Michael Griebel, and Gregor Maier. The sample complexity of learning lipschitz operators with respect to gaussian measures, 2025.

[2] Aras Bacho, Aleksei G. Sorokin, Xianjin Yang, Théo Bourdais, Edoardo Calvello, Matthieu Darcy, Alexander Hsu, Bamdad Hosseini, and Houman Owhadi. Operator learning at machine precision, 2025.

[3] Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model Reduction And Neural Networks For Parametric PDEs. *The SMAI Journal of computational mathematics*, 7:121–157, 2021.

[4] Yadi Cao, Yuxuan Liu, Liu Yang, Rose Yu, Hayden Schaeffer, and Stanley Osher. Vicon: Vision in-context operator networks for multi-physics fluid dynamics prediction. *arXiv preprint arXiv:2411.16063*, 2024.

[5] Javier Castro. The kolmogorov infinite dimensional equation in a hilbert space via deep learning methods. *Journal of Mathematical Analysis and Applications*, 527(2):127413, 2023.

[6] Javier Castro, Claudio Muñoz, and Nicolás Valenzuela. The calderón's problem via deeponets. *Vietnam Journal of Mathematics*, 52(3):775–806, 2024.

[7] T. Chen and H. Chen. Approximations of continuous functionals by neural networks with application to dynamic systems. *IEEE Transactions on Neural Networks*, 4(6):910–918, 1993.

[8] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

[9] Stephan Dahlke, Filippo De Mari, Philipp Grohs, and Demetrio Labate, editors. *Harmonic and Applied Analysis*. Applied and Numerical Harmonic Analysis. Birkhäuser, Cham, 2015.

[10] Maarten V. de Hoop, Daniel Zhengyu Huang, Elizabeth Qian, and Andrew M. Stuart. The cost-accuracy trade-off in operator learning with neural networks, 2022.

[11] D. L. Donoho. Sparse components of images and optimal atomic decompositions. *Constructive Approximation*, 17(3):353–382, 2001.

[12] Takashi Furuya, Michael Anthony Puthawala, Matti Lassas, and Maarten V. de Hoop. Globally injective and bijective neural operators. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[13] Philipp Grohs, Samuel Lanthaler, and Margaret Trautner. Theory-to-practice gap for neural networks and neural operators, 2025.

[14] Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[15] Lukas Herrmann, Christoph Schwab, and Jakob Zech. Neural and spectral operator surrogates: unified construction and expression rate bounds. *Advances in Computational Mathematics*, 50(4):72, 2024.

[16] Daniel Zhengyu Huang, Nicholas H. Nelsen, and Margaret Trautner. An operator learning perspective on parameter-to-observable maps. *Foundations of Data Science*, 7(1):163–225, 2025.

[17] Pengzhan Jin, Shuai Meng, and Lu Lu. Mionet: Learning multiple-input operators via tensor product. *SIAM Journal on Scientific Computing*, 44(6):A3490–A3514, 2022.

[18] Derek Jollie, Jingmin Sun, Zecheng Zhang, and Hayden Schaeffer. Time-series forecasting and refinement within a multimodal pde foundation model. *Journal of Machine Learning for Modeling and Computing*, 6(2):77–89, 2025.

[19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[20] Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for fourier neural operators. *J. Mach. Learn. Res.*, 22(1), January 2021.

[21] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: learning maps between function spaces with applications to pdes. *J. Mach. Learn. Res.*, 24(1), January 2023.

[22] Nikola B. Kovachki, Samuel Lanthaler, and Hrushikesh Mhaskar. Data complexity estimates for operator learning, 2024.

[23] Nikola B. Kovachki, Samuel Lanthaler, and Andrew M. Stuart. Chapter 9 - operator learning: Algorithms and analysis. In Siddhartha Mishra and Alex Townsend, editors, *Numerical Analysis Meets Machine Learning*, volume 25 of *Handbook of Numerical Analysis*, pages 419–467. Elsevier, 2024.

[24] Samuel Lanthaler. Operator learning with pca-net: upper and lower complexity bounds. *J. Mach. Learn. Res.*, 24(1), January 2023.

[25] Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error estimates for deeponets: a deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 03 2022.

[26] Samuel Lanthaler and Andrew M Stuart. The parametric complexity of operator learning. *IMA Journal of Numerical Analysis*, page draf028, 08 2025.

[27] Jose Antonio Lara Benitez, Takashi Furuya, Florian Faucher, Anastasis Kratsios, Xavier Tricoche, and Maarten V. de Hoop. Out-of-distributional risk bounds for neural operators with applications to the helmholtz equation. *J. Comput. Phys.*, 513(C), September 2024.

[28] Hao Liu, Jiahui Cheng, and Wenjing Liao. Deep neural networks are adaptive to function regularity and data distribution in approximation and estimation. *Journal of Machine Learning Research*, 26(213):1–56, 2025.

[29] Hao Liu, Biraj Dahal, Rongjie Lai, and Wenjing Liao. Generalization error guaranteed auto-encoder-based nonlinear model reduction for operator learning. *Applied and Computational Harmonic Analysis*, 74:101717, 2025.

[30] Hao Liu, Haizhao Yang, Minshuo Chen, Tuo Zhao, and Wenjing Liao. Deep nonparametric estimation of operators between infinite dimensional spaces. *J. Mach. Learn. Res.*, 25(1), January 2024.

[31] Hao Liu, Zecheng Zhang, Wenjing Liao, and Hayden Schaeffer. Neural scaling laws of deep relu and deep operator network: A theoretical study, 2024.

[32] Yuxuan Liu, Jingmin Sun, Xinjie He, Griffin Pinney, Zecheng Zhang, and Hayden Schaeffer. PROSE-FD: A multimodal PDE foundation model for learning multiple operators for forecasting fluid dynamics. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*, 2024.

[33] Yuxuan Liu, Jingmin Sun, and Hayden Schaeffer. Bcat: A block causal transformer for pde foundation models for fluid dynamics. *arXiv preprint arXiv:2501.18972*, 2025.

[34] Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Prose: Predicting multiple operators and symbolic expressions using multimodal transformers. *Neural Networks*, 180:106707, 2024.

[35] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

[36] Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022.

[37] Carlo Marcati and Christoph Schwab. Exponential convergence of deep operator networks for elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 61(3):1513–1545, 2023.

[38] Carlo Marcati and Christoph Schwab. Expression rates of neural operators for linear elliptic pdes in polytopes. *CoRR*, abs/2409.17552, 2024.

[39] Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Géraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for physical surrogate models. *arXiv preprint arXiv:2310.02994*, 2023.

[40] Elisa Negrini, Yuxuan Liu, Liu Yang, Stanley J Osher, and Hayden Schaeffer. A multimodal pde foundation model for prediction and scientific text descriptions. *arXiv preprint arXiv:2502.06026*, 2025.

[41] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018.

[42] Christoph Schwab, Andreas Stein, and Jakob Zech. Deep operator network approximation rates for lipschitz operators. *Analysis and Applications*, 24(01):199–239, 2026.

[43] Jingmin Sun, Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Towards a foundation model for partial differential equations: Multioperator learning and extrapolation. *Physical Review E*, 111(3):035304, 2025.

[44] Jingmin Sun, Zecheng Zhang, and Hayden Schaeffer. Lemon: Learning to learn multi-operator networks, 2025.

[45] Zhuoyuan Wang, Hanjiang Hu, Xiyu Deng, Saviz Mowlavi, and Yorie Nakahira. Opinf-llm: Parametric pde solving with llms via operator inference, 2026.

[46] Adrien Weihs and Hayden Schaeffer. Generalization bounds and statistical guarantees for multi-task and multiple operator learning with mno networks, 2026.

[47] Adrien Weihs, Jingmin Sun, Zecheng Zhang, and Hayden Schaeffer. A deep learning framework for multi-operator learning: Architectures and approximation theory, 2025.

[48] Liu Yang, Siting Liu, Tingwei Meng, and Stanley J Osher. In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39):e2310142120, 2023.

[49] Liu Yang, Tingwei Meng, Siting Liu, and Stanley J Osher. Prompting in-context operator learning with sensor data, equations, and natural language. *arXiv preprint arXiv:2308.05061*, 2023.

[50] Zhanhong Ye, Zining Liu, Bingyang Wu, Hongjie Jiang, Leheng Chen, Minyan Zhang, Xiang Huang, Qinghe Meng Zou, Hongsheng Liu, and Bin Dong. Pdeformer-2: A versatile foundation model for two-dimensional partial differential equations. *arXiv preprint arXiv:2507.15409*, 2025.

[51] Benjamin J Zhang, Siting Liu, Stanley J Osher, and Markos A Katsoulakis. Probabilistic operator learning: generative modeling and uncertainty quantification for foundation models of differential equations. *arXiv preprint arXiv:2509.05186*, 2025.

[52] Zecheng Zhang. Modno: Multi-operator learning with distributed neural operators. *Computer Methods in Applied Mechanics and Engineering*, 431:117229, 2024.

[53] Zecheng Zhang, Wing Tat Leung, and Hayden Schaeffer. A discretization-invariant extension and analysis of some deep operator networks. *Journal of Computational and Applied Mathematics*, 456:116226, 2025.

[54] Zecheng Zhang, Hao Liu, Wenjing Liao, and Guang Lin. Coefficient-to-basis network: a fine-tunable operator learning framework for inverse problems with adaptive discretizations and theoretical guarantees. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 383(2305):20240054, 09 2025.

[55] Zecheng Zhang, Christian Moya, Lu Lu, Guang Lin, and Hayden Schaeffer. D2no: Efficient handling of heterogeneous input function spaces with distributed deep neural operators. *Computer Methods in Applied Mechanics and Engineering*, 428:117084, 2024.

[56] Zecheng Zhang, Leung Wing Tat, and Hayden Schaeffer. Belnet: basis enhanced learning, a mesh-free neural operator. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 479(2276):20230043, 2023.

Appendix

In this section, we present detailed proofs of all our results.

A Near-Optimal Approximation Rates

Proof of Theorem 3.1. For $u \in U$ and $x \in \Omega_V$, we define the functional $f_{u,x} : \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0) \mapsto \mathbb{R}$ as $f_{u,x}(\alpha) = G[\alpha][u](x)$. In particular, $|f_{u,x}| \le \beta_V$ and $f_{u,x}$ is Lipschitz in $\mathrm{L}^\infty(\Omega_W)$: specifically, we have

$$\begin{aligned} |f_{u,x}(\alpha_1) - f_{u,x}(\alpha_2)| &= |G[\alpha_1][u](x) - G[\alpha_2][u](x)| \\ &\le L_G \|\alpha_1 - \alpha_2\|_{\mathrm{L}^{r_G}(\Omega_W)} \\ &\le L_G |\Omega_W|^{1/(r_G)} \|\alpha_1 - \alpha_2\|_{\mathrm{L}^\infty(\Omega_U)}. \end{aligned} \tag{24}$$

where we use Assumption **O.2** for (24). Next, we want to approximate the entire family of functionals $\{f_{u,x} : \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0) \mapsto \mathbb{R}\}_{u\in U,\, x\in\Omega_V}$ by neural networks. Since the family is both uniformly Lipschitz and bounded by β_V, we can apply Proposition C.1 and Remark C.2 with $W \times \{\emptyset\} \cong W$ (or directly [47, Theorem 3.6 and Remark 3.7]). Specifically, there exist constants

- C'' depending on $\beta_W, L_G|\Omega_W|^{1/r_G}$
- C_ζ depending on $L_G|\Omega_W|^{1/r_G}, L_W$

such that the following holds. For any $\varepsilon_0 > 0$,

- let $\zeta := C_\zeta \varepsilon_0$ and let $\{y_m\}_{m=1}^{n_{c_W}} \subset \Omega_W$ be points so that $\{\mathcal{B}_\zeta(y_m)\}_{m=1}^{n_{c_W}}$ is a cover of Ω_W for some n_{c_W}
- let $P = C''\sqrt{n_{c_W}}\varepsilon_0^{-1}$ and consider the network class $\mathcal{F}_3 = \mathcal{F}_{\mathrm{NN}}(n_{c_W}, 1, L_3, p_3, K_3, \kappa_3, R_3)$ whose parameters scale as

$$\begin{aligned} &L_3 = \mathcal{O}\left(n_{c_W}^2 \log(n_{c_W}) + n_{c_W}^2 \log(\varepsilon_0^{-1}) + n_{c_W}^2 \log(2)\right), \quad p_3 = \mathcal{O}(1), \\ &K_3 = \mathcal{O}\left(n_{c_W}^2 \log n_{c_W} + n_{c_W}^2 \log(\varepsilon_0^{-1}) + n_{c_W}^2 \log(2)\right), \\ &\kappa_3 = \mathcal{O}(n_{c_W}^{n_{c_W}/2+1} 2^{n_{c_W}+1} \varepsilon_0^{-n_{c_W}-1}), \qquad R_3 = 1 \end{aligned}$$

  where the constants hidden in $\mathcal{O}$ depend on β_W and $L_G|\Omega_W|^{1/r_G}$.

Then, there exists networks $\{l_p\}_{p=1}^{P^{n_{c_W}}} \subset \mathcal{F}_3$ and functions $\{\alpha_p\}_{p=1}^{P^{n_{c_W}}} \subset \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0)$ such that

$$\sup_{\alpha\in W} \left| f_{u,x}(\alpha) - \sum_{p=1}^{P^{n_{c_W}}} f_{u,x}(\alpha_p) l_p(\boldsymbol{\alpha}) \right| = \sup_{\alpha\in W} \left| G[\alpha][u](x) - \sum_{p=1}^{P^{n_{c_W}}} G[\alpha_p][u](x) l_p(\boldsymbol{\alpha}) \right| \le \varepsilon_0 \tag{25}$$

where $\boldsymbol{\alpha} = (\alpha(y_1), \dots, \alpha(y_{n_{c_W}}))^\top$. We also have $0 \le l_p \le 1$ for $1 \le p \le P^{n_{c_W}}$.

