Infinite Neural Operators: Gaussian processes on functions

Daniel Augusto de Souza[*]
University College London

Yuchen Zhu
University College London

Harry Jake Cunningham
University College London

Yuri Saporito
Fundação Getulio Vargas

Diego Mesquita
Fundação Getulio Vargas

Marc Peter Deisenroth
University College London

Abstract

A variety of infinitely wide neural architectures (e.g., dense NNs, CNNs, and transformers) induce Gaussian process (GP) priors over their outputs. These relationships provide both an accurate characterization of the prior predictive distribution and enable the use of GP machinery to improve the uncertainty quantification of deep neural networks. In this work, we extend this connection to neural operators (NOs), a class of models designed to learn mappings between function spaces. Specifically, we show conditions for when arbitrary-depth NOs with Gaussian-distributed convolution kernels converge to function-valued GPs. Based on this result, we show how to compute the covariance functions of these NO-GPs for two NO parametrizations, including the popular Fourier neural operator (FNO). With this, we compute the posteriors of these GPs in regression scenarios, including PDE solution operators. This work is an important step towards uncovering the inductive biases of current FNO architectures and opens a path to incorporate novel inductive biases for use in kernel-based operator learning methods.

1 Introduction

Neural Operators (NOs, Kovachki et al., 2023) are deep learning architectures designed to learn mappings between function spaces—with direct applications in many areas of science and engineering (Pathak et al., 2022; Li et al., 2024). NOs generalize conventional convolutional neural networks using *kernel integral operators*, which integrate the input function against a learnable kernel at each layer. Importantly, unlike CNNs, NOs can be trained with inputs of mixed, arbitrary resolutions and output predictions in discretizations of arbitrary granularity.

Despite their growing adoption, most works on NOs are primarily empirical, and most of the theoretical properties of NOs are still unexplored. In contrast, the convergence of Bayesian neural networks to Gaussian processes as their width goes to infinity has been amply studied (Neal, 1995; Novak et al., 2019; Yang, 2019). However, due to the infinite dimensionality of function spaces, it is unclear whether GPs are a limiting case for NOs and, if this is the case, how to characterize them.

In this work, we elucidate this question and present a set of assumptions that guarantee the existence of the infinite limit of NOs as Gaussian elements in the space of operators. Additionally, we present how to derive the covariance function for infinite-width NOs in an analogous fashion to the covariance functions of infinitely wide, densely connected NNs. Finally, we characterize the infinite-width limit of Fourier neural operators (FNOs) and propose a novel Bayesian NO architecture based on Matérn GP-distributed integral kernels.

[*] Corresponding author: daniel.souza.21@ucl.ac.uk

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Our experiments reinforce our theoretical results, showcasing the agreement between increasingly wide NOs and our derived expressions for the infinite limit at initialization. Additionally, we compare the performance of these models in a regression setting.

2 Background

This section provides a brief background on NOs (Section 2.1), along with basic notions of probability in Hilbert spaces (Section 2.2) and Gaussian processes on functions.

2.1 Operator learning and neural operators

Kovachki et al. (2023) propose neural operators, a family of parametrized operators. Recall that multilayer perceptrons transform vectors using successive layers of sums of linear transformations followed by element-wise non-linear activation functions. Analogously, Kovachki et al. (2023) define the building layers of neural operators (NOs) as sums of both point-wise linear operations and kernel integral operators, possibly followed by point-wise element-wise non-linear activation functions.

Well-defined dot products in function spaces are central to coherently defining NOs. Thus, we will often assume functions lie in a vector space in which their dot product is finite wrt some measure $\mu_{\mathcal{X}}$ over their domain $\mathcal{X}$. We define the Lebesgue space $L^2\big(\mathcal{X}, \mu_{\mathcal{X}}; \mathbb{R}^d\big)$ as the equivalence classes of functions in this vector space that agree almost everywhere in $\mathcal{X}$ with respect to $\mu_{\mathcal{X}}$. When clear from context, we will simply denote this vector space by $L^2(\mathcal{X})$. Whenever needed to evaluate functions point-wise, we further assume the function lies in an appropriate Reproducing Kernel Hilbert Space (RKHSs). In this work, we will be using both the Lebesgue space $L^2(\mathcal{X})$ and RKHSs, when adequate.

Point-wise operators. These operations are carried over from standard neural networks. Thus, given a function $\mathbf{f}\colon \mathcal{X} \to \mathbb{R}^d$, we consider dense layer-operations, with parameters $\boldsymbol{W} \in \mathbb{R}^{b\times d}$, defined as $(\boldsymbol{W}\mathbf{f})(\boldsymbol{x})\colon \mathcal{X} \to \mathbb{R}^b \coloneqq \boldsymbol{W}\mathbf{f}(\boldsymbol{x})$, and element-wise activations, with a given $\sigma\colon \mathbb{R} \to \mathbb{R}$, to be defined as $\sigma[\mathbf{f}]_j(\boldsymbol{x}) \coloneqq \sigma(\mathrm{f}_j(\boldsymbol{x}))$. By composing and adding results between layers, we can build neural operators that basically act just on the output of the functions.

Kernel integral operator. The majority of interesting behaviors require expanding the receptive field and aggregate results from different function evaluations into one. The *kernel integral operator* $A_{\mathbf{K}}\colon (\mathcal{X} \to \mathbb{R}^d) \to (\mathcal{Y} \to \mathbb{R}^b)$, parametrized by a matrix-valued kernel function $\mathbf{K} : \mathcal{Y}\times\mathcal{X} \to \mathbb{R}^{b\times d}$ together with a measure $\mu_{\mathcal{X}}$ on $\mathcal{X}$, is defined as:

$$A_{\mathbf{K}}[\mathbf{f}](\boldsymbol{y})\colon \mathcal{Y} \to \mathbb{R}^b = \int_{\mathcal{X}} \mathbf{K}(\boldsymbol{y}, \boldsymbol{x})\, \mathbf{f}(\boldsymbol{x})\, \mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}). \tag{1}$$

Under this operation, the function evaluated at a single evaluating point $\boldsymbol{y}$ linearly aggregates information on all evaluating points in the domain $\mathcal{X}$ as modulated by the kernel $\mathbf{K}$ and the measure $\mu_{\mathcal{X}}$. Note, that this function may not converge for all values of $\boldsymbol{y}$, but, for any kernel $\mathbf{K} \in L^2\big(\mathcal{Y}\times\mathcal{X}, \mu_{\mathcal{Y}}\times\mu_{\mathcal{X}}; \mathbb{R}^{b\times d}\big)$, the operator $A_{\mathbf{K}}\colon L^2(\mathcal{X}, \mu_{\mathcal{X}}) \to L^2(\mathcal{Y}, \mu_{\mathcal{Y}})$ is well-defined.