By Assumption **O.1**, $G[\alpha_p] \in \mathcal{G}$ for all $1 \le p \le P^{n_{c_W}}$. In particular, this is a finite family of operators, and by applying Proposition 3.25 and Remark 3.27 with $W \times \{\emptyset\} \cong W$, there exist constants

- C depending on γ_V, L_V
- C' depending on β_U, $L_{\mathcal{G}}$, γ_U, $r_{\mathcal{G}}$,
- C_U depending on $L_{\mathcal{G}}$, γ_U, $r_{\mathcal{G}}$, L_U

such that the following holds. For any $0 < \varepsilon_1$ sufficiently small,

- let $N := 2C\sqrt{d_V}\varepsilon_1^{-1}$ and consider the network class $\mathcal{F}_1 := \mathcal{F}_{\mathrm{NN}}(d_V, 1, L_1, p_1, K_1, \kappa_1, R_1)$ whose parameters scale as

$$\begin{aligned} &L_1 = \mathcal{O}\left(d_V^2 \log d_V + d_V^2(\log(\varepsilon_1^{-1}) + \log(2))\right), \qquad && p_1 = \mathcal{O}(1), \\ &K_1 = \mathcal{O}\left(d_V^2 \log d_V + d_V^2(\log(\varepsilon_1^{-1}) + \log(2))\right), \qquad && \kappa_1 = \mathcal{O}(d_V^{d_V/2+1}\varepsilon_1^{-(d_V+1)}2^{d_V+1}), \\ &R_1 = 1 \end{aligned}$$

  where the constants hidden in $\mathcal{O}$ depend on γ_V and L_V;

- let $\delta_U = C_U\varepsilon_1$ and let $\{c_i\}_{i=1}^{n_{c_U}} \subset \Omega_U$ be points so that $\{\mathcal{B}_{\delta_U}(c_i)\}_{i=1}^{n_{c_U}}$ is a cover of Ω_U for some n_{c_U};

- let $H := 4C'\sqrt{n_{c_U}}\varepsilon_1^{-1}$ and consider the network class $\mathcal{F}_2 := \mathcal{F}_{\mathrm{NN}}(n_{c_U}, 1, L_2, p_2, K_2, \kappa_2, R_2)$ with parameters scaling as

$$\begin{aligned} &L_2 = \mathcal{O}\left(n_{c_U}^2 \log(n_{c_U}) + (n_{c_U}^2[\log(\varepsilon_1^{-1}) + 2\log(2)]\right), \qquad && p_2 = \mathcal{O}(1), \\ &K_2 = \mathcal{O}\left(n_{c_U}^2 \log(n_{c_U}) + n_{c_U}^2[\log(\varepsilon_1^{-1}) + 2\log(2)]\right), \\ &\kappa_2 = \mathcal{O}(n_{c_U}^{n_{c_U}/2+1}\varepsilon_1^{-(n_{c_U}+1)}2^{2(n_{c_U}+1)}), \qquad && R_2 = 1 \end{aligned}$$

  where the constants hidden in $\mathcal{O}$ depend on β_U, $L_{\mathcal{G}}$, γ_U, $r_{\mathcal{G}}$.

Then, there exists

- networks $\{\tau_\ell\}_{k=1}^{N^{d_V}} \subset \mathcal{F}_1$ and $\{b_k\}_{k=1}^{H^{n_{c_U}}} \subset \mathcal{F}_2$
- functions $\{u_k\}_{k=1}^{H^{n_{c_U}}} \subset \mathcal{B}_{\beta_U, \|\cdot\|_{\mathrm{L}^\infty}, \Omega_U}(0)$
- points $\{v_\ell\}_{\ell=1}^{N^{d_V}} \subset \Omega_V$

such that

$$\sup_{u\in U}\sup_{x\in\Omega_V}\left|G[\alpha_p][u](x) - \sum_{\ell=1}^{N^{d_V}}\sum_{k=1}^{H^{n_{c_U}}} G[\alpha_p][u_k](v_\ell)b_k(\boldsymbol{u})\tau_\ell(x)\right| \le \varepsilon_1 \tag{26}$$

where $\boldsymbol{u} = (u(c_1), u(c_2), \dots, u(c_{n_{c_U}}))^\top$.

We continue by estimating as follows:

$$\begin{aligned} &\sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}\left|G[\alpha][u](x) - \sum_{p=1}^{P^{n_{c_W}}}\sum_{k=1}^{H^{n_{c_U}}}\sum_{\ell=1}^{N^{d_V}} G[\alpha_p][u_k](v_\ell)l_p(\boldsymbol{\alpha})b_k(\boldsymbol{u})\tau_\ell(x)\right| \\ &\le \sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}\left|G[\alpha][u](x) - \sum_{p=1}^{P^{n_{c_W}}} G[\alpha_p][u](x)l_p(\boldsymbol{\alpha})\right| \\ &+ \sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}\left|\sum_{p=1}^{P^{n_{c_W}}} l_p(\boldsymbol{\alpha})\left[G[\alpha_p][u](x) - \sum_{k=1}^{H^{n_{c_U}}}\sum_{\ell=1}^{N^{d_V}} G[\alpha_p][u_k](v_\ell)b_k(\boldsymbol{u})\tau_\ell(x)\right]\right| \end{aligned}$$

$$\leq \varepsilon_0 + \sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}\sum_{p=1}^{P^{n_{c_W}}} l_p(\boldsymbol{\alpha})\left|G[\alpha_p][u](x) - \sum_{k=1}^{H^{n_{c_U}}}\sum_{\ell=1}^{N^{d_V}} G[\alpha_p][u_k](v_\ell)b_k(\boldsymbol{u})\tau_\ell(x)\right| \tag{27}$$

$$\leq \varepsilon_0 + \varepsilon_1 \sum_{p=1}^{P^{n_{c_W}}} l_p(\boldsymbol{\alpha}) \tag{28}$$

where we used (25) and the fact that $0 \leq l_p \leq 1$ for (27) as well as (26) for (28).

For the last term, we note that for any $0 < \eta < \beta_W$, the functional $f_\eta : \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0) \mapsto \mathbb{R}$ defined as $f(\alpha) = \eta$ has Lipschitz constant smaller that $L_G|\Omega_W|^{1/r_G}$ and is bounded by β_V. In particular, this means that $f_\eta \cup \{f_{u,x}\}_{u\in U,\, x\in\Omega_V}$ can be jointly approximated: specifically, (25) also applies to f_η. We therefore have

$$\varepsilon_0 \geq \sup_{\alpha\in W}\left|f_\eta(\alpha) - \sum_{p=1}^{P^{n_{c_W}}} f_\eta(\alpha_p)l_p(\boldsymbol{\alpha})\right| = \eta \sup_{\alpha\in W}\left|1 - \sum_{p=1}^{P^{n_{c_W}}} l_p(\boldsymbol{\alpha})\right|$$

which implies that

$$\sum_{p=1}^{P^{n_{c_W}}} l_p(\boldsymbol{\alpha}) \leq 1 + \frac{\varepsilon_0}{\eta} \tag{29}$$

for all $\alpha \in W$. Setting $\varepsilon_0 = \frac{\varepsilon}{2}$, $\varepsilon_1 = \frac{\varepsilon}{2(1+\frac{\varepsilon}{2\eta})}$ and inserting (29) into (28) yields:

$$\sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}\left|G[\alpha][u](x) - \sum_{p=1}^{P^{n_{c_W}}}\sum_{k=1}^{H^{n_{c_U}}}\sum_{\ell=1}^{N^{d_V}} G[\alpha_p][u_k](v_\ell)l_p(\boldsymbol{\alpha})b_k(\boldsymbol{u})\tau_\ell(x)\right| \leq \varepsilon.$$

The final network scalings for $\mathcal{F}_3$ are:

$$\begin{aligned}
&L_3 = \mathcal{O}\left(n_{c_W}^2\log(n_{c_W}) + n_{c_W}^2(\log(\varepsilon^{-1}) + 2\log(2))\right), \quad p_3 = \mathcal{O}(1),\\
&K_3 = \mathcal{O}\left(n_{c_W}^2\log(n_{c_W}) + n_{c_W}^2(\log(\varepsilon^{-1}) + 2\log(2))\right),\\
&\kappa_3 = \mathcal{O}(n_{c_W}^{n_{c_W}/2+1}2^{2(n_{c_W}+1)}\varepsilon^{-n_{c_W}-1}), \qquad R_3 = 1, \qquad P = 2C''\sqrt{n_{c_W}}\varepsilon^{-1}.
\end{aligned}$$

Noting that $\varepsilon_1 = \frac{\varepsilon}{2(1+\frac{\varepsilon}{2\eta})} = \frac{\varepsilon\eta}{2\eta+\varepsilon} \asymp \frac{\varepsilon}{2}$ for sufficiently small ε, the final network scalings for $\mathcal{F}_2$ are:

$$\begin{aligned}
&L_2 = \mathcal{O}\left(n_{c_U}^2\log(n_{c_U}) + (n_{c_U}^2[\log(\varepsilon^{-1}) + 3\log(2)]\right), \qquad p_2 = \mathcal{O}(1),\\
&K_2 = \mathcal{O}\left(n_{c_U}^2\log(n_{c_U}) + n_{c_U}^2[\log(\varepsilon^{-1}) + 3\log(2)]\right),\\
&\kappa_2 = \mathcal{O}(n_{c_U}^{n_{c_U}/2+1}\varepsilon^{-(n_{c_U}+1)}2^{3(n_{c_U}+1)}), \qquad R_2 = 1, \qquad H = 8C'\sqrt{n_{c_U}}\varepsilon^{-1}.
\end{aligned}$$

Similarly, the final scalings for $\mathcal{F}_1$ are:

$$\begin{aligned}
&L_1 = \mathcal{O}\left(d_V^2\log d_V + d_V^2(\log(\varepsilon^{-1}) + 2\log(2))\right), \qquad p_1 = \mathcal{O}(1),\\
&K_1 = \mathcal{O}\left(d_V^2\log d_V + d_V^2(\log(\varepsilon^{-1}) + 2\log(2))\right), \qquad \kappa_1 = \mathcal{O}(d_V^{d_V/2+1}\varepsilon^{-(d_V+1)}2^{2(d_V+1)}),\\
&R_1 = 1, \qquad N = 4C\sqrt{d_V}\varepsilon^{-1}.
\end{aligned}$$

□

Proof of Theorem 3.4. In the proof, $C > 0$ will denote a constant independent of ε, n_α, n_u, n_x and η that may change from line to line.

The proof of (9) follows the same argument as [46, Theorem 3.5]. Indeed, that proof does not rely on the specific architectural scalings of the network classes $\mathcal{F}_i$, but only on the fact that the chosen hypothesis class $\mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P^{n_{c_W}}, H^{n_{c_U}}, N^{d_V})$ contains an approximation of G with accuracy ε. In the present setting, this approximation property is provided by Theorem 3.1 and [46, Corollary 2.9].

It therefore only remains to prove the learning rate. For ease of notation, we write

$$\mathcal{N}(\eta) := \mathcal{N}\left(\eta, \mathrm{Cl}_a(I, \mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \{y_s\}, \{c_s\}, P^{n_{c_W}}, H^{n_{c_U}}, N^{d_V}), \|\cdot\|_{\mathrm{L}^\infty(W\times U\times\Omega_V)}\right).$$

We start from (9) and estimate as follows

$$\begin{aligned}
&\mathbb{E}_{S_{G,\{y_s\},\{c_s\}}}\mathbb{E}_{\alpha\sim\mu_\alpha}\,\mathbb{E}_{u\sim\mu_u}\,\mathbb{E}_{\{x_j\}_{j=1}^{n_x}\sim\mu_x^{\otimes n_x}}\left[\frac{1}{n_x}\sum_{j=1}^{n_x}\left(G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}[\boldsymbol{\alpha}][\mathbf{u}](x_j) - G[\alpha][u](x_j)\right)^2\right] \\
&\leq 4\varepsilon^2 + \eta(8\sigma+6) + \frac{8\sigma\eta}{\sqrt{n_\alpha n_u n_x}}\sqrt{\log\left(\mathcal{N}(\eta/(4\beta_V))\right)+\log(2)} + \frac{16\sigma^2}{n_\alpha n_u n_x}\left(\log\left(\mathcal{N}\left(\eta/(4\beta_V)\right)\right)+\log(2)\right) \\
&\quad + \frac{112\beta_V^2}{3n_\alpha}\log(\mathcal{N}(\eta/(4\beta_V))) \qquad (30) \\
&\lesssim 4\varepsilon^2 + \eta(8\sigma+6) + \frac{8\sigma\eta}{\sqrt{n_\alpha n_u n_x}}\sqrt{\log\left(\mathcal{N}(\eta/(4\beta_V))\right)} + \frac{16\sigma^2}{n_\alpha n_u n_x}\log\left(\mathcal{N}\left(\eta/(4\beta_V)\right)\right) \\
&\quad + \frac{112\beta_V^2}{3n_\alpha}\log(\mathcal{N}(\eta/(4\beta_V))) \qquad (31)
\end{aligned}$$

where we used the fact that $\mathcal{N}(\eta) \leq \mathcal{N}(\tilde{\eta})$ if $\tilde{\eta} \leq \eta$ for (30).

We next estimate the metric entropy as a function of ε. To this end, we recall [46, Equations 99 and 100], namely

$$\log(\mathcal{N}(\eta)) \lesssim P^{n_{c_W}} H^{n_{c_U}} N^{d_V}\left[\log\left(\frac{T}{\eta}\right) + K_3\log\left(\frac{L_3\kappa_3 T}{\eta}\right) + K_2\log\left(\frac{L_2\kappa_2 T}{\eta}\right) + K_1\log\left(\frac{L_1\kappa_1 T}{\eta}\right)\right] \qquad (32)$$

for some T satisfying

$$T \lesssim P^{n_{c_W}} H^{n_{c_U}} N^{d_V}\left[L_1\kappa_1^{L_1-1} + L_2\kappa_2^{L_2-1} + L_3\kappa_3^{L_3-1}\right] \qquad (33)$$

Using the scaling from Theorem 3.1 with $\varepsilon/2$ and recalling that $n_{c_W} \lesssim \delta_W^{-d_W} \lesssim \varepsilon^{-d_W}$ and $n_{c_U} \lesssim \delta_U^{-d_U} \lesssim \varepsilon^{-d_U}$ by [31, Lemma 2], we have

- $N^{d_V} \lesssim \varepsilon^{-d_V}$
- $H^{n_{c_U}} \lesssim \varepsilon^{-(1+d_U/2)\varepsilon^{-d_U}}$
- $P^{n_{c_W}} \lesssim \varepsilon^{-(1+d_W/2)\varepsilon^{-d_W}}$
- $\kappa_1^{L_1-1} \lesssim \varepsilon^{-(d_V+1)\log(\varepsilon^{-1})}$
- $\kappa_2 \lesssim \varepsilon^{-\varepsilon^{-d_U}(1+d_U/2)}$, $K_2 \asymp L_2 \lesssim \varepsilon^{-2d_U}\log(\varepsilon^{-1})$ and hence $\kappa_2^{L_2-1} \lesssim \varepsilon^{-\varepsilon^{-3d_U}(1+d_U/2)\log(\varepsilon^{-1})}$
- $\kappa_3 \lesssim \varepsilon^{-\varepsilon^{-d_W}(1+d_W/2)}$, $K_3 \asymp L_3 \lesssim \varepsilon^{-2d_W}\log(\varepsilon^{-1})$ and hence $\kappa_3^{L_3-1} \lesssim \varepsilon^{-\varepsilon^{-3d_W}(1+d_W/2)\log(\varepsilon^{-1})}$.

Let $d = \max\{d_U, d_W\}$. We estimate as follows, starting from (33):

$$\begin{aligned}
T &\lesssim P^{n_{c_W}} H^{n_{c_U}}\left[L_2\kappa_2^{L_2-1} + L_3\kappa_3^{L_3-1}\right] \\
&\lesssim \varepsilon^{-2(1+d/2)\varepsilon^{-d}}\varepsilon^{-\varepsilon^{-3d}(1+d/2)\log(\varepsilon^{-1})} \\
&\lesssim \varepsilon^{-\varepsilon^{-3d}(1+d/2)\log(\varepsilon^{-1})}.
\end{aligned}$$

Using the latter, we have

$$\begin{aligned}\log(L_2\kappa_2 T) &\lesssim \log\left(\varepsilon^{-\varepsilon^{-d_U}(1+d_U/2)-2d_U}\log(\varepsilon^{-1})\varepsilon^{-\varepsilon^{-3d}(1+d/2)\log(\varepsilon^{-1})}\right)\\ &\lesssim \log\left(\varepsilon^{-\varepsilon^{-3d}(1+d/2)\log(\varepsilon^{-1})}\log(\varepsilon^{-1})\right)\\ &= \varepsilon^{-3d}(1+d/2)\log(\varepsilon^{-1})+\log(\log(\varepsilon^{-1}))\\ &\lesssim \varepsilon^{-3d}(1+d/2)\log(\varepsilon^{-1})\end{aligned}$$

as well as (analogously)

$$\log(L_3\kappa_3 T) \lesssim \varepsilon^{-3d}(1+d/2)\log(\varepsilon^{-1}).$$

Then, continuing from (32):

$$\begin{aligned}\log(\mathcal{N}(\eta)) &\lesssim P^{n_{cW}}H^{n_{cU}}\left[\log(\eta^{-1})(K_2+K_3)+K_3\log\left(L_3\kappa_3 T\right)+K_2\log(L_2\kappa_2 T)\right]\\ &\lesssim \varepsilon^{-2(1+d/2)\varepsilon^{-d}}\Big[\varepsilon^{-2d}\log(\varepsilon^{-1})\log(\eta^{-1})\\ &\quad+\varepsilon^{-3d}(1+d/2)\log(\varepsilon^{-1})\left(\varepsilon^{-2d_U}\log(\varepsilon^{-1})+\varepsilon^{-2d_W}\log(\varepsilon^{-1})\right)\Big]\\ &\lesssim \varepsilon^{-2(1+d/2)\varepsilon^{-d}}\left(\log(\eta^{-1})+1\right). \qquad (34)\end{aligned}$$

Picking $\eta = 4\beta_V n_\alpha^{-1}$ to balance the η-dependent and the n_α^{-1}-terms, we insert (34) into (31) to obtain:

$$\begin{aligned}&\mathbb{E}_{S_{G,\{y_s\},\{c_s\}}}\mathbb{E}_{\alpha\sim\mu_\alpha}\mathbb{E}_{u\sim\mu_u}\mathbb{E}_{\{x_j\}_{j=1}^{n_x}\sim\mu_x^{\otimes n_x}}\left[\frac{1}{n_x}\sum_{j=1}^{n_x}\left(G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}[\boldsymbol{\alpha}][\mathbf{u}](x_j)-G[\alpha][u](x_j)\right)^2\right]\\ &\lesssim \varepsilon^2+\frac{C}{n_\alpha}+\frac{C}{n_\alpha^{3/2}\sqrt{n_u n_x}}\sqrt{\log\left(\mathcal{N}(n_\alpha^{-1})\right)}+\frac{C}{n_\alpha n_u n_x}\log\left(\mathcal{N}\left(n_\alpha^{-1}\right)\right)+\frac{C}{n_\alpha}\log(\mathcal{N}(n_\alpha^{-1})))\\ &\lesssim \varepsilon^2+\frac{C}{n_\alpha}\log(\mathcal{N}(n_\alpha^{-1})))\\ &\lesssim \varepsilon^2+\frac{\varepsilon^{-2(1+d/2)\varepsilon^{-d}}\log(n_\alpha)}{n_\alpha}\\ &=: T_1+T_2. \qquad (35)\end{aligned}$$

Finally, we pick $\varepsilon(n_\alpha)$ so that T_2 is at most of the same order as T_1. We fix

$$\varepsilon = \left(\frac{d}{4(1+d/2)}\frac{\log n_\alpha}{\log\log n_\alpha}\right)^{-1/d}$$

and compute

$$\log(\varepsilon^{-1}) = \frac{1}{d}\log\left(\frac{d}{4(1+d/2)}\frac{\log n_\alpha}{\log\log n_\alpha}\right) \lesssim \frac{1}{d}\left(\log\log n_\alpha - \log\log\log n_\alpha\right) \lesssim \frac{1}{d}\log\log n_\alpha.$$

From the latter,

$$\begin{aligned}\varepsilon^{-2(1+d/2)\varepsilon^{-d}} &= \exp\left(2(1+d/2)\varepsilon^{-d}\log(\varepsilon^{-1})\right)\\ &\lesssim \exp\left(2(1+d/2)\frac{d}{4(1+d/2)}\frac{\log n_\alpha}{\log\log n_\alpha}\frac{1}{d}\log\log n_\alpha\right)\\ &= n_\alpha^{1/2}\end{aligned}$$

and therefore $T_2 \lesssim \frac{\log n_\alpha}{n_\alpha^{1/2}}$ or equivalently $\log T_2 \lesssim \log\log n_\alpha - \frac{1}{2}\log n_\alpha$. For T_1, we have:

$$\log T_1 = \frac{-2}{d}\left[\log\left(\frac{d}{4(1+d/2)}\right)+\log\log n_\alpha - \log\log\log n_\alpha\right].$$

We note that $\lim_{n_\alpha\to\infty}\log T_2 = -\infty$ due to the $\frac{1}{2}\log n_\alpha$ term; similarly, $\lim_{n_\alpha\to\infty}\log T_1 = -\infty$ due to the $-\log\log n_\alpha$ term. This implies that T_2 goes to 0 much faster than T_1, so T_1 dominates the bound (35) and we conclude

$$\mathbb{E}_{S_{G,\{y_s\},\{c_s\}}}\mathbb{E}_{\alpha\sim\mu_\alpha}\,\mathbb{E}_{u\sim\mu_u}\,\mathbb{E}_{\{x_j\}_{j=1}^{n_x}\sim\mu_x^{\otimes n_x}}\left[\frac{1}{n_x}\sum_{j=1}^{n_x}\left(G_{a,I,\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3,S}[\boldsymbol{\alpha}][\mathbf{u}](x_j) - G[\alpha][u](x_j)\right)^2\right]$$
$$\lesssim \left(\frac{d}{4(1+d/2)}\frac{\log n_\alpha}{\log\log n_\alpha}\right)^{-2/\max\{d_W,d_U\}}.$$

□

B Approximation Complexity Lower Bounds and Minimax Rates for MNO

Proof of Theorem 3.8. Let $r \in \mathbb{N}$ and $\delta > 0$. We start by applying [26, Theorem 2.11]: there exists a r-times Frechet differentiable functional $F : W \to \mathbb{R}$ and $\varepsilon_0 := \varepsilon_0(\eta, \delta, r)$ such that for any $\varepsilon \le \varepsilon_0$ and functional of neural network type S_ε with

$$\sup_{\alpha\in K}|F(\alpha) - S_\varepsilon(\alpha)| \le \varepsilon,$$

we have

$$\mathcal{C}(S_\varepsilon) \ge \exp\left(c\varepsilon^{-1/[(\eta+1+\delta)r]}\right) \tag{36}$$

for some $c := c(\eta, \delta, r) > 0$.

Next, we choose a nontrivial element $\phi \in V$. Then there exists $x_0 \in \Omega_V$ such that $\phi(x_0) \neq 0$; rescaling ϕ, we may assume $\phi(x_0) = 1$. We define the constant operator

$$T : U \to V, \qquad T(u) = \phi$$

and its associated multiple operator map

$$G : W \to \{U \to V\}, \qquad G[\alpha] = F(\alpha)\,T.$$

Since F is r-times Frechet differentiable and T is independent of α, G is also r-times Frechet differentiable. Also, for any fixed $u_0 \in U$ and every $\alpha \in W$, we have

$$\mathrm{ev}_{x_0}\circ\mathrm{ev}_{u_0}\circ G[\alpha] = G[\alpha][u_0](x_0) = F(\alpha)\phi(x_0) = F(\alpha). \tag{37}$$

Let $0 < \varepsilon \le \overline{\varepsilon} := \varepsilon_0/\max\{1, C_V\}$ where $C_V > 0$ is such that

$$\|v\|_{C(\Omega_V)} \le C_V\|v\|_V \tag{38}$$

for all $v \in V$ due to the fact that $V \hookrightarrow C(\Omega_V)$ continuously. For NN_ε a multiple operator map of neural network type satisfying (11) with ε, we estimate as follows with $u_0 \in U$:

$$\sup_{\alpha\in K}|F(\alpha) - \mathrm{ev}_{x_0}\circ\mathrm{ev}_{u_0}\circ\mathrm{NN}_\varepsilon[\alpha]| = \sup_{\alpha\in K}\left|\mathrm{ev}_{x_0}\circ\mathrm{ev}_{u_0}\circ\big(G[\alpha] - \mathrm{NN}_\varepsilon[\alpha]\big)\right| \tag{39}$$
$$\le C_V\sup_{\alpha\in K}\left\|\big(G[\alpha] - \mathrm{NN}_\varepsilon[\alpha]\big)(u_0)\right\|_V \tag{40}$$
$$\le C_V\sup_{\alpha\in K}\|G[\alpha] - \mathrm{NN}_\varepsilon[\alpha]\|_{\mathrm{op}}$$
$$\le C_V\varepsilon \tag{41}$$
$$\le \varepsilon_0$$

where we used (37) for (39), (38) for (40) and (11) for (41). By Definition 3.6, for the fixed pair (u_0, x_0) there exists a ReLU neural network $\Phi_{x_0,M_U(u_0)}$ such that

$$\mathrm{ev}_{x_0}\circ\mathrm{ev}_{u_0}\circ\mathrm{NN}_\varepsilon[\alpha] = \Phi_{x_0,M_U(u_0)}(M_W(\alpha))$$

for all $\alpha \in W$. Hence, by Definition 2.9, the map $\mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \mathrm{NN}_\varepsilon : W \to \mathbb{R}$ is a functional of neural network type and we can therefore apply (36) to deduce that $\mathcal{C}(\mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \mathrm{NN}_\varepsilon) \geq \exp\left(c\varepsilon^{-1/[(\eta+1+\delta)r]}\right)$. Using the latter, we conclude:

$$\mathcal{C}(\mathrm{NN}_\varepsilon) \geq \mathcal{C}(\mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \mathrm{NN}_\varepsilon) \geq \exp\left(c\varepsilon^{-1/[(\eta+1+\delta)r]}\right).$$

□

Proof of Lemma 3.10. In the proof, $C > 0$ will denote a constant that may change from line to line.

Let $r \in \mathbb{N}$ and $\delta > 0$. Without loss of generality, assume that $\eta_W \leq \eta_U$. We apply [26, Theorem 2.11] to obtain a r-times Frechet differentiable functional $F : W \to \mathbb{R}$ and $\varepsilon_0 := \varepsilon_0(\eta_W, \delta, r)$ such that for any $\varepsilon \leq \varepsilon_0$ and functional of neural network type S_ε with

$$\sup_{\alpha \in K_W} |F(\alpha) - S_\varepsilon(\alpha)| \leq \varepsilon,$$

we have

$$\mathcal{C}(S_\varepsilon) \geq \exp\left(c\varepsilon^{-1/[(\eta_W+1+\delta)r]}\right) \tag{42}$$

for some $c := c(\eta_W, \delta, r) > 0$.