Constructing neural operators. Given these building blocks, Kovachki et al. (2023) describe a neural operator as a three-part layered model. Firstly, a sequence of point-wise operators are applied to pre-process the function and change the dimension of its output. This is called the *Lift layer*. The second component is a combination of point-wise and kernel integral operators, in the so-called *Neural Operator layer*. Finally, the *Projection layer*, a sequence of point-wise operators is applied to the final result.

Specifically, a neural operator layer combines a matrix-valued kernel $\mathbf{K}$ and matrix $\boldsymbol{W}$ into

$$H[\mathbf{f}](\boldsymbol{x})\colon \mathcal{X} \to \mathbb{R}^h = A_{\mathbf{K}}[\mathbf{f}](\boldsymbol{x}) + \boldsymbol{W}\mathbf{f}(\boldsymbol{x}) = \int_{\mathcal{X}} \mathbf{K}(\boldsymbol{x}, \boldsymbol{z})\, \mathbf{f}(\boldsymbol{z})\, \mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{z}) + \boldsymbol{W}\mathbf{f}(\boldsymbol{x}). \tag{2}$$

Setting the matrix-valued kernel to zero recovers the lift and projection layers. Therefore, a neural operator with depth d and scalar output can be written succinctly as the composition:

$$Z[\mathbf{f}](\boldsymbol{x})\colon \mathcal{X} \to \mathbb{R} = (\boldsymbol{w}^{\intercal} \circ \sigma \circ H_d \circ \cdots \sigma \circ H_1)[\mathbf{f}](\boldsymbol{x}). \tag{3}$$

2.2 Probability in Hilbert spaces

Given a probability space $(\Omega, \Sigma, \mathbb{P})$, and a Hilbert space $\mathcal{H}$, random elements in $\mathcal{H}$ are functions $x\colon \Omega \to \mathcal{H}$, such that the inner product $\omega \in \Omega \mapsto \langle y, x(\omega)\rangle_{\mathcal{H}}$ is a real-valued random variable, for any $y \in \mathcal{H}$. As usual, we follow the standard notation of denoting the random elements/variables not as functions x but as elements x. Likewise, expectation is defined in terms of the random variables $\langle y, x(\omega)\rangle$, for each $y \in \mathcal{H}$. We say that the expectation of x, when it exists, is the element of $\mathcal{H}$, denoted by $\mathbb{E}[x]$, such that $\mathbb{E}[\langle y, x\rangle] = \langle y, \mathbb{E}[x]\rangle$, for any $y \in \mathcal{H}$.

We denote the space of Hilbert–Schmidt (HS) operators mapping elements from a Hilbert space $\mathcal{A}$ to $\mathcal{B}$ by $\mathrm{HS}(\mathcal{A};\mathcal{B})$. This space is the completion of the span of rank-one operators of the form $a \otimes b \colon \mathcal{A} \to \mathcal{B}$, defined as $(a \otimes b)(x) = \langle x, a\rangle_{\mathcal{A}}\, b$ for all $a \in \mathcal{A}$ and $b \in \mathcal{B}$. For L^2 spaces, we have the isomorphism $\mathrm{HS}\big(L^2(\mathcal{X};\mathbb{R}^d), L^2(\mathcal{Y};\mathbb{R}^b)\big) \cong L^2\big(\mathcal{X}\times\mathcal{Y};\mathbb{R}^{b\times d}\big)$, under which $(\mathbf{f}\otimes\mathbf{g})[\mathbf{h}](\cdot) = \int_{\mathcal{X}} \mathbf{g}\mathbf{f}^{\intercal}(\cdot,\boldsymbol{x})\mathbf{h}(\boldsymbol{x})\,\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x})$, where $\mathbf{f} \in L^2\big(\mathcal{X};\mathbb{R}^d\big)$, $\mathbf{g} \in L^2\big(\mathcal{Y};\mathbb{R}^b\big)$, and $\mathbf{g}\mathbf{f}^{\intercal} \in L^2\big(\mathcal{X}\times\mathcal{Y};\mathbb{R}^{b\times d}\big)$.

Moreover, the (cross-)covariance operator of two centered variables x and y is defined as the expectation of the tensor product $\mathbb{E}[x \otimes y]$. When this expectation exists, it is also a HS operator denoted as $\mathrm{Cov}(x,y)$. From these definitions, we have that $\langle z_2, \mathrm{Cov}(x,y)[z_1]\rangle = \mathrm{cov}(\langle z_2, y\rangle\langle z_1, x\rangle)$, for any $z_1, z_2 \in \mathcal{H}$. In L^2 spaces, we will make use of the isomorphism above and represent the covariance operator by its integration kernel. So, for any random elements $\mathbf{f} \in L^2\big(\mathcal{X};\mathbb{R}^d\big)$ and $\mathbf{g} \in L^2\big(\mathcal{Y};\mathbb{R}^b\big)$, we introduce the function $\mathbf{C}[\mathbf{f},\mathbf{g}] \colon \mathcal{X}\times\mathcal{Y} \to \mathbb{R}^{b\times d}$ such that $\mathrm{Cov}(\mathbf{f},\mathbf{g})[\mathbf{h}](\cdot) = \int_{\mathcal{X}} \mathbf{C}[\mathbf{f},\mathbf{g}](\cdot,\boldsymbol{x})\mathbf{h}(\boldsymbol{x})\,\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x})$.