The rest of the proof is similar to the one of Theorem 3.8. We choose a nontrivial element $\phi \in V$. Then there exists $x_0 \in \Omega_V$ such that $\phi(x_0) \neq 0$; rescaling ϕ, we may assume $\phi(x_0) = 1$. We define the multiple operator map

$$G : W \times U \to V, \qquad G[\alpha][u] = F(\alpha)\phi.$$

Since F is r-times Frechet differentiable, G is also r-times Frechet differentiable on $W \times U$ by Assumption **N.1**. Also, for any fixed $u_0 \in U$ and every $\alpha \in W$, we have

$$\mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ G[\alpha] = G[\alpha][u_0](x_0) = F(\alpha)\phi(x_0) = F(\alpha). \tag{43}$$

Let $0 < \varepsilon \leq \overline{\varepsilon} := \varepsilon_0 / \max\{1, C_V\}$ where $C_V > 0$ is such that

$$\|v\|_{C(\Omega_V)} \leq C_V \|v\|_V \tag{44}$$

for all $v \in V$ due to the fact that $V \hookrightarrow C(\Omega_V)$ continuously. For NN_ε a symmetric multiple operator map of neural network type satisfying (14) with ε, we estimate as follows with $u_0 \in U$:

$$\sup_{\alpha \in K_W} |F(\alpha) - \mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \mathrm{NN}_\varepsilon[\alpha]| = \sup_{\alpha \in K_W} \left|\mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \big(G[\alpha] - \mathrm{NN}_\varepsilon[\alpha]\big)\right| \tag{45}$$

$$\leq C_V \sup_{\alpha \in K_W} \|G[\alpha][u_0] - \mathrm{NN}_\varepsilon[\alpha][u_0]\|_V \tag{46}$$

$$\leq \varepsilon_0 \tag{47}$$

where we used (43) for (45), (44) for (46) and (14) for (47). By Definition 3.9, for the fixed pair (u_0, x_0) there exists a ReLU neural network $\Phi_{x_0, M_U(u_0)}$ such that

$$\mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \mathrm{NN}_\varepsilon[\alpha] = \Phi_{x_0, M_U(u_0)}(M_W(\alpha))$$

for all $\alpha \in W$. Hence, by Definition 2.9, the map $\mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \mathrm{NN}_\varepsilon : W \to \mathbb{R}$ is a functional of neural network type and we can therefore apply (42) to deduce that $\mathcal{C}(\mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \mathrm{NN}_\varepsilon) \geq \exp\left(c\varepsilon^{-1/[(\eta_W+1+\delta)r]}\right)$. Using the latter, we conclude:

$$\mathcal{C}(\mathrm{NN}_\varepsilon) \geq \mathcal{C}(\mathrm{ev}_{x_0} \circ \mathrm{ev}_{u_0} \circ \mathrm{NN}_\varepsilon) \geq \exp\left(c\varepsilon^{-1/[(\eta_W+1+\delta)r]}\right).$$

□

The following result will be essential in the proof of Lemma 3.13.

Lemma B.1 (L^r-norm of sin)**.** *For every $1 \le r < \infty$ and every nonzero multi-index $\kappa \in \mathbb{N}^d$, one has*

$$\|\sin(\kappa \cdot x)\|_{\mathrm{L}^r([0,2\pi]^d)} = \left((2\pi)^{d-1} \int_0^{2\pi} |\sin t|^r \,\mathrm{d}t\right)^{1/r}.$$

In particular, the quantity $c_r := \|\sin(\kappa \cdot x)\|_{\mathrm{L}^r([0,2\pi]^d)}$ is independent of κ.

Proof. Let $\kappa = (\kappa_1, \dots, \kappa_d) \in \mathbb{N}^d$ be a non-zero multi-index. Pick an index $j \in \{1, \dots, d\}$ such that $\kappa_j \neq 0$. Writing $x = (x_j, x')$, where $x' = (x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_d)$, we have

$$\begin{aligned}
\int_{[0,2\pi]^d} |\sin(\kappa \cdot x)|^r \,\mathrm{d}x &= \int_{[0,2\pi]^{d-1}} \int_0^{2\pi} \left|\sin\left(\kappa_j x_j + \sum_{m \neq j} \kappa_m x_m\right)\right|^r \,\mathrm{d}x_j \,\mathrm{d}x' \\
&= \int_{[0,2\pi]^{d-1}} \frac{1}{\kappa_j} \int_{\sum_{m\neq j} \kappa_m x_m}^{\sum_{m\neq j} \kappa_m x_m + 2\pi\kappa_j} |\sin t|^r \,\mathrm{d}t \,\mathrm{d}x' \qquad (48) \\
&= \int_{[0,2\pi]^{d-1}} \frac{1}{\kappa_j} \sum_{n=0}^{\kappa_j - 1} \int_{\sum_{m\neq j} \kappa_m x_m + 2\pi n}^{\sum_{m\neq j} \kappa_m x_m + 2\pi(n+1)} |\sin t|^r \,\mathrm{d}t \,\mathrm{d}x' \\
&= \int_{[0,2\pi]^{d-1}} \frac{\kappa_j}{\kappa_j} \int_0^{2\pi} |\sin t|^r \,\mathrm{d}t \,\mathrm{d}x' \qquad (49) \\
&= (2\pi)^{d-1} \int_0^{2\pi} |\sin t|^r \,\mathrm{d}t
\end{aligned}$$

where we used the change of variables $t = \kappa_j x_j + \sum_{m\neq j} \kappa_m x_m$ for (48) and the fact that $|\sin t|^r$ is 2π-periodic for (49). Taking the r-th root yields the claim. □

Proof of Lemma 3.13. In the proof $C > 0$ denotes a constant that may change from line to line, independently of the summation indices and κ.

Up to an affine rescaling of the domain, it suffices to treat the case $\Omega_U = [0, 2\pi]^{d_U}$. For each multi-index $\kappa \in \mathbb{N}^{d_U}$, define

$$\tilde{e}_\kappa(x) := \sin(\kappa \cdot x), \qquad e_\kappa(x) := \frac{\tilde{e}_\kappa(x)}{\|\tilde{e}_\kappa\|_{\mathrm{L}^r(\Omega_U)}} = \frac{\tilde{e}_\kappa(x)}{c_r}$$

where we used Lemma B.1 for the last equality. These functions are orthogonal in $\mathrm{L}^2(\Omega_U)$, and hence linearly independent in $\mathrm{L}^r(\Omega_U)$, with $\|e_\kappa\|_{\mathrm{L}^r(\Omega_U)} = 1$. For each $\kappa \in \mathbb{N}^{d_U}$, define $e^*_\kappa \in \mathrm{L}^r(\Omega_U)^*$ as

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

By the identity $\sin(a)\sin(b) = \frac{1}{2}(\cos(a-b) - \cos(a+b))$, it is straightforward to check that $e^*_\kappa(e_{\kappa'}) = \delta_{\kappa\kappa'}$. Moreover, by Hölder's inequality with r' such that $1/r + 1/r' = 1$,

$$|e^*_\kappa(u)| \le \frac{2c_r c_{r'}}{(2\pi)^d} \|u\|_{\mathrm{L}^r(\Omega_U)} \le C\|u\|_{\mathrm{L}^r(\Omega_U)},$$

so $\|e^*_\kappa\|_{\mathrm{L}^r(\Omega_U)^*} \le C$ uniformly in κ.

Next, we enumerate $\{e_\kappa\}_{\kappa \in \mathbb{N}^{d_U}}$ as $\{e_j\}_{j=1}^\infty$ in such a way that $j \mapsto \|\kappa(j)\|_\infty$ is non-decreasing and consider functions on $[0, 2\pi]^{d_U}$ of the form

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

First, since $\|e_j\|_{\mathrm{L}^\infty(\Omega_U)} = c_r^{-1}$, $\|u\|_{\mathrm{L}^\infty(\Omega_U)} \le A c_r^{-1} \sum_{j=1}^\infty j^{-\eta}$ and the latter converges because $\eta > 1$. Choosing $A(\eta) > 0$ sufficiently small furthermore ensures that $\|u\|_{\mathrm{L}^\infty(\Omega_U)} \le \beta_U$ uniformly.

Second, we estimate the Lipschitz constant. For a function $f : [0, 2\pi]^{d_U} \mapsto \mathbb{R}$, we define

$$\mathrm{Lip}(f) = \inf\{L > 0 \mid |f(x) - f(y)| \le L\|x - y\|_2 \text{ for all } x, y\}$$

and note that that $\mathrm{Lip}(f_1 + f_2) \leq \mathrm{Lip}(f_1) + \mathrm{Lip}(f_2)$ as well as $\mathrm{Lip}(f) \leq C\|\nabla f\|_{\mathrm{L}^\infty(\Omega_U)}$ by the mean-value theorem. By noting that $\mathrm{Lip}(e_\kappa) \leq C\|\nabla e_\kappa\|_{\mathrm{L}^\infty(\Omega_U)} \leq C\|\kappa\|_\infty$, we estimate as follows:

$$\mathrm{Lip}(u) \leq A\sum_{j=1}^{\infty} j^{-\eta}\mathrm{Lip}(e_j) \leq CA\sum_{j=1}^{\infty} j^{-\eta}\|\kappa(j)\|_\infty. \tag{50}$$

Let us now consider the inverse enumeration $\kappa \mapsto j(\kappa)$. For a fixed $\kappa \in \mathbb{N}^{d_U}$, we write $K = \|\kappa\|_\infty$ and, due to the fact that $j \mapsto \kappa(j)$ is non-decreasing, we have

$$(K-1)^{d_U} \leq |\{\kappa' \in \mathbb{N}^{d_U} \mid \|\kappa'\|_\infty < K\}| \leq j(\kappa) \leq |\{\kappa' \in \mathbb{N}^{d_U} \mid \|\kappa'\|_\infty \leq K\}| = K^{d_U} \tag{51}$$

as well as

$$|\{\kappa' \in \mathbb{N}^{d_U} \mid \|\kappa'\|_\infty = K\}| \leq d_U K^{d_U-1}, \tag{52}$$

since fixing one coordinate equal to K leaves at most K^{d_U-1} choices for the remaining coordinates, and there are d_U possible coordinates that may attain the maximum. We continue our estimation:

$$\begin{aligned}
\sum_{j=1}^{\infty} j^{-\eta}\|\kappa(j)\|_\infty &= \sum_{K=1}^{\infty}\sum_{\|\kappa(j)\|_\infty = K} j(\kappa)^{-\eta}K \\
&= \sum_{\|\kappa(j)\|_\infty = 1} j(\kappa)^{-\eta} + \sum_{K=2}^{\infty}\sum_{\|\kappa(j)\|_\infty = K} j(\kappa)^{-\eta}K \\
&\leq 1 + \sum_{K=2}^{\infty}\sum_{\|\kappa(j)\|_\infty = K} K^1(K-1)^{-\eta d_U} && (53)\\
&\leq C\sum_{K=2}^{\infty} K^{d_U}(K-1)^{-\eta d_U} && (54)\\
&\leq C\sum_{K=2}^{\infty} K^{d_U(1-\eta)} && (55)
\end{aligned}$$

where we used (51) for (53), (52) for (54) and the fact that $K/2 \leq K-1$ for $K \geq 2$ for (55). The latter quantity converges since $\eta > 1 + \frac{1}{d_U}$ and, from (50), we therefore deduce that $A(\eta)$ can be chosen so that $\mathrm{Lip}(u) \leq L_U$ uniformly. This concludes the proof that U contains an infinite-dimensional hypercube. □

Proof of Corollary 3.14. Without loss of generality, assume that $d_W \geq d_U$ and pick $\eta > 1 + \frac{1}{d_W}$.

First, we verify that W satisfying Assumption **S.1** is a compact subset of a Banach space and contains an infinite-dimensional cube Q_η: the latter part of this claim is given by Lemma 3.13 when the Banach space is chosen to be $\mathrm{L}^{r_G}(\Omega_W)$ and $\eta > 1 + 1/d_W$. To show the former part, we start by assuming that $\{\alpha_i\}_{i=1}^\infty \subset W$. These functions are uniformly bounded and, since

$$|\alpha_i(x) - \alpha_i(y)| \leq L_W\|x - y\|,$$

also equicontinuous. Indeed, for every $\varepsilon > 0$, $\|x - y\| \leq \frac{\varepsilon}{L_W}$ implies that $|\alpha_i(x) - \alpha_i(y)| < \varepsilon$ for every $i \in \mathbb{N}$ and $x, y \in \Omega_W$. By the Arzela-Ascoli theorem, there therefore exists a subsequence $\{\alpha_{i_k}\}_{k=1}^\infty$ converging in $\mathrm{C}^0(\Omega_W)$ (and hence in $\mathrm{L}^{r_G}(\Omega_W)$) to some $\alpha \in \mathrm{L}^{r_G}(\Omega_W)$. To conclude compactness of W in $\mathrm{L}^{r_G}(\Omega_W)$, we need to show that $\alpha \in W$. First, we have

$$\|\alpha\|_{\mathrm{C}^0(\Omega_W)} \leq \|\alpha - \alpha_{i_k}\|_{\mathrm{C}^0(\Omega_W)} + \|\alpha_{i_k}\|_{\mathrm{C}^0(\Omega_W)} \leq \|\alpha - \alpha_{i_k}\|_{\mathrm{C}^0(\Omega_W)} + \beta_W$$

and taking the limit implies that $\|\alpha\|_{\mathrm{C}^0(\Omega_W)} = \|\alpha\|_{\mathrm{L}^\infty(\Omega_W)} \leq \beta_W$. Second, we note that

$$\begin{aligned}
|\alpha(x) - \alpha(y)| &\leq |\alpha(x) - \alpha_{i_k}(x)| + |\alpha_{i_k}(x) - \alpha_{i_k}(y)| + |\alpha(y) - \alpha_{i_k}(y)| \\
&\leq L_W\|x - y\| + |\alpha(x) - \alpha_{i_k}(x)| + |\alpha(y) - \alpha_{i_k}(y)|
\end{aligned}$$

and taking the limit shows that α is L_W-Lipschitz, and hence in W.

Second, by Remark 3.11, Lemma 3.10 remains valid when the norm is chosen to be $\sup_{\alpha \in K_W}\sup_{u \in K_U}\|\cdot\|_{\mathrm{L}^\infty}$ and in this formulation, the output class V need not be a Banach space. In particular, we can pick it to satisfy Assumption **S.1** and the claim of the corollary follows after an application of Lemma 3.10. □

Proof of Lemma 3.16. Let $u \in U$ and $x \in \Omega_V$. Then,

$$\begin{aligned}\mathrm{ev}_x \circ \mathrm{ev}_u \circ \mathrm{NN}_W[\alpha] &= \sum_{p=1}^{P}\sum_{k=1}^{H}\sum_{\ell=1}^{N} \theta_{pk\ell}\, l_p(M_W(\alpha))\, b_k(M_U(u))\, \tau_\ell(x) \\ &= \tilde{\mathrm{NN}}(M_U(u))(x)^\top l\,(M_W(\alpha)) \\ &=: \Phi_{x,M_U(u)}(M_W(\alpha))\end{aligned}$$

where $\mathrm{NN}_W[\alpha] = \mathrm{NN}[\alpha][\cdot](\cdot)$, $l : \mathbb{R}^m \mapsto \mathbb{R}^P$ is the parallelization of $\{l_p\}_{p=1}^P$ [41, Definition 2.7], i.e. the network of the form (1) such that $l(y) = (l_1(y), \ldots, l_P(y))$, and $\tilde{\mathrm{NN}}(M_U(u))(x) \in \mathbb{R}^P$ is a vector with entries

$$\sum_{k=1}^{H}\sum_{\ell=1}^{N} \theta_{pkl} b_k(M_U(u))\tau_\ell(x)$$

for $1 \le p \le P$. Therefore, $\mathrm{ev}_x \circ \mathrm{ev}_u \circ \mathrm{NN}[\alpha] = \Phi_{x,M_U(u)}(M_W(\alpha))$ is an inner product between a ReLU-neural network and fixed vector which, by (1), can easily be checked to be a ReLU-neural network as in (1).

The same argument can be applied with $\alpha \in W$ and $x \in \Omega_V$ to deduce that $\mathrm{ev}_x \circ \mathrm{ev}_\alpha \circ \mathrm{NN}_U[u] = \Psi_{x,M_W(\alpha)}(M_U(u))$ for some ReLU-neural network $\Psi_{x,M_W(\alpha)}$ and where $\mathrm{NN}_U[u] = \mathrm{NN}[\cdot][u](\cdot)$. We therefore conclude that NN is a symmetric multiple operator map of neural network type.

For complexity, we have

$$\mathcal{C}(\mathrm{NN}) \le \max\left\{ \sup_{x\in\Omega_V}\sup_{u\in U} N_\#\left(\Phi_{x,M_U(u)}\right),\ \sup_{x\in\Omega_V}\sup_{\alpha\in W} N_\#\left(\Psi_{x,M_W(\alpha)}\right)\right\}. \tag{56}$$

First, we assume that

$$\max\left\{ \sup_{x\in\Omega_V}\sup_{u\in U} N_\#\left(\Phi_{x,M_U(u)}\right),\ \sup_{x\in\Omega_V}\sup_{\alpha\in W} N_\#\left(\Psi_{x,M_W(\alpha)}\right)\right\} = \sup_{x\in\Omega_V}\sup_{u\in U} N_\#\left(\tilde{\mathrm{NN}}(M_U(u))(x)^\top l\right),$$

we can continue from (56) and estimate as follows:

$$\begin{aligned}\mathcal{C}(\mathrm{NN}) &\le \sup_{x\in\Omega_V}\sup_{u\in U} N_\#\left(\tilde{\mathrm{NN}}(M_U(u))(x)^\top l\right) \\ &\le \sup_{x\in\Omega_V}\sup_{u\in U} 2\|\tilde{\mathrm{NN}}(M_U(u))(x)^\top\|_0 + 2N_\#(l) \qquad (57)\end{aligned}$$

where we used [41, Remark 2.6] for (57). Now, if for some $1 \le p \le P$, $\sum_{k=1}^{H}\sum_{\ell=1}^{N} \theta_{pkl} b_k(M_U(u))\tau_\ell(x) \neq 0$, there exists some triple (p,k,ℓ) such that $\theta_{pkl} \neq 0$, $b_k(M_U(u)) \neq 0$ and $\tau_\ell(x) \neq 0$. This implies that b_k and τ_ℓ are nonzero networks from which we deduce that at least one coefficient in each has to be different from 0. In particular, this yields

$$\|\tilde{\mathrm{NN}}(M_U(u))(x)^\top\|_0 \le \|\Theta\|_0 + \sum_{k=1}^{H} N_\#(b_k) + \sum_{\ell=1}^{N} N_\#(\tau_\ell)$$

where $\Theta = \{\theta_{pk\ell}\}$. Using the latter and [41, Definition 2.7], continuing from (57), we obtain:

$$\mathcal{C}(\mathrm{NN}) \le 2\left(\|\Theta\|_0 + HK_2 + NK_1 + \sum_{p=1}^{P} N_\#(l_p)\right) \le 2(\|\Theta\|_0 + HK_2 + NK_1 + PK_3).$$

If one assumes that

$$\max\left\{ \sup_{x\in\Omega_V}\sup_{u\in U} N_\#\left(\Phi_{x,M_U(u)}\right),\ \sup_{x\in\Omega_V}\sup_{\alpha\in W} N_\#\left(\Psi_{x,M_W(\alpha)}\right)\right\} = \sup_{x\in\Omega_V}\sup_{\alpha\in W} N_\#\left(\Psi_{x,M_W(\alpha)}\right)$$

instead, the same upper bound on complexity can be attained analogously.

□

Proof of Theorem 3.19. 1. The first claim of the theorem is a combination of Corollary 3.14 and Lemma 3.16: indeed, the architecture is symmetric multiple operator map of neural network type and its complexity is upper bounded by $2(\|\Theta\|_0 + HK_2 + NK_1 + PK_3)$.

2. From part 1, we may take $r = 1$ to obtain $G : \mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U) \to V$ which is Frechet differentiable on $\mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U)$. Specifically, from the proof of Corollary 3.14, we know that

$$G[\alpha][u](x) = F(\alpha)\phi(x),$$

where $F : \mathrm{L}^{r_G}(\Omega_W) \to \mathbb{R}$ is the Frechet differentiable functional provided by [26, Theorem 2.11], and $\phi \in V$ is a fixed nontrivial function.

Next, the proof of [26, Lemma A.7] shows that $\sup_{\alpha \in \mathrm{L}^{r_G}(\Omega_W)} \|DF(\alpha)\|_{\mathrm{L}^{r_G}(\Omega_W)^*} < \infty$. Hence, F is Lipschitz on $\mathrm{L}^{r_G}(\Omega_W)$ with Lipschitz constant $\mathrm{Lip}(F)$. It remains to deduce the two required Lipschitz properties. First, for fixed α and any u_1, u_2, we have

$$\|G[\alpha][u_1] - G[\alpha][u_2]\|_{\mathrm{L}^\infty(\Omega_V)} = \|F(\alpha)\phi - F(\alpha)\phi\|_{\mathrm{L}^\infty(\Omega_V)} = 0 \leq L_{\mathcal{G}}\|u_1 - u_2\|_{\mathrm{L}^{r_{\mathcal{G}}}(\Omega_U)}$$

for any $L_{\mathcal{G}} > 0$ and $r_{\mathcal{G}} \geq 1$. Second, for α_1, α_2 and any u, we estimate as follows:

$$\begin{aligned}\|G[\alpha_1] - G[\alpha_2]\|_{\mathrm{L}^\infty(\{u:\|u\|_{\mathrm{L}^\infty(\Omega_U)} \leq \beta_U\} \times \Omega_V)} &= |F(\alpha_1) - F(\alpha_2)|\, \|\phi\|_{\mathrm{L}^\infty(\Omega_V)} \\ &\leq \mathrm{Lip}(F)\beta_V \|\alpha_1 - \alpha_2\|_{\mathrm{L}^{r_G}(\Omega_W)} \\ &=: L_G \|\alpha_1 - \alpha_2\|_{\mathrm{L}^{r_G}(\Omega_W)}\end{aligned}$$

where we used the fact that F in Lipschitz in L^{r_G} and the fact that $\phi \in V$ for the inequality. This concludes the proof.

3. The lower bound in (16) is given by combining parts 1 and 2 of the theorem. The upper bound is a direct consequence of Theorem 3.1.

□

C An Extension of DeepONet to Multi-Task Learning

Proof of Lemma 3.21. Let $(\alpha, u) \in W \times U$ and $x \in \Omega_V$. Then

$$\begin{aligned}\mathrm{ev}_x \circ \mathrm{NN}[\alpha][u] &= \sum_{k=1}^{H} \sum_{\ell=1}^{N} \theta_{k\ell}\, b_k(M_W(\alpha), M_U(u))\, \tau_\ell(x) \\ &= \sum_{k=1}^{H} \left(\sum_{\ell=1}^{N} \theta_{k\ell} \tau_\ell(x) \right) b_k(M_W(\alpha), M_U(u)) \\ &=: \tilde{\tau}(x)^\top b(M_W(\alpha), M_U(u)) \\ &=: \Phi_x(M_W(\alpha), M_U(u)),\end{aligned}$$

where $b : \mathbb{R}^{m+q} \to \mathbb{R}^H$ is the parallelization of $\{b_k\}_{k=1}^H$ [41, Definition 2.7], i.e. the network of the form 1 such that $b(y, z) = \big(b_1(y, z), \ldots, b_H(y, z)\big)$, and $\tilde{\tau}(x) \in \mathbb{R}^H$ is the vector with entries

$$\tilde{\tau}_k(x) := \sum_{\ell=1}^{N} \theta_{k\ell} \tau_\ell(x)$$

for $1 \leq k \leq H$. Therefore, for every $x \in \Omega_V$, the map $(\alpha, u) \mapsto \mathrm{ev}_x\, \mathrm{NN}[\alpha][u]$ is the inner product between the ReLU neural network b evaluated at the concatenated linear encoding $(M_W(\alpha), M_U(u))$ and the fixed vector $\tilde{\tau}(x)$. Hence, by (1), Φ_x is itself a ReLU neural network, and NN is an operator map of neural network type from $W \times U$ to V.

For complexity, by Definition 2.10,

$$\mathcal{C}(\mathrm{NN}) \leq \sup_{x \in \Omega_V} N_{\#}\big(\tilde{\tau}(x)^\top b\big)$$

$$\leq \sup_{x\in\Omega_V} \Big(2\|\tilde{\tau}(x)^\top\|_0 + 2N_\#(b)\Big) \tag{58}$$

where we used [41, Remark 2.6] for (58). Now, if for some $1 \leq k \leq H$, $\tilde{\tau}_k(x) \neq 0$, then $\sum_{\ell=1}^N \theta_{k\ell}\tau_\ell(x) \neq 0$. Hence there exists some $\ell \in \{1,\dots,N\}$ such that $\theta_{k\ell} \neq 0$ and $\tau_\ell(x) \neq 0$. This implies that τ_ℓ is a nonzero network from which we deduce that at least one coefficient of τ_ℓ is different from 0. Therefore,

$$\|\tilde{\tau}(x)^\top\|_0 \leq \|\Theta\|_0 + \sum_{\ell=1}^N N_\#(\tau_\ell).$$

Using the latter estimate and [41, Definition 2.7], we continue from (58) and obtain

$$\begin{aligned}\mathcal{C}(\mathrm{NN}) &\leq 2\left(\|\Theta\|_0 + \sum_{\ell=1}^N N_\#(\tau_\ell) + N_\#(b)\right)\\ &\leq 2\left(\|\Theta\|_0 + NK_1 + \sum_{k=1}^H N_\#(b_k)\right)\\ &\leq 2\big(\|\Theta\|_0 + NK_1 + HK_2\big).\end{aligned}$$

□

Proof of Lemma 3.23. We start by defining $\tilde{e}_j := (e_j, 0) \in W \times \{0\}$ for $j \in \mathbb{N}$. We verify the conditions in Definition 2.8.

First, the family $\{\tilde{e}_j\}_{j\in\mathbb{N}}$ is linearly independent. Indeed, if $\sum_{j=1}^m a_j\tilde{e}_j = 0$ for some $m \in \mathbb{N}$ and scalars $a_1,\dots,a_m$, then $\sum_{j=1}^m a_j e_j = 0$ in W. Since $\{e_j\}_{j\in\mathbb{N}}$ is linearly independent, it follows that $a_j = 0$ for all j.

Second, we have $\|\tilde{e}_j\|_{W\times U} = \|(e_j,0)\|_{W\times U} = \|e_j\|_W = 1$.

Third, we check the cube inclusion. Let $y_j \in [0,1]$ for all j, and consider

$$\tilde{w} := A\sum_{j=1}^\infty j^{-\eta} y_j \tilde{e}_j = A\sum_{j=1}^\infty j^{-\eta} y_j (e_j, 0) = \left(A\sum_{j=1}^\infty j^{-\eta} y_j e_j,\, 0\right).$$

Since $Q_\eta(A;\{e_j\}_{j\in\mathbb{N}}) \subset K_W$, the first component belongs to K_W. Therefore $\tilde{w} \in K_W \times \{0\}$.