In this work, we will make use of an extension of the strong law of large numbers to random elements:

Theorem 2.1 (Strong law of large numbers (Mourier, 1956))**.** *Let $\mathcal{H}$ be a separable Hilbert space and $\{x_j\}_{j\in\mathbb{N}}$ be a countable sequence of identically distributed random elements. Consider the sample average $y_N = (1/N)\sum_{j=1}^{N} x_j$. If, for any j, the expected norm $\mathbb{E}[\|x_j\|]$ exists, then, the sequence $\{y_N\}_{N\in\mathbb{N}}$ converges almost surely to the constant random element $y_\infty = \mathbb{E}[x_j]$.*

2.3 Operator valued kernels and Hilbert space valued Gaussian processes

Now, given a set $\mathcal{X}$ and a separable Hilbert space $\mathcal{H}$, an *operator-valued valued kernel* $\mathrm{C}\colon \mathcal{X}\times\mathcal{X} \to \mathrm{HS}(\mathcal{H};\mathcal{H})$ is any Hermitian positive-definite function, i.e., for all $\boldsymbol{x},\boldsymbol{x}' \in \mathcal{X}$, $\mathrm{C}(\boldsymbol{x},\boldsymbol{x}') = \mathrm{C}(\boldsymbol{x}',\boldsymbol{x})^{\intercal}$, and, for any $n > 0$, $\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^{n} \subset \mathcal{X}\times\mathcal{H}$ and $\{\alpha_{ij}\}_{i,j=1}^{n} \subset \mathbb{R}$, we have that $\sum_{i,j=1}^{n} \alpha_{ij}\langle \boldsymbol{y}_j, \mathrm{C}(\boldsymbol{x}_i,\boldsymbol{x}_j)[\boldsymbol{y}_i]\rangle > 0$ (Kadri et al., 2016).x

Consider an operator-valued kernel $\mathrm{C}\colon \mathcal{X}\times\mathcal{X} \to \mathrm{HS}(\mathcal{H};\mathcal{H})$ such that $\boldsymbol{x} \mapsto \mathrm{C}(\boldsymbol{x},\boldsymbol{x})$ is of trace-class. We say $\mathrm{f} : \mathcal{X}\times\Omega \to \mathcal{H}$ is a centered Gaussian process with covariance function C if, for any $n > 0$ and $\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^{n} \subset \mathcal{X}\times\mathcal{H}$, the vector $(\langle y_1, \mathrm{f}(\boldsymbol{x}_1,\cdot)\rangle, \ldots, \langle y_n, \mathrm{f}(\boldsymbol{x}_n,\cdot)\rangle)$ is a random element distributed as an n-dimensional Gaussian with covariance

$$\mathbb{E}[\langle y_2, \mathrm{f}(\boldsymbol{x}_2,\cdot)\rangle\, \langle y_1, \mathrm{f}(\boldsymbol{x}_1,\cdot)\rangle] = \langle y_2, \mathrm{C}(\boldsymbol{x}_1,\boldsymbol{x}_2)[y_1]\rangle. \tag{4}$$

We denote this by $\mathrm{f} \sim \mathrm{GP}(0,\mathrm{C})$. For simplicity, we also define $\mathrm{f}(\boldsymbol{x}) \coloneqq \mathrm{f}(\boldsymbol{x},\cdot)$.

3 Infinite-width neural operators as Gaussian processes

It is well known that infinite-width limits of various Bayesian neural networks are Gaussian processes (Neal, 1995; Matthews et al., 2018). We generalize this connection and show that infinite-width neural operators are function-valued Gaussian processes.

Analogous to Novak et al. (2019), who place Gaussian priors on the convolution kernels of a CNN, the natural step towards function-valued GPs is to put independent GP priors on the component operators. Similarly, we require the weights and kernel for any component operator to be i.i.d. and with covariance shrinking with width. Theorem 3.1 states the main result of this work.

Theorem 3.1 (Infinite-width neural operators are Gaussian processes)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ be a measurable space and let $\mathcal{H}\big(\mathcal{X};\mathbb{R}^J\big) \subset L^2\big(\mathcal{X};\mathbb{R}^J\big)$ be an RKHS for any $J \in \mathbb{N}^+$. Then, for a given depth $D \in \mathbb{N}^+$, consider a vector of positive integers $\boldsymbol{J} = [J_0, J_1, \ldots, J_{D-1}, 1]^{\intercal} \in \mathbb{N}^{D+1}$ and a $\boldsymbol{J}$-indexed neural operators $Z_{\boldsymbol{J}}^{(D)}$ of depth D:*

$$Z_{\boldsymbol{J}}^{(D)} \coloneqq H^{(D)} \circ \sigma \circ Z_{\boldsymbol{J}}^{(D-1)} \in (\mathcal{X} \to \mathbb{R}^{J_0}) \to (\mathcal{X} \to \mathbb{R}), \tag{5}$$

where,

$$Z_{\boldsymbol{J}}^{(1)} \coloneqq H^{(1)} \in L^2\big(\mathcal{X};\mathbb{R}^{J_0}\big) \to \mathcal{H}\big(\mathcal{X};\mathbb{R}^{J_1}\big), \tag{6}$$

$$\begin{aligned} H^{(\ell)} &:= (A_{\mathbf{K}^{(\ell)}} + \boldsymbol{W}^{(\ell)}) \in L^2\big(\mathcal{X}; \mathbb{R}^{J_{\ell-1}}\big) \to \mathcal{H}\big(\mathcal{X}; \mathbb{R}^{J_\ell}\big), && (7) \\ \boldsymbol{W}^{(\ell)} &\in \mathbb{R}^{J_\ell \times J_{\ell-1}}, && (8) \\ \mathbf{K}^{(\ell)} &\in \mathcal{H}\big(\mathcal{X} \times \mathcal{X}; \mathbb{R}^{J_\ell \times J_{\ell-1}}\big), && (9) \end{aligned}$$

and $\sigma \colon \mathbb{R} \to \mathbb{R}$ *such that* $(\sigma \circ \mathrm{f}) \in L^2(\mathcal{X})$ *for any* $\mathrm{f} \in L^2(\mathcal{X})$.

When all parameters are independently distributed a priori *according to*

$$\boldsymbol{W}^{(\ell)} \sim \mathcal{N}\big(\mathbf{0}, \sigma_\ell^2/J_{\ell-1}\boldsymbol{I}\big), \text{ and } \mathbf{K}^{(\ell)} \sim \mathrm{GP}(\mathbf{0}, \mathrm{c}_{\mathrm{k}^{(\ell)}}/J_{\ell-1}\boldsymbol{I}), \qquad \text{for } \ell \in \{1, \dots, d\}, \tag{10}$$

then, the iterated limit $\lim\limits_{J_{D-1}\to\infty} \cdots \lim\limits_{J_1\to\infty} Z_{\boldsymbol{J}}^{(D)}$*, in the sense of Definition B.1, is equal to a function-valued GP* $Z_\infty^{(D)} \sim \mathrm{GP}(0, \mathrm{c}_\infty)$*, where* $\mathrm{c}_\infty[\mathbf{f}, \mathbf{g}]$ *is available in closed-form.*

An outline of the proof is presented in Section 3.2, where we present the explicit formula for c_∞, which depends on the conditional covariance function between layers. Before delving into these details, we introduce the compositionality property of covariance functions in Section 3.1. This property enables the closed-form computation of the conditional covariances, thereby fully characterizing the limiting covariance function c_∞.