Fourth, we construct a bounded biorthogonal sequence in $(W\times U)^*$. Let $\{e_j^*\}_{j\in\mathbb{N}} \subset W^*$ be a bounded biorthogonal sequence for $\{e_j\}_{j\in\mathbb{N}}$, and define $\tilde{e}_j^*(w,u) := e_j^*(w)$ for $(w,u) \in W\times U$. Then $\tilde{e}_j^* \in (W\times U)^*$, and for all $j,k \in \mathbb{N}$,

$$\tilde{e}_j^*(\tilde{e}_k) = \tilde{e}_j^*(e_k,0) = e_j^*(e_k) = \delta_{jk}.$$

Moreover, $\{\tilde{e}_j^*\}_{j\in\mathbb{N}}$ is uniformly bounded, since by Assumption **N.1** we have

$$|\tilde{e}_j^*(w,u)| = |e_j^*(w)| \leq \|e_j^*\|_{W^*}\|w\|_W \leq C_{\text{prod}}\|e_j^*\|_{W^*}\,\|(w,u)\|_{W\times U}$$

which implies $\|\tilde{e}_j^*\|_{(W\times U)^*} \leq C_{\text{prod}}\|e_j^*\|_{W^*}$. This concludes the proof. □

The following proposition is an essential building block for the proof of Proposition 3.25.

Proposition C.1 (Functional Approximation in Product Spaces)**.** *Let $d_W, d_U > 0$ be integers,*

$$\gamma_W, \gamma_U, \beta_W, \beta_U, L_W, L_U > 0$$

*and assume that $W(d_W,\gamma_W,L_W,\beta_W)$, $U(d_U,\gamma_U,L_U,\beta_U)$ satisfy Assumption **S.1**. We equip the product space $W\times U$ with a norm $\|\cdot\|_{W\times U}$ that satisfies Assumption **N.2**. Let $f : \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0)\times\mathcal{B}_{\beta_U,\|\cdot\|_{\mathrm{L}^\infty},\Omega_U}(0) \mapsto \mathbb{R}$ be a functional such that*

$$|f((\alpha_1,u_1)) - f((\alpha_2,u_2))| \leq L_f\|(\alpha_1,u_1)-(\alpha_2,u_2)\|_{W\times U} \tag{59}$$

for all $(\alpha_1,u_1),(\alpha_2,u_2) \in \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0)\times\mathcal{B}_{\beta_U,\|\cdot\|_{\mathrm{L}^\infty},\Omega_U}(0)$. There exist constants

- *C depending on β_W, β_U, C_{prod}, L_f*
- *C_W depending on C_{prod}, L_f, L_W*
- *C_U depending on C_{prod}, L_f, L_U*

such that the following holds. For any $\varepsilon > 0$,

- *let $\delta_W = C_W\varepsilon$ and let $\{a_i\}_{i=1}^{n_{c_W}} \subset \Omega_W$ be points so that $\{\mathcal{B}_{\delta_W}(a_i)\}_{i=1}^{n_{c_W}}$ is a cover of Ω_W for some n_{c_W};*
- *let $\delta_U = C_U\varepsilon$ and let $\{c_i\}_{i=1}^{n_{c_U}} \subset \Omega_U$ be points so that $\{\mathcal{B}_{\delta_U}(c_i)\}_{i=1}^{n_{c_U}}$ is a cover of Ω_U for some n_{c_U};*
- *let $H = 2C\sqrt{n_{c_W}+n_{c_U}}\varepsilon^{-1}$ and consider the network class $\mathcal{F}_{\mathrm{NN}}(n_{c_W}+n_{c_U}, 1, L, p, K, \kappa, R)$ with parameters scaling as*

$$
\begin{aligned}
&L = \mathcal{O}\left((n_{c_W}+n_{c_U})^2\log(n_{c_W}+n_{c_U}) + (n_{c_W}+n_{c_U})^2[\log(\varepsilon^{-1})+\log(2)]\right), \qquad p = \mathcal{O}(1),\\
&K = \mathcal{O}\left((n_{c_W}+n_{c_U})^2\log(n_{c_W}+n_{c_U}) + (n_{c_W}+n_{c_U})^2[\log(\varepsilon^{-1})+\log(2)]\right),\\
&\kappa = \mathcal{O}((n_{c_W}+n_{c_U})^{(n_{c_W}+n_{c_U})/2+1}\varepsilon^{-(n_{c_W}+n_{c_U}+1)}2^{n_{c_W}+n_{c_U}+1}), \qquad R = 1
\end{aligned}
$$

where the constants hidden in $\mathcal{O}$ depend on β_W, β_U, C_{prod} and L_f,

Then, there exists networks $\{b_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{F}_{\mathrm{NN}}(n_{c_W}+n_{c_U}, 1, L, p, K, \kappa, R)$ and functions $\{\alpha_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0)$, $\{u_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{B}_{\beta_U,\|\cdot\|_{\mathrm{L}^\infty},\Omega_U}(0)$ such that

$$
\sup_{\alpha\in W}\sup_{u\in U}\left|f(\alpha,u) - \sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} f(\alpha_k,u_k)b_k(\boldsymbol{\alpha},\boldsymbol{u})\right| \le \varepsilon,
$$

where $\boldsymbol{\alpha} = \frac{\max\{\beta_W,\beta_U\}}{\beta_W}(\alpha(a_1),\alpha(a_2),\ldots,\alpha(a_{n_{c_W}}))^\top$, $\boldsymbol{u} = \frac{\max\{\beta_W,\beta_U\}}{\beta_U}(u(c_1),u(c_2),\ldots,u(c_{n_{c_U}}))^\top$. We also have that $0 \le b_k \le 1$ for all $1 \le k \le H^{n_{c_W}+n_{c_W}}$.

Proof. Let $\delta_W, \delta_U > 0$, and let $\mathcal{C}_W = \{\mathcal{B}_{\delta_W}(a_i)\}_{i=1}^{n_{c_W}}, \mathcal{C}_U = \{\mathcal{B}_{\delta_U}(c_m)\}_{m=1}^{n_{c_U}}$ be finite covers of Ω_W and Ω_U by Euclidean balls, respectively. By [31, Lemma 1], there exist partitions of unity $\{\rho_i\}_{i=1}^{n_{c_W}} \subset \mathrm{C}^\infty(\Omega_W)$ and $\{\omega_m\}_{m=1}^{n_{c_U}} \subset \mathrm{C}^\infty(\Omega_U)$ subordinate to $\mathcal{C}_W$ and $\mathcal{C}_U$, respectively.

We define the discrete-to-continuum liftings

$$
I_{\mathcal{C}_W} : [-\beta_W,\beta_W]^{n_{c_W}} \to \mathrm{C}^\infty(\Omega_W), \qquad I_{\mathcal{C}_U} : [-\beta_U,\beta_U]^{n_{c_U}} \to \mathrm{C}^\infty(\Omega_U)
$$

by

$$
I_{\mathcal{C}_W}[z](x) = \sum_{i=1}^{n_{c_W}} [z]_i\,\rho_i(x), \qquad I_{\mathcal{C}_U}[w](x) = \sum_{m=1}^{n_{c_U}} [w]_m\,\omega_m(x),
$$

and the continuum-to-discrete projections

$$
P_{\mathcal{C}_W} : \mathrm{C}^0(\Omega_W) \to [-\beta_W,\beta_W]^{n_{c_W}}, \qquad P_{\mathcal{C}_U} : \mathrm{C}^0(\Omega_U) \to [-\beta_U,\beta_U]^{n_{c_U}}
$$

by

$$
P_{\mathcal{C}_W}(\alpha) = \big(\alpha(a_1),\ldots,\alpha(a_{n_{c_W}})\big)^\top, \qquad P_{\mathcal{C}_U}(u) = \big(u(c_1),\ldots,u(c_{n_{c_U}})\big)^\top.
$$

Assume now that $f : W \times U \to \mathbb{R}$ is Lipschitz with respect to the product norm $\|(\alpha,u)\|_{W\times U}$ with Lipschitz constant L_f.

We first estimate the discretization error. For $\alpha \in W$ and $x \in \Omega_W$, we have

$$
\begin{aligned}
|\alpha(x) - I_{\mathcal{C}_W}[P_{\mathcal{C}_W}(\alpha)](x)| &= \left|\sum_{i=1}^{n_{c_W}} \big(\alpha(x)-\alpha(a_i)\big)\rho_i(x)\right|\\
&\le \sum_{i:\|x-a_i\|_2\le\delta_W} |\alpha(x)-\alpha(a_i)|\,|\rho_i(x)|
\end{aligned}
$$

$$\leq L_W \delta_W \sum_{i=1}^{n_{c_W}} \rho_i(x)$$
$$= L_W \delta_W$$

which implies $\|\alpha - I_{\mathcal{C}_W}[P_{\mathcal{C}_W}(\alpha)]\|_{\mathrm{L}^\infty(\Omega_W)} \leq L_W \delta_W$. Similarly, for every $u \in U$, we deduce that $\|u - I_{\mathcal{C}_U}[P_{\mathcal{C}_U}(u)]\|_{\mathrm{L}^\infty(\Omega_U)} \leq L_U \delta_U$. Therefore, by the latter two estimates and (59),

$$\big\|\big(\alpha, u\big) - \big(I_{\mathcal{C}_W}[P_{\mathcal{C}_W}(\alpha)],\, I_{\mathcal{C}_U}[P_{\mathcal{C}_U}(u)]\big)\big\|_{W\times U} \leq C_{\text{prod}} \max\{L_W\delta_W,\ L_U\delta_U\}.$$

Finally, using the Lipschitz continuity of f, we obtain

$$\begin{aligned}\big|f(\alpha,u) - f(I_{\mathcal{C}_W}[P_{\mathcal{C}_W}(\alpha)],\, I_{\mathcal{C}_U}[P_{\mathcal{C}_U}(u)])\big| &\leq L_f \big\|\big(\alpha,u\big) - \big(I_{\mathcal{C}_W}[P_{\mathcal{C}_W}(\alpha)],\, I_{\mathcal{C}_U}[P_{\mathcal{C}_U}(u)]\big)\big\|_{W\times U}\\ &\leq C L_f \max\{L_W\delta_W,\ L_U\delta_U\},\end{aligned}$$

and choosing $\delta_W = \frac{\varepsilon}{2C_{\text{prod}}L_f L_W}$, $\delta_U = \frac{\varepsilon}{2C_{\text{prod}}L_f L_U}$ therefore yields

$$\big|f(\alpha,u) - f(I_{\mathcal{C}_W}[P_{\mathcal{C}_W}(\alpha)],\, I_{\mathcal{C}_U}[P_{\mathcal{C}_U}(u)])\big| \leq \frac{\varepsilon}{2}. \tag{60}$$

Next, we define $\hat{f} : [-\beta_W, \beta_W]^{n_{c_W}} \times [-\beta_U, \beta_U]^{n_{c_U}} \to \mathbb{R}$ by $\hat{f}(z,w) := f(I_{\mathcal{C}_W}[z],\, I_{\mathcal{C}_U}[w])$. This function serves as a finite-dimensional surrogate of the functional f. Our strategy is to approximate $\hat{f}$ using neural-network approximation results in finite dimensions, and then transfer this approximation back to the original functional f. We estimate as follows for $(z_1,w_1),(z_2,w_2) \in [-\beta_W,\beta_W]^{n_{c_W}} \times [-\beta_U,\beta_U]^{n_{c_U}}$:

$$\begin{aligned}|\hat{f}(z_1,w_1) - \hat{f}(z_2,w_2)| &= |f(I_{\mathcal{C}_W}[z_1], I_{\mathcal{C}_U}[w_1]) - f(I_{\mathcal{C}_W}[z_2], I_{\mathcal{C}_U}[w_2])|\\ &\leq L_f \,\|(I_{\mathcal{C}_W}[z_1] - I_{\mathcal{C}_W}[z_2],\, I_{\mathcal{C}_U}[w_1] - I_{\mathcal{C}_U}[w_2])\|_{W\times U}\\ &\leq C_{\text{prod}} L_f \max\{\|I_{\mathcal{C}_W}[z_1] - I_{\mathcal{C}_W}[z_2]\|_{\mathrm{L}^\infty(\Omega_W)}, \|I_{\mathcal{C}_U}[w_1] - I_{\mathcal{C}_U}[w_2]\|_{\mathrm{L}^\infty(\Omega_U)}\} \qquad (61)\end{aligned}$$

where we Assumption **N.2** for (61). We note that

$$\begin{aligned}\|I_{\mathcal{C}_W}[z_1] - I_{\mathcal{C}_W}[z_2]\|_{\mathrm{L}^\infty(\Omega_W)} &\leq \sup_{x\in\Omega_W} \sum_{i=1}^{n_{c_W}} |[z_1]_i - [z_2]_i|\rho_i(x)\\ &\leq \|z_1 - z_2\|_{\ell^2(\mathbb{R}^{n_{c_W}})} \sup_{x\in\Omega_W} \sqrt{\sum_{i=1}^{n_{c_W}} \rho_i^2(x)}\\ &\leq \|z_1 - z_2\|_{\ell^2(\mathbb{R}^{n_{c_W}})} \sup_{x\in\Omega_W} \sqrt{\sum_{i=1}^{n_{c_W}} \rho_i(x)} \qquad (62)\\ &\leq \|z_1 - z_2\|_{\ell^2(\mathbb{R}^{n_{c_W}})} \qquad (63)\end{aligned}$$

where we used the fact that $0 \leq \rho_i \leq 1$ for (62). The same argument can be repeated for $\|I_{\mathcal{C}_U}[w_1] - I_{\mathcal{C}_U}[w_2]\|_{\mathrm{L}^\infty(\Omega_U)}$ so that inserting (63) into (61) yields

$$\begin{aligned}|f(I_{\mathcal{C}_W}[z_1], I_{\mathcal{C}_U}[w_1]) - f(I_{\mathcal{C}_W}[z_2], I_{\mathcal{C}_U}[w_2])| &\leq C_{\text{prod}} L_f \max\Big\{\|z_1 - z_2\|_{\ell^2(\mathbb{R}^{n_{c_W}})}, \|w_1 - w_2\|_{\ell^2(\mathbb{R}^{n_{c_U}})}\Big\}\\ &\leq C_{\text{prod}} L_f \sqrt{\|z_1 - z_2\|^2_{\ell^2(\mathbb{R}^{n_{c_W}})} + \|w_1 - w_2\|^2_{\ell^2(\mathbb{R}^{n_{c_U}})}} \qquad (64)\\ &= C_{\text{prod}} L_f \|(z_1 - z_2, w_1 - w_2)\|_{\ell^2(\mathbb{R}^{n_{c_W}+n_{c_U}})} \qquad (65)\end{aligned}$$

where we used $\max\{a,b\} \leq \sqrt{a^2+b^2}$ for (64). The latter shows that $\hat{f}$ is Lipschitz on the compact cube $[-\beta_W,\beta_W]^{n_{c_W}} \times [-\beta_U,\beta_U]^{n_{c_U}}$.

With $\beta := \max\{\beta_W, \beta_U\}$, let $T : [-\beta,\beta]^{n_{c_W}+n_{c_U}} \to [-\beta_W,\beta_W]^{n_{c_W}} \times [-\beta_U,\beta_U]^{n_{c_U}}$, be defined by

$$T(\xi,\zeta) := \left(\frac{\beta_W}{\beta}\,\xi,\ \frac{\beta_U}{\beta}\,\zeta\right), \qquad \xi \in \mathbb{R}^{n_{c_W}},\ \zeta \in \mathbb{R}^{n_{c_U}}.$$

This map is 1-Lipschitz with respect to the $\ell^2(\mathbb{R}^{n_{c_W}+n_{c_U}})$-norm and consequently, the function

$$\tilde{f} : [-\beta,\beta]^{n_{c_W}+n_{c_U}} \to \mathbb{R}, \qquad \tilde{f}(\xi,\zeta) := \hat{f}(T(\xi,\zeta))$$

is $C_{\text{prod}}L_f$-Lipschitz with respect to the $\ell^2(\mathbb{R}^{n_{c_W}+n_{c_U}})$-norm as a composition of Lipschitz functions by (65). We conclude that $\tilde{f} \in V(n_{c_W}+n_{c_U},\beta,C_{\text{prod}}L_f,C_{\tilde{f}})$ for some set of functions V satisfying Assumption **S.1** and where $C_{\tilde{f}} > 0$ is a constant depending on $\tilde{f}$. We can therefore apply [31, Theorem 5]: specifically, there exists a constant C depending on β_W, β_U, C_{prod}, L_f so that the following holds. Let $H := 2C\sqrt{n_{c_W}+n_{c_U}}\varepsilon^{-1}$ and consider the network class $\mathcal{F}_{\text{NN}}(n_{c_W}+n_{c_W},1,L,p,K,\kappa,R)$ whose parameters scale as

$$\begin{aligned}
&L = \mathcal{O}\left((n_{c_W}+n_{c_U})^2\log(n_{c_W}+n_{c_U}) + (n_{c_W}+n_{c_U})^2[\log(\varepsilon^{-1})+\log(2)]\right), \qquad p = \mathcal{O}(1),\\
&K = \mathcal{O}\left((n_{c_W}+n_{c_U})^2\log(n_{c_W}+n_{c_U}) + (n_{c_W}+n_{c_U})^2[\log(\varepsilon^{-1})+\log(2)]\right),\\
&\kappa = \mathcal{O}((n_{c_W}+n_{c_U})^{(n_{c_W}+n_{c_U})/2+1}\varepsilon^{-(n_{c_W}+n_{c_U}+1)}2^{n_{c_W}+n_{c_U}+1}), \qquad R = 1
\end{aligned}$$

where the constants hidden in $\mathcal{O}$ depend on β_W, β_U, C_{prod} and L_f. Then, there exists:

- networks $\{b_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{F}_{\text{NN}}(n_{c_W}+n_{c_U},1,L,p,K,\kappa,R)$
- points $\{s_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset [-\beta,\beta]^{n_{c_W}+n_{c_U}}$

such that

$$\sup_{(\xi,\zeta)\in[-\beta,\beta]^{n_{c_W}+n_{c_U}}} \left|\tilde{f}(\xi,\zeta) - \sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} \tilde{f}(s_k)\, b_k(\xi,\zeta)\right| \le \frac{\varepsilon}{2}. \tag{66}$$

We also note that [31, Theorem 5] yields $0 \le b_k \le 1$. For any $(\alpha,u) \in W \times U$, we define the inverse-rescaled coordinates

$$(\xi_\alpha,\zeta_u) := T^{-1}(P_{\mathcal{C}_W}(\alpha),P_{\mathcal{C}_U}(u)) = \left(\frac{\beta}{\beta_W}P_{\mathcal{C}_W}(\alpha),\, \frac{\beta}{\beta_U}P_{\mathcal{C}_U}(u)\right) \in [-\beta,\beta]^{n_{c_W}+n_{c_U}},$$

since $\|\alpha\|_{\mathrm{L}^\infty(\Omega_W)} \le \beta_W$ and $\|u\|_{\mathrm{L}^\infty(\Omega_U)} \le \beta_U$. This implies that $\hat{f}(P_{\mathcal{C}_W}(\alpha),P_{\mathcal{C}_U}(u)) = \tilde{f}(\zeta_\alpha,\xi_u)$ and therefore, by (66),

$$\sup_{\alpha\in W}\sup_{u\in U}\left|\hat{f}(P_{\mathcal{C}_W}(\alpha),P_{\mathcal{C}_U}(u)) - \sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} \tilde{f}(s_k)\, b_k(\xi_\alpha,\zeta_u)\right| \le \frac{\varepsilon}{2}. \tag{67}$$

Combining (60) and (67), we conclude as follows:

$$\begin{aligned}
&\sup_{\alpha\in W}\sup_{u\in U}\left|f(\alpha,u) - \sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} \tilde{f}(s_k)\, b_k\left(\frac{\beta}{\beta_W}P_{\mathcal{C}_W}(\alpha),\, \frac{\beta}{\beta_U}P_{\mathcal{C}_U}(u)\right)\right|\\
&\le \sup_{\alpha\in W}\sup_{u\in U}|f(\alpha,u) - \hat{f}(P_{\mathcal{C}_W}(\alpha),P_{\mathcal{C}_U}(u))|\\
&+ \sup_{\alpha\in W}\sup_{u\in U}\left|\hat{f}(P_{\mathcal{C}_W}(\alpha),P_{\mathcal{C}_U}(u)) - \sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} \tilde{f}(s_k)\, b_k\left(\frac{\beta}{\beta_W}P_{\mathcal{C}_W}(\alpha),\, \frac{\beta}{\beta_U}P_{\mathcal{C}_U}(u)\right)\right|\\
&\le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.
\end{aligned}$$

Finally, for each sampling point $s_k = (z_k,w_k) \in [-\beta,\beta]^{n_{c_W}} \times [-\beta,\beta]^{n_{c_U}}$, we have

$$\tilde{f}(s_k) = \hat{f}\left(\frac{\beta_W}{\beta}z_k, \frac{\beta_U}{\beta}w_k\right) = f\left(\frac{\beta_W}{\beta}I_{\mathcal{C}_W}[z_k], \frac{\beta_U}{\beta}I_{\mathcal{C}_U}[w_k]\right) =: f(\alpha_k,u_k)$$

where $\alpha_k \in \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0)$ and $u_k \in \mathcal{B}_{\beta_U,\|\cdot\|_{\mathrm{L}^\infty},\Omega_U}(0)$, which yields the claimed representation.

□

Remark C.2 (Uniform Functional Approximation in Product Spaces). We can extend Proposition C.1 to families of functionals. The same argument as in [47, Remark 3.7] shows that the construction is uniform over any family $\{f_j\}_{j\in J}$ of functionals defined using the same spaces W and U, provided the constants entering the proof are bounded uniformly in j. More precisely, let us assume that

$$\sup_{j\in J} L_{f_j} < \infty \qquad \text{and} \qquad \sup_{j\in J} C_{\hat{f}_j} = \sup_{j\in J}\sup_{\alpha\in W}\sup_{u\in U} |f_j(\alpha,u)| < \infty,$$

where L_{f_j} is the Lipschitz constant of f_j, and $C_{\hat{f}_j}$ denotes the uniform bound placing the associated finite-dimensional surrogate $\hat{f}_j$ in the same approximation class as in the proof of Proposition C.1. Then there exist sampling pairs $\{(\alpha_k,u_k)\}_{k=1}^{H^{n_{cW}+n_{cU}}}$ and neural networks $\{b_k\}_{k=1}^{H^{n_{cW}+n_{cU}}}$ in the same ε-dependent network class as in Proposition C.1, such that

$$\sup_{j\in J}\sup_{\alpha\in W}\sup_{u\in U}\left| f_j(\alpha,u) - \sum_{k=1}^{H^{n_{cW}+n_{cU}}} f_j(\alpha_k,u_k)\, b_k(\boldsymbol{\alpha},\boldsymbol{u})\right| \le \varepsilon.$$

In particular, the conclusion always applies to any finite family of functionals.

Proof of Proposition 3.25. By Assumption, for all $(\alpha,u)\in W\times U$, the functions $x\mapsto G[\alpha][u](x)$ are in V. Consequently, we can apply [31, Theorem 5]. Specifically, there exists a constant C depending on γ_V, L_V so that the following holds. For $\varepsilon_0>0$, let $N := C\sqrt{d_V}\varepsilon_0^{-1}$ and consider the network class $\mathcal{F}_1 := \mathcal{F}_{\mathrm{NN}}(d_V,1,L_1,p_1,K_1,\kappa_1,R_1)$ whose parameters scale as

$$\begin{aligned} &L_1 = \mathcal{O}\left(d_V^2\log d_V + d_V^2\log(\varepsilon_0^{-1})\right), \quad p_1=\mathcal{O}(1), \quad K_1 = \mathcal{O}\left(d_V^2\log d_V + d_V^2\log(\varepsilon_0^{-1})\right),\\ &\kappa_1 = \mathcal{O}(d_V^{d_V/2+1}\varepsilon_0^{-(d_V+1)}), \qquad\qquad R_1 = 1 \end{aligned}$$

where the constants hidden in $\mathcal{O}$ depend on γ_V and L_V. Then, there exists

- networks $\{\tau_\ell\}_{k=1}^{N^{d_V}}\subset\mathcal{F}_1$
- points $\{v_\ell\}_{\ell=1}^{N^{d_V}}\subset\Omega_V$

such that

$$\sup_{x\in\Omega_V}\left|G[\alpha][u](x) - \sum_{\ell=1}^{N^{d_V}} G[\alpha][u](v_\ell)\,\tau_\ell(x)\right| \le \varepsilon_0. \tag{68}$$

Notably, we also recall from the proof of [31, Theorem 5] that $0\le\tau_\ell(x)\le 1$ (where the last inequality follows by definition of the network class $\mathcal{F}_1$ with $R_2=1$).

Next, we consider the N^{d_V} functionals $f_\ell : \mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0)\times\mathcal{B}_{\beta_U,\|\cdot\|_{\mathrm{L}^\infty},\Omega_U}(0)\mapsto\mathbb{R}$ defined by $f_\ell(\alpha,u)=G[\alpha][u](v_\ell)$. For $(\alpha_1,u_1),(\alpha_2,u_2)\in\mathcal{B}_{\beta_W,\|\cdot\|_{\mathrm{L}^\infty},\Omega_W}(0)\times\mathcal{B}_{\beta_U,\|\cdot\|_{\mathrm{L}^\infty},\Omega_U}(0)$, we estimate as follows:

$$\begin{aligned} |f_\ell(\alpha_1,u_1)-f_\ell(\alpha_2,u_2)| &= |G[\alpha_1][u_1](v_\ell)-G[\alpha_2][u_2](v_\ell)|\\ &\le |G[\alpha_1][u_1](v_\ell)-G[\alpha_1][u_2](v_\ell)| + |G[\alpha_1][u_2](v_\ell)-G[\alpha_2][u_2](v_\ell)|\\ &\le \|G[\alpha_1][u_1]-G[\alpha_1][u_2]\|_{\mathrm{L}^\infty(\Omega_V)} + \|G[\alpha_1]-G[\alpha_2]\|_{\mathrm{L}^\infty(\{u:\Omega_U\mapsto\mathbb{R}\,|\,\|u\|_{\mathrm{L}^\infty}\le\beta_U\}\times\Omega_V)}\\ &\le L_{\mathcal{G}}\|u_1-u_2\|_{\mathrm{L}^{r_{\mathcal{G}}}(\Omega_U)} + L_G\|\alpha_1-\alpha_2\|_{\mathrm{L}^{r_G}(\Omega_W)} \qquad (69)\\ &\le \max\{L_{\mathcal{G}}|\Omega_U|^{1/r_{\mathcal{G}}}, L_G|\Omega_W|^{1/r_G}\}\max\{\|u_1-u_2\|_{\mathrm{L}^\infty(\Omega_U)},\|\alpha_1-\alpha_2\|_{\mathrm{L}^\infty(\Omega_W)}\} \qquad (70) \end{aligned}$$

where we used Assumptions **O.1** and **O.2** for (69). We note that the condition (70) (instead of Lipschitz continuity with respect to the product norm (59)—this implies that we replace $L_f C_{\text{prod}}$ by C_{prod}) is sufficient in the proof of Proposition C.1. We therefore apply the latter (in conjunction with Remark C.2). Specifically, there exist constants

- C' depending on β_W, β_U, $L_{\mathcal{G}}$, γ_U, $r_{\mathcal{G}}$, L_G, γ_W, r_G