3.1 Operator-valued covariance functions

We realize the following crucial points: i) The covariance function only depends on the inner product of the values of the input functions, and ii) Using the strong law of large numbers, the covariance of the composition of operators can be described by composing its covariance functions. This is presented in the next lemma, with proof postponed to Appendix B.2.

Lemma 3.2 (Compositionality of covariance functions)**.** *Let* $B_1 \colon L^2\big(\mathcal{X}; \mathbb{R}^d\big) \to L^2\big(\mathcal{X}; \mathbb{R}^J\big)$ *be a random operator and* $B_2 \colon L^2\big(\mathcal{X}; \mathbb{R}^J\big) \to L^2(\mathcal{X})$ *be a centered function-valued Gaussian process. If the following assumptions hold:*

- *For all* $\mathbf{f} \in L^2\big(\mathcal{X}; \mathbb{R}^d\big)$ *and* $\boldsymbol{x} \in \mathcal{X}$*, each component of* $B_1[\mathbf{f}](\boldsymbol{x}) \in \mathbb{R}^J$ *is independent and identically distributed such that the covariance function* $\mathbf{C}_{B_1}[\mathbf{f}, \mathbf{g}] = \mathrm{c}_{B_1}[\mathbf{f}, \mathbf{g}]\boldsymbol{I}_J$*;*
- *The covariance function of* B_2 *can be expressed, for all* $\mathbf{f}, \mathbf{g} \in L^2\big(\mathcal{X}; \mathbb{R}^J\big)$ *as* $\mathrm{c}_{B_2}[\mathbf{f}, \mathbf{g}] = \mathrm{c}_{B_2}\big[\frac{1}{J}\mathbf{g}^\intercal\mathbf{f}\big]$ *and the function* $\mathrm{h} \mapsto \mathrm{c}_{B_2}[\mathrm{h}]$ *is a continuous map from* $L^2(\mathcal{X} \times \mathcal{X})$ *to itself.*

Then, $B_2 \circ B_1$ *converges in distribution to a function-valued Gaussian process as* $J \to \infty$*, and*

$$\mathrm{c}_{B_2 \circ B_1}[\mathbf{f}_1, \mathbf{f}_2] = \mathrm{c}_{B_2}[\mathrm{c}_{B_1}[\mathbf{f}_1, \mathbf{f}_2]]. \tag{11}$$

For each operator discussed in Section 2.1, below we state the conditions under which they are function-valued Gaussian processes, and derive their covariance functions.

Point-wise linear operator. Given a vector $\boldsymbol{w} \in \mathbb{R}^d$ and a function $\mathbf{f} \colon \mathcal{X} \to \mathbb{R}^d$, then, define the linear operator $(\boldsymbol{w}^\intercal\mathbf{f}) \colon \mathcal{X} \to \mathbb{R}$ such that $(\boldsymbol{w}^\intercal\mathbf{f})(\boldsymbol{x}) = \sum_{p=1}^d w_p \mathrm{f}_p(\boldsymbol{x})$. If the entries of the weight vector follow an i.i.d. Gaussian distribution, i.e. $\boldsymbol{w} \sim \mathcal{N}\big(\mathbf{0}, \sigma^2\boldsymbol{I}\big)$, then, this is a centered Gaussian process taking values from $L^2\big(\mathcal{X}; \mathbb{R}^d\big)$ to $L^2(\mathcal{X}; \mathbb{R})$ with covariance function:

$$\mathrm{c}_{\boldsymbol{w}}[\mathbf{f}_1, \mathbf{f}_2](\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma^2\, \mathbf{f}_2^\intercal(\boldsymbol{x}_2)\, \mathbf{f}_1(\boldsymbol{x}_1), \text{ such that,} \tag{12}$$

$$\mathrm{cov}(\langle \mathrm{h}_2, \boldsymbol{w}^\intercal\mathbf{f}_2\rangle \, \langle \mathrm{h}_1, \boldsymbol{w}^\intercal\mathbf{f}_1\rangle) = \iint_{\mathcal{X}} \mathrm{h}_2(\boldsymbol{x}_2)\, \mathrm{h}_1(\boldsymbol{x}_1)\, \mathrm{c}_{\boldsymbol{w}}[\mathbf{f}_1, \mathbf{f}_2](\boldsymbol{x}_1, \boldsymbol{x}_2)\, \mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}_1)\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}_2). \tag{13}$$

Note that $\mathrm{c}_{\boldsymbol{w}}$ only depends on the function $\mathbf{f}_2^\intercal\mathbf{f}_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, so we abuse notation and write $\mathrm{c}_{\boldsymbol{w}}[\mathbf{f}_1, \mathbf{f}_2] = \mathrm{c}_{\boldsymbol{w}}[\mathbf{f}_2^\intercal\mathbf{f}_1]$. Moreover, this function is homogeneous: $\alpha\mathrm{c}_{\boldsymbol{w}}[\mathbf{f}_2^\intercal\mathbf{f}_1] = \mathrm{c}_{\boldsymbol{w}}[\alpha\mathbf{f}_2^\intercal\mathbf{f}_1]$, for $\alpha > 0$.

Kernel integral operator. As defined in Section 2.1, given a function $\mathbf{k} \colon \mathcal{Y} \times \mathcal{X} \to \mathbb{R}^d$ and an input function $\mathbf{f} \colon \mathcal{X} \to \mathbb{R}^d$, we consider the linear operator $A_{\mathbf{k}^\intercal}[\mathbf{f}] \colon \mathcal{Y} \to \mathbb{R}$. If $\mathbf{k}$ follows an i.i.d. GP such that $\mathbf{k} \in L^2(\mathcal{Y} \times \mathcal{X}) \sim \mathrm{GP}(0, \mathrm{c}_{\mathbf{k}})$, then we have that $A_{\mathbf{k}^\intercal}$ is a centered function-valued GP mapping from $L^2\big(\mathcal{X}; \mathbb{R}^d\big)$ to $L^2(\mathcal{Y})$ with covariance function, denoted here by:

$$\mathrm{c}_{A_{\mathbf{k}^\intercal}}[\mathbf{f}_1, \mathbf{f}_2](\boldsymbol{y}_1, \boldsymbol{y}_2) = \iint_{\mathcal{X}} \mathrm{c}_{\mathbf{k}}(\boldsymbol{y}_1, \boldsymbol{x}_1, \boldsymbol{y}_2, \boldsymbol{x}_2)\, \mathbf{f}_2^\intercal(\boldsymbol{x}_2)\, \mathbf{f}_1(\boldsymbol{x}_1)\, \mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}_1)\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}_2) \tag{14}$$

$$= A_{\mathrm{c}_{\mathbf{k}}}[\mathbf{f}_2^\intercal \mathbf{f}_1](\boldsymbol{y}_1, \boldsymbol{y}_2). \tag{15}$$

Note $\mathrm{c}_{A_{\mathbf{k}\intercal}}$ also only depends on the inner product of $\mathbf{f}_2$ and $\mathbf{f}_1$ and is homogeneous. Thus, we denote $\mathrm{c}_{A_{\mathbf{k}\intercal}}[\mathbf{f}_1, \mathbf{f}_2] = \mathrm{c}_{A_{\mathbf{k}\intercal}}[\mathbf{f}_2^\intercal \mathbf{f}_1]$.

Point-wise element-wise activation. Given a non-linear function $\sigma\colon \mathbb{R} \to \mathbb{R}$, we abuse the notation and define the non-linear operator $\sigma[\cdot]\colon L^2(\mathcal{X}) \to L^2(\mathcal{X})$ as

$$\sigma[\mathrm{f}](\boldsymbol{x}) = \sigma(\mathrm{f}(\boldsymbol{x})). \tag{16}$$

Note that some restrictions on σ need to be placed for this to be a well-defined operator in $L^2(\mathcal{X})$. As an example of such condition, for their theoretical analysis, Kovachki et al. (2023) restricts activations to measurable linearly bounded functions, noting that the popular ReLU, ELU, tanh, and sigmoid activations satisfy this condition. In Appendix B.1, we provide a proof that this condition is sufficient for finite measure domains.

Consider a centered Gaussian operator $B\colon L^2(\mathcal{X}) \to \mathcal{H}(\mathcal{X})$ with covariance function c_B such that $\mathcal{H}(\mathcal{X}) \subset L^2(\mathcal{X})$ is an RKHS with reproducing kernel $\mathrm{k}_{\mathcal{H}}$. When $\sigma[\cdot]$ is a well-defined operator, the operator $(\sigma \circ B)$ is a random operator in $L^2(\mathcal{X}) \to L^2(\mathcal{X})$ with covariance function:

$$\mathrm{c}_{(\sigma\circ B)}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \operatorname{cov}((\sigma \circ B)[\mathrm{f}_1](\boldsymbol{x}_1),\, (\sigma \circ B)[\mathrm{f}_2](\boldsymbol{x}_2)). \tag{17}$$

Now, since $B[\mathrm{f}_1]$ and $B[\mathrm{f}_2]$ are Gaussian processes with outputs in an RKHS $\mathcal{H}$, we can consider the following bivariate Gaussian r.v. $\boldsymbol{b}_{[\mathrm{f}_1,\mathrm{f}_2]} = [B[\mathrm{f}_1](\boldsymbol{x}_1), B[\mathrm{f}_2](\boldsymbol{x}_2)]^\intercal$:

$$\boldsymbol{b}_{[\mathrm{f}_1,\mathrm{f}_2]} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathrm{c}_B[\mathrm{f}_1,\mathrm{f}_1](\boldsymbol{x}_1,\boldsymbol{x}_1) & \mathrm{c}_B[\mathrm{f}_1,\mathrm{f}_2](\boldsymbol{x}_1,\boldsymbol{x}_2) \\ \mathrm{c}_B[\mathrm{f}_2,\mathrm{f}_1](\boldsymbol{x}_2,\boldsymbol{x}_1) & \mathrm{c}_B[\mathrm{f}_2,\mathrm{f}_2](\boldsymbol{x}_2,\boldsymbol{x}_2) \end{bmatrix} \right). \tag{18}$$

This random variable is well-defined due to the reproducing property, $B[\mathrm{f}](\boldsymbol{x}) = \langle \mathrm{k}_{\mathcal{H}}(\cdot, \boldsymbol{x}), B[\mathrm{f}] \rangle$.

Thus, we can continue to conclude

$$\begin{aligned} \mathrm{c}_{(\sigma\circ B)}(\boldsymbol{x}_1, \boldsymbol{x}_2) &= \int_{\mathbb{R}^2} \sigma(b_{\mathrm{f}_1})\, \sigma(b_{\mathrm{f}_2})\ \mathcal{N}\big(\boldsymbol{b}_{[\mathrm{f}_1,\mathrm{f}_2]} \mid \mathbf{0}, \boldsymbol{L}^\intercal \boldsymbol{L}\big) \mathrm{d}\boldsymbol{b}_{[\mathrm{f}_1,\mathrm{f}_2]} && (19) \\ &= \int_{\mathbb{R}^2} \sigma(\boldsymbol{l}_1^\intercal \boldsymbol{\xi})\, \sigma(\boldsymbol{l}_2^\intercal \boldsymbol{\xi})\ \mathcal{N}(\boldsymbol{\xi} \mid \mathbf{0}, \boldsymbol{I}) \mathrm{d}\boldsymbol{\xi} && (20) \\ &=: \mathrm{c}_\sigma[\mathrm{c}_B[\mathrm{f}_1, \mathrm{f}_2]](\boldsymbol{x}_1, \boldsymbol{x}_2), && (21) \end{aligned}$$

where $\boldsymbol{L}$ is a square-root of the covariance matrix of $\boldsymbol{b}_{[\mathrm{f}_1,\mathrm{f}_2]}$ and $\boldsymbol{l}_i$ is the i-th row of this matrix.

The expected value c_σ as a function of $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ in Eq. (20) is known as the *dual kernel* of σ. The dual kernels for many activation functions have closed-form solutions (e.g., sigmoid (Williams, 1996, Eq. 10) and ReLU (Cho and Saul, 2009, Eq. 1)) or can be efficiently approximated (Han et al., 2022). Any of these solutions can be directly used in our context by computing the covariance matrix of $\boldsymbol{b}_{[\mathrm{f}_1,\mathrm{f}_2]}$ and applying the rows of its square-root as arguments.

In conclusion, we construct an covariance function $\mathrm{c}_\sigma\colon \mathcal{H}(\mathcal{X} \times \mathcal{X}) \to L^2(\mathcal{X} \times \mathcal{X})$ such that, for a given covariance function $\mathrm{c}_B\colon L^2(\mathcal{X} \times \mathcal{X}) \to \mathcal{H}(\mathcal{X} \times \mathcal{X})$:

$$\langle \mathrm{h}_2 \mathrm{h}_1, \mathrm{c}_\sigma[\mathrm{c}_B[\mathrm{f}_1, \mathrm{f}_2]] \rangle = \operatorname{cov}(\langle \mathrm{h}_1, (\sigma \circ B)[\mathrm{f}_1] \rangle, \langle \mathrm{h}_2, (\sigma \circ B)[\mathrm{f}_2] \rangle), \tag{22}$$

for all $\mathrm{f}_1, \mathrm{f}_2, \mathrm{h}_1, \mathrm{h}_2 \in L^2(\mathcal{X})$

3.2 Outline of the proof for Theorem 3.1

We now describe a sketch for the proof, we refer the readers to Appendix B.3 for the complete proof.

Step 1. We start by showing that, under the conditions of Theorem 3.1, each linear layer in a neural operator layer is a function-valued Gaussian process when conditioned on its inputs. Moreover, as discussed in Section 3.1, the conditional covariance function of each node on each layer only depends on the *empirical covariance function* of its inputs $\bar{\mathrm{c}}[\mathbf{f}, \mathbf{g}](\boldsymbol{x}', \boldsymbol{x}) = (1/J) \sum_{j=1}^{J} \mathrm{g}_j(\boldsymbol{x})\, \mathrm{f}_j(\boldsymbol{x}')$. We denote this dependency by writing the conditional covariance function as $\mathrm{c}^{(\ell|\ell-1)}[\cdot](\boldsymbol{x}, \boldsymbol{x}')$.

Step 2. Due to the chosen prior distribution of each layer, we know that each node in $H_\ell[\cdot] \in \mathbb{R}^{J_\ell}$ is i.i.d. and, therefore, we can apply Lemma 3.2 to conclude that, as $J_{\ell-1} \to \infty$, the covariance $\mathrm{c}^{(\ell|\ell-1)}[H_{\ell-1}[\mathbf{g}]^\intercal H_{\ell-1}[\mathbf{f}]/J_{\ell-1}]$ converges almost surely to $\mathrm{c}^{(\ell|\ell-1)}[\mathrm{c}_{H_{\ell-1}}[\mathbf{f}, \mathbf{g}]]$.

Step 3. Combining both steps, we show, by induction on ℓ up until $\ell = d$, that, as $J \to \infty$, the covariance function of $Z_J[\mathbf{f}]$ is simply the composition of all the previous covariances as denoted in Step 1. So, we have that the covariance function of Z_∞ is:

$$\mathrm{c}^{(d)}[\mathbf{f}, \mathbf{g}] = \mathrm{c}^{(d|d-1)}[\mathrm{c}^{(d-1|d-2)}[\cdots \mathrm{c}^{(2|1)}[\mathrm{c}_{H_1}[\mathbf{f}, \mathbf{g}]] \cdots]. \tag{23}$$

Finally, denote $\mathrm{c}^{(d)}$ as c_∞.

4 Parametrizations and computations

To apply the results of Theorem 3.1, we must specify a covariance function for the integral kernel operators $A_{\mathbf{K}}$. This choice corresponds to selecting a particular neural operator parameterization, following the approach of Kovachki et al. (2023).

In this section, we derive the operator-valued covariance functions for $A_{\boldsymbol{K}}$ under two parametrizations of the integral operator. The first is based on the band-limited Fourier Neural Operator (Section 4.1); the second models the kernel as a non-stationary process, with a prior distribution derived from the classical Matérn family of covariance functions (Section 4.2).

A common assumption for both cases is that the input domain is compact. This ensures that samples of the kernel components $\mathbf{k}_j$ reside in a L^2 space. By further choosing the domain to be the d_x-dimensional flat torus $\mathbb{T}^{d_x} = \mathbb{R}^{d_x}/2\pi\mathbb{Z}^{d_x}$, we are able to exploit Fourier analysis tools. In particular, by assuming that the input functions are band-limited enables tractable computations through the connection of Fourier series with discrete Fourier transforms for evaluations in regular grids.

4.1 Fourier neural operator

Out of the parametrizations proposed by Kovachki et al. (2023), the Fourier neural operator is the most popular due to its computational benefits. By imposing three assumptions into the convolutions kernel – periodicity, shift-invariance, and band-limitedness – we can use the convolution theorem to compute the integrals using sums up to the chosen band-limit of the kernel in the Fourier space.

Concretely, assuming periodicity is equivalent to choosing the domain to be some d_x-dimensional flat torus $\mathcal{X} = \mathbb{T}^{d_x}$, and shift-invariance means kernels satisfy $\mathbf{k}_j(\boldsymbol{w}, \boldsymbol{x}) = \mathbf{k}_j(\boldsymbol{w} - \boldsymbol{x})$, where we abuse notation and represent the kernel as a univariate function of the same name $\mathbf{k}_j : \mathbb{T}^{d_x} \to \mathbb{R}^d$. Under these conditions, any $\mathbf{k}_j$ admits a Fourier series representation:

$$\boldsymbol{k}_j(\boldsymbol{w} - \boldsymbol{x}) = \sum_{\boldsymbol{s} \in \mathbb{Z}^{d_x}} \mathrm{FS}_{\boldsymbol{s}}[\boldsymbol{k}_j] \psi_{-\boldsymbol{s}}(\boldsymbol{w} - \boldsymbol{x}), \tag{24}$$

where $\mathrm{FS}_{\boldsymbol{s}}$ is the $(s_1, \ldots, s_{d_x})$-th coefficient of the Fourier series and $\psi_{\boldsymbol{s}}(\boldsymbol{x}) = \exp[-i \cdot \boldsymbol{s}^\intercal \boldsymbol{x}]$, with $i = \sqrt{-1}$ being the imaginary unit. Moreover, to have a band-limited kernel implies that only finitely many Fourier coefficients are non-zero, i.e. there is some $B_j \in \mathbb{N}$, $1 \le j \le d_x$, such that $\mathrm{FS}_{\boldsymbol{s}}[\mathbf{k}_j] = 0$, if $|s_j| > B_j$, for all $1 \le j \le d_x$.

Under these conditions, despite all input functions $\mathbf{f}$ being represented with a (potentially infinite) Fourier series, by the convolution theorem, the NO layer $H_j[\mathbf{f}]$ is band-limited and its Fourier series coefficients can be computed directly from the product of Fourier coefficients of the kernel function $\mathbf{k}$ and the input function $\mathbf{f}$. Thus, we have that:

$$\mathrm{FS}_{\boldsymbol{s}}[H_j[\mathbf{f}]] = \mathrm{FS}_{\boldsymbol{s}}[\mathbf{k}_j]^\intercal \, \mathrm{FS}_{\boldsymbol{s}}[\mathbf{f}] + \boldsymbol{w}_j^{(1)^\intercal} \, \mathrm{FS}_{\boldsymbol{s}}[\mathbf{f}]. \tag{25}$$

Parameterization of an FNO. Following Section 3.1, when $\mathbf{k}$ is a $\mathbb{R}^d$-valued GP, the kernel integral operator $A_{\mathbf{k}}$ is a function-valued Gaussian process with covariance function of $A_{\mathbf{k}}$ in terms of the covariance function of $\mathbf{k}$, $\mathbf{C}_{\mathbf{k}}$:

$$\mathrm{c}_{A_{\mathbf{k}^\intercal}}[\mathbf{f}_1, \mathbf{f}_2](\boldsymbol{z}_1, \boldsymbol{z}_2) = A_{\mathbf{C}_{\mathbf{k}}}[\mathbf{f}_2^\intercal \mathbf{f}_1](\boldsymbol{z}_1, \boldsymbol{z}_2). \tag{26}$$

The most popular choice proposed by Kovachki et al. (2023) is to directly parametrize the Fourier coefficients of the kernel. Thus, we let these $2B+1$ Fourier coefficients follow i.i.d. centered complex Gaussian distributions with variance $\sigma_{\mathbf{k}}^2$ (Appendix A.1), obtaining the covariance function $\mathbf{C}_{\mathbf{k}}$:

$$\mathbf{k}(\boldsymbol{z} - \boldsymbol{x}) = \sum_{\boldsymbol{s} \in \{-B, \ldots, B\}^{d_x}} \mathrm{FS}_{\boldsymbol{s}}[\mathbf{k}] \cdot \psi_{-\boldsymbol{s}}(\boldsymbol{z} - \boldsymbol{x}) \sim \mathrm{GP}(0, \mathbf{C}_{\mathbf{k}}), \tag{27}$$

$$\mathbf{C}_{\mathbf{k}}((\boldsymbol{z}-\boldsymbol{x}),(\boldsymbol{z}'-\boldsymbol{x}')) = \sigma_{\boldsymbol{k}}^2 \boldsymbol{I}_d \sum_{\boldsymbol{s}\in\{-B,\dots,B\}^{d_x}} \psi_{-\boldsymbol{s}}(\boldsymbol{z}-\boldsymbol{x})\psi_{\boldsymbol{s}}(\boldsymbol{z}'-\boldsymbol{x}'), \tag{28}$$

where B is a hyperparameter of the model controlling the band-limit of the integral kernel.

This allows us to derive a finite-sum representation of the covariance of $A_{\mathbf{k}}$ parameterized by $\sigma_{\mathbf{k}}^2$.

$$\mathrm{c}_{A_{\mathbf{k}^\intercal}}[\mathbf{f}_1,\mathbf{f}_2](\boldsymbol{z},\boldsymbol{z}') = \sigma_{\mathbf{k}}^2(2\pi)^{2d_x} \sum_{\boldsymbol{s}\in\{-B,\dots,B\}^{d_x}} \mathrm{FS}_{-\boldsymbol{s}}[\mathbf{f}_2]^\intercal\,\mathrm{FS}_{\boldsymbol{s}}[\mathbf{f}_1]\psi_{-\boldsymbol{s}}(\boldsymbol{z}-\boldsymbol{z}')\,. \tag{29}$$

4.2 Toroidal Matérn operator

In this section, we propose a model in which the kernel does not admit a shift-invariant decomposition. Another popular decomposition used in the Gaussian process literature is the tensor-product factorization, where the covariance function of a GP factorizes over the input dimension. That is, $\mathrm{f}\colon \mathbb{R}^{d_x} \to \mathbb{R} \sim \mathrm{GP}(0,\mathrm{c})$, where $\mathrm{c}(\boldsymbol{a},\boldsymbol{b}) = \prod_j^{d_x} \mathrm{c}_j(a_j,b_j)$; although the covariance factorizes over the input dimensions, in general, samples from f do not.

Our proposal will make use of the ubiquitous Matérn family of covariance functions, which are characterized by the smoothness parameter ν. Following Borovitskiy et al. (2020), we define the Matérn covariance functions in the d_x-dimensional flat torus $\mathbb{T}^{d_x} = \mathbb{T}\otimes\cdots\otimes\mathbb{T}$ as:

$$\mathrm{c}(\boldsymbol{x},\boldsymbol{x}';\nu,\ell) = (2\pi)^{-d_x} \sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \psi_{\boldsymbol{n}}(\boldsymbol{x})\psi_{-\boldsymbol{n}}(\boldsymbol{x}')\hat{\mathrm{c}}\Big(\sum_{j=1}^{d_x} n_j^2;\nu,\ell\Big). \tag{30}$$

where ℓ is the lengthscale hyper-parameter and the spectral density $\hat{\mathrm{c}}$ is defined as:

$$\hat{\mathrm{c}}(\lambda\,;\nu,\ell) = \begin{cases} \exp\Big[-\frac{\ell^2}{2}\lambda\Big], & \text{if } \nu=\infty, \\ \big(\frac{2\nu}{\ell^2}+\lambda\big)^{-\nu-\frac{d}{2}}, & \text{otherwise.} \end{cases} \tag{31}$$

In general, this kernel is not tensor-product factorized, but for the special case of $\nu=\infty$, the squared exponential covariance function, the factorization holds (Appendix A.2.1). Thus, in general, we enforce the tensor-product factorization:

$$\mathrm{c}(\boldsymbol{x},\boldsymbol{x}';\nu,\boldsymbol{\ell}) = \prod_{j=1}^{d_x} \mathrm{c}\big(x_j,x_j';\nu,\ell_j\big) = (2\pi)^{-d_x} \sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \psi_{\boldsymbol{n}}(\boldsymbol{x})\psi_{-\boldsymbol{n}}(\boldsymbol{x}') \prod_{j=1}^{d_x} \hat{\mathrm{c}}\big(n_j^2;\nu,\ \ell_j\big), \tag{32}$$

where $\boldsymbol{\ell}$ is the automatic relevance determination (ARD) lengthscale hyper-parameter.

Parameterization of a toroidal Matérn operator. So, we consider a convolution kernel $\boldsymbol{k} : \mathcal{X}\times\mathcal{X}\to\mathbb{R}^d$ defined as the product of Matérn covariance functions:

$$\mathbf{C}_{\boldsymbol{k}}(\boldsymbol{z},\boldsymbol{x},\boldsymbol{z}',\boldsymbol{x}') = \mathrm{c}(\boldsymbol{z},\boldsymbol{z}';\nu_z,\ \boldsymbol{\ell}_z)\ \mathrm{c}(\boldsymbol{x},\boldsymbol{x}';\nu_x,\ \boldsymbol{\ell}_x)\ \boldsymbol{I}_d. \tag{33}$$

where $\mathrm{c}(\cdot,\cdot;\nu,\ \boldsymbol{\ell}) : \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is the Matérn covariance functions with smoothness parameter ν and length-scale $\boldsymbol{\ell}$.

Again, following Section 3.1, we express the covariance of the operator as:

$$\mathrm{c}_{A_{\mathbf{k}^\intercal}}[\mathbf{f}_1,\mathbf{f}_2](\boldsymbol{z}_1,\boldsymbol{z}_2) = A_{\mathbf{C}_{\mathbf{k}}}[\mathbf{f}_2^\intercal\mathbf{f}_1](\boldsymbol{z}_1,\boldsymbol{z}_2). \tag{34}$$

Thus, by using the identity Eq. (33), we can derive:

$$\mathrm{c}_{A_{\boldsymbol{k}^\intercal}}[\mathbf{f}_1,\mathbf{f}_2](\boldsymbol{z},\boldsymbol{z}') = \mathrm{c}(\boldsymbol{z},\boldsymbol{z}';\nu_z,\ \boldsymbol{\ell}_z)\ (2\pi)^{d_x} \sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \mathrm{FS}_{\boldsymbol{n}}[\mathbf{f}_1]^\intercal\,\mathrm{FS}_{-\boldsymbol{n}}[\mathbf{f}_2] \prod_{j=1}^{d_x} \hat{\mathrm{c}}\big(n_j^2;\nu_x,\boldsymbol{\ell}_{x,j}\big). \tag{35}$$

5 Experimental validation

In this section, we empirically show i) the agreement between finite width neural operators with increasing width and their corresponding infinite-width neural operator, and ii) evaluate our model against FNO in a regression task.

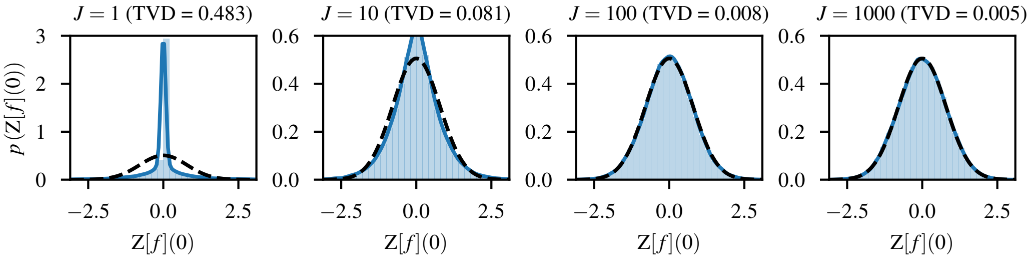


Figure 1: A density estimation of the ● empirical distribution of the output of increasing channel dimension compared to the ◌ infinite width distribution. On top of each plot we show the total variation distance of the empirical distribution against the infinite width distribution.

Section 5.1 explores the distribution of untrained randomly initialized Fourier neural operators of varying width and the distribution of the infinite-width FNO (∞-FNO). As expected from the theoretical results, these distributions should eventually match as the hidden dimension increases.

Section 5.2 considers two tasks: a synthetic regression example, where the task is to predict the output of a non-linear operator, and the task of predicting the final evolution of Burgers' equation given the initial state. This situation is not covered in our theory, since it only applies to the distribution of the neural operators at initialization, but our experiments show the behavior of the posteriors of infinite-width neural operators against Adam trained finite-width neural operators of increasing width.

Throughout this section, our stating point is a single hidden-layer neural operator $Z[\mathrm{f}]\colon \mathbb{T} \to \mathbb{R} := (\boldsymbol{w}_2^\intercal \circ \mathrm{ReLU} \circ (A_{\mathbf{K}} + \boldsymbol{W}_1))[\mathrm{f}]$. More details for each experiment can be found in Appendix C. All experiments were implemented in Python, mainly based on the GPyTorch (Gardner et al., 2018) library, and run in a desktop computer using a Titan RTX. Code is avaliable at `https://github.com/spectraldani/infinite-neural-operator`.

5.1 Empirical demonstration of results

In this experiment, we demonstrate that our analytical computation of the variance for a neural operator layer H agrees with empirical estimates, and we validate Theorem 3.1 by showing that the output of a neural operator Z converges to a Gaussian distribution as the number of hidden channels J increases.

Throughout all experiments, the input function $\mathrm{f}\colon \mathbb{T} \to \mathbb{R}$ has band-limit $B = 3$, with its output values $\mathrm{f}(x)$ sampled from a uniform distribution $\mathcal{U}(-1, 1)$. Both operators are evaluated at $x = 0$, so we analyze the empirical distributions of $H[\mathrm{f}](0)$ and $Z[\mathrm{f}](0)$, respectively.

Following Section 4.1, we parametrize the integral kernel operators using band-limited functions. The band-limit of the kernel is set equal to that of the input f, and the kernel coefficients are drawn from a Gaussian distribution with unit variance scaled by the inverse of the number of hidden channels.

As shown in Fig. 2, the empirical estimate of the variance converges to the theoretical value as the number of Monte Carlo samples increases, supporting the correctness of our variance computation. Furthermore, Fig. 1 shows that, as the number of hidden channels grows, the total variation distance (TVD) between the empirical distribution and a Gaussian distribution approaches zero, thereby further verifying the validity of Theorem 3.1.

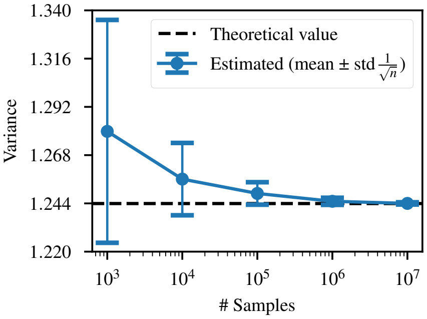