- C_W depending on $L_{\mathcal{G}}, \gamma_U, r_{\mathcal{G}}, L_G, \gamma_W, r_G, L_W$
- C_U depending on $L_{\mathcal{G}}, \gamma_U, r_{\mathcal{G}}, L_G, \gamma_W, r_G, L_U$

such that the following holds. For any $\varepsilon_1 > 0$,

- let $\delta_W = C_W \varepsilon_1$ and let $\{a_i\}_{i=1}^{n_{c_W}} \subset \Omega_W$ be points so that $\{\mathcal{B}_{\delta_W}(a_i)\}_{i=1}^{n_{c_W}}$ is a cover of Ω_W for some n_{c_W};
- let $\delta_U = C_U \varepsilon_1$ and let $\{c_i\}_{i=1}^{n_{c_U}} \subset \Omega_U$ be points so that $\{\mathcal{B}_{\delta_U}(c_i)\}_{i=1}^{n_{c_U}}$ is a cover of Ω_U for some n_{c_U};
- let $H = 2C'\sqrt{n_{c_W} + n_{c_U}}\varepsilon_1^{-1}$ and consider the network class $\mathcal{F}_2 := \mathcal{F}_{\mathrm{NN}}(n_{c_W}+n_{c_U}, 1, L_2, p_2, K_2, \kappa_2, R_2)$ with parameters scaling as

$$
\begin{aligned}
&L_2 = \mathcal{O}\left((n_{c_W} + n_{c_U})^2 \log(n_{c_W} + n_{c_U}) + (n_{c_W} + n_{c_U})^2[\log(\varepsilon_1^{-1}) + \log(2)]\right), \qquad p_2 = \mathcal{O}(1),\\
&K_2 = \mathcal{O}\left((n_{c_W} + n_{c_U})^2 \log(n_{c_W} + n_{c_U}) + (n_{c_W} + n_{c_U})^2[\log(\varepsilon_1^{-1}) + \log(2)]\right),\\
&\kappa_2 = \mathcal{O}((n_{c_W} + n_{c_U})^{(n_{c_W}+n_{c_U})/2+1}\varepsilon_1^{-(n_{c_W}+n_{c_U}+1)}2^{n_{c_W}+n_{c_U}+1}), \qquad R_2 = 1
\end{aligned}
$$

  where the constants hidden in $\mathcal{O}$ depend on $\beta_W, \beta_U, L_{\mathcal{G}}, \gamma_U, r_{\mathcal{G}}, L_G, \gamma_W, r_G$.

Then, there exists networks $\{b_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{F}_2$ and functions

$$
\{\alpha_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{B}_{\beta_W, \|\cdot\|_{\mathrm{L}^\infty}, \Omega_W}(0),\ \{u_k\}_{k=1}^{H^{n_{c_W}+n_{c_U}}} \subset \mathcal{B}_{\beta_U, \|\cdot\|_{\mathrm{L}^\infty}, \Omega_U}(0)
$$

such that

$$
\sup_{1\le\ell\le N^{d_V}} \sup_{\alpha\in W} \sup_{u\in U} \left| f_\ell(\alpha, u) - \sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} f_\ell(\alpha_k, u_k) b_k(\boldsymbol{\alpha}, \boldsymbol{u}) \right| \le \varepsilon_1, \tag{71}
$$

where $\boldsymbol{\alpha} = \frac{\max\{\beta_W,\beta_U\}}{\beta_W}(\alpha(a_1), \alpha(a_2), \ldots, \alpha(a_{n_{c_W}}))^\top$, $\boldsymbol{u} = \frac{\max\{\beta_W,\beta_U\}}{\beta_U}(u(c_1), u(c_2), \ldots, u(c_{n_{c_U}}))^\top$.

We continue by estimating as follows:

$$
\begin{aligned}
&\sup_{\alpha\in W} \sup_{u\in U} \sup_{x\in\Omega_V} \left| G[\alpha][u](x) - \sum_{\ell=1}^{N^{d_V}} \sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} G[\alpha_k][u_k](v_\ell) b_k(\boldsymbol{\alpha}, \boldsymbol{u}) \tau_\ell(x) \right|\\
&\le \sup_{\alpha\in W} \sup_{u\in U} \sup_{x\in\Omega_V} \left| G[\alpha][u](x) - \sum_{\ell=1}^{N^{d_V}} G[\alpha][u](v_\ell)\tau_\ell(x) \right|\\
&+ \sup_{\alpha\in W} \sup_{u\in U} \sup_{x\in\Omega_V} \left| \sum_{\ell=1}^{N^{d_V}} \left[G[\alpha][u](v_\ell)\tau_\ell(x) - \sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} G[\alpha_k][u_k](v_\ell) b_k(\boldsymbol{\alpha}, \boldsymbol{u}) \tau_\ell(x) \right] \right|\\
&\le \varepsilon_0 + \sup_{\alpha\in W} \sup_{u\in U} \sup_{x\in\Omega_V} \sum_{\ell=1}^{N^{d_V}} \tau_\ell(x) \left| G[\alpha][u](v_\ell) - \sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} G[\alpha_k][u_k](v_\ell) b_k(\boldsymbol{\alpha}, \boldsymbol{u}) \right| \qquad (72)\\
&\le \varepsilon_0 + \varepsilon_1 \sup_{x\in\Omega_V} \sum_{\ell=1}^{N^{d_V}} \tau_\ell(x) \qquad (73)
\end{aligned}
$$

where we used (68) and the fact that $0 \le \tau_\ell(x) \le 1$ for (72) and (71) for (73). For the last term, we note that for any $0 < \eta < \beta_V$, the constant function equal to η is included in V. In particular, by (68), we have

$$
\varepsilon_0 \ge \sup_{x\in\Omega_V} \left| \eta - \eta \sum_{\ell=1}^{N^{d_V}} \tau_\ell(x) \right| = \eta \sup_{x\in\Omega_V} \left| 1 - \sum_{\ell=1}^{N^{d_V}} \tau_\ell(x) \right|
$$

which implies that

$$\sum_{\ell=1}^{N^{d_V}} \tau_\ell(x) \leq 1 + \frac{\varepsilon_0}{\eta} \tag{74}$$

for all $x \in \Omega_V$. Setting $\varepsilon_0 = \frac{\varepsilon}{2}$, $\varepsilon_1 = \frac{\varepsilon}{2(1+\frac{\varepsilon}{2\eta})}$ and inserting (74) into (73) yields:

$$\sup_{\alpha\in W}\sup_{u\in U}\sup_{x\in\Omega_V}\left|G[\alpha][u](x) - \sum_{\ell=1}^{N^{d_V}}\sum_{k=1}^{H^{n_{c_W}+n_{c_U}}} G[\alpha_k][u_k](v_\ell)b_k(\boldsymbol{\alpha},\boldsymbol{u})\tau_\ell(x)\right| \leq \varepsilon.$$

The final network scalings for $\mathcal{F}_1$ are:

$$\begin{aligned}
&L_1 = \mathcal{O}\left(d_V^2 \log d_V + d_V^2(\log(\varepsilon^{-1}) + \log(2))\right), \qquad && p_1 = \mathcal{O}(1),\\
&K_1 = \mathcal{O}\left(d_V^2 \log d_V + d_V^2(\log(\varepsilon^{-1}) + \log(2))\right), \qquad && \kappa_1 = \mathcal{O}(d_V^{d_V/2+1}\varepsilon^{-(d_V+1)}2^{d_V+1}),\\
&R_1 = 1, \qquad N = 2C\sqrt{d_V}\varepsilon^{-1}.
\end{aligned}$$

Noting that $\varepsilon_1 = \frac{\varepsilon}{2(1+\frac{\varepsilon}{2\eta})} = \frac{\varepsilon\eta}{2\eta+\varepsilon} \asymp \frac{\varepsilon}{2}$ for sufficiently small ε, the final network scalings for $\mathcal{F}_2$ are:

$$\begin{aligned}
&L_2 = \mathcal{O}\left((n_{c_W}+n_{c_U})^2\log(n_{c_W}+n_{c_U}) + (n_{c_W}+n_{c_U})^2[\log(\varepsilon^{-1}) + 2\log(2)]\right), \qquad && p_2 = \mathcal{O}(1),\\
&K_2 = \mathcal{O}\left((n_{c_W}+n_{c_U})^2\log(n_{c_W}+n_{c_U}) + (n_{c_W}+n_{c_U})^2[\log(\varepsilon^{-1}) + 2\log(2)]\right),\\
&\kappa_2 = \mathcal{O}((n_{c_W}+n_{c_U})^{(n_{c_W}+n_{c_U})/2+1}\varepsilon^{-(n_{c_W}+n_{c_U}+1)}2^{2(n_{c_W}+n_{c_U}+1)}),\ R_2 = 1,\ H = 4C'\sqrt{n_{c_W}+n_{c_U}}\varepsilon^{-1}.
\end{aligned}$$

□

Proof of Theorem 3.29. 1. From the proof of Corollary 3.14, we know that W and U are compact sets of $\mathrm{L}^{r_G}(\Omega_W)$ and $\mathrm{L}^{r_G}(\Omega_U)$, respectively. With $\eta > \min\left\{1+\frac{1}{d_W}, 1+\frac{1}{d_U}\right\}$, by Lemma 3.13, at least one of W and U will contain a hypercube Q_η. Without loss of generality, we assume that W does so. By Lemma 3.23, we can embed Q_η into the product space, i.e. we obtain that $W \times \{0\} \subset W \times U$ will contain a hypercube $\tilde{Q}_\eta$. We note that $W \times U$ is also compact as product of compact spaces.

Let $r \in \mathbb{N}$ and $\delta > 0$. By the above, all the conditions to apply [26, Corollary 2.12] on the product Banach space $\mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U)$ are satisfied. In particular, we deduce the existence of $\bar{\varepsilon}$ and $c > 0$ such that for $0 < \varepsilon \leq \bar{\varepsilon}$ and any operator of neural network type NN_ε satisfying (23), we have $\mathcal{C}(\mathrm{NN}_\varepsilon) \geq \exp(c\varepsilon^{-1/(\eta+1+\delta)r})$.

We conclude by noting that any network of the form (18) is an operator of neural network type on the product space $\mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U)$ by Lemma 3.21. The latter also implies that

$$\|\Theta\|_0 + HK_2 + NK_1 \gtrsim \mathcal{C}(\mathrm{NN}_\varepsilon) \geq \exp(c\varepsilon^{-1/(\eta+1+\delta)r}).$$

2. From part 1, we may take $r = 1$ to obtain $G : \mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U) \to V$ which is Frechet differentiable on $\mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U)$. Specifically, from the proof of [26, Corollary 2.12], we know that

$$G[\alpha][u](x) = F(\alpha, u)\phi(x),$$

where $F : \mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U) \to \mathbb{R}$ is the Frechet differentiable functional provided by [26, Theorem 2.11], and $\phi \in V$ is a fixed nontrivial function.

Next, the proof of [26, Lemma A.7] shows that

$$\sup_{\alpha\in\mathrm{L}^{r_G}(\Omega_W)}\sup_{u\in\mathrm{L}^{r_G}(\Omega_U)} \|DF(\alpha,u)\|_{(\mathrm{L}^{r_G}(\Omega_W)\times\mathrm{L}^{r_G}(\Omega_U))^*} < \infty.$$

Hence, F is Lipschitz on $\mathrm{L}^{r_G}(\Omega_W) \times \mathrm{L}^{r_G}(\Omega_U)$ with Lipschitz constant $\mathrm{Lip}(F)$. It remains to deduce the two required Lipschitz properties. First, for fixed α and any u_1, u_2, we have

$$\|G[\alpha][u_1] - G[\alpha][u_2]\|_{\mathrm{L}^\infty(\Omega_V)} = \|F(\alpha,u_1)\phi - F(\alpha,u_2)\phi\|_{\mathrm{L}^\infty(\Omega_V)}$$

$$
\begin{aligned}
&\leq \mathrm{Lip}(F)\|\phi\|_{\mathrm{L}^\infty(\Omega_V)}\|(\alpha,u_1)-(\alpha,u_2)\|_{\mathrm{L}^{r_G}(\Omega_W)\times\mathrm{L}^{r_G}(\Omega_U)} \\
&\leq C_{\text{prod}}\mathrm{Lip}(F)\beta_V\|u_1-u_2\|_{\mathrm{L}^{r_G}(\Omega_U)} \qquad (75)
\end{aligned}
$$

where we used Assumption **N.2** for (75). We can therefore define $L_{\mathcal{G}} = C_{\text{prod}}\mathrm{Lip}(F)\beta_V > 0$ and $r_{\mathcal{G}} = r_G$. Second, for α_1, α_2 and any u, we estimate as follows:

$$
\begin{aligned}
\|G[\alpha_1]-G[\alpha_2]\|_{\mathrm{L}^\infty(\{u:\|u\|_{\mathrm{L}^\infty(\Omega_U)}\leq\beta_U\}\times\Omega_V)} &= \sup_{u\in U}|F(\alpha_1,u)-F(\alpha_2,u)|\,\|\phi\|_{\mathrm{L}^\infty(\Omega_V)} \\
&\leq \mathrm{Lip}(F)\beta_V\|(\alpha_1,u)-(\alpha_2,u)\|_{\mathrm{L}^{r_G}(\Omega_W)\times\mathrm{L}^{r_G}(\Omega_U)} \\
&\leq \mathrm{Lip}(F)C_{\text{prod}}\beta_V\|\alpha_1-\alpha_2\|_{\mathrm{L}^{r_G}(\Omega_W)} \qquad (76) \\
&=: L_G\|\alpha_1-\alpha_2\|_{\mathrm{L}^{r_G}(\Omega_W)}
\end{aligned}
$$

where we used Assumption **N.2** for (76). This concludes the proof.

3. The lower bound in (16) is given by combining parts 1 and 2 of the theorem. The upper bound is a direct consequence of Proposition 3.25 and Remark 3.26.

□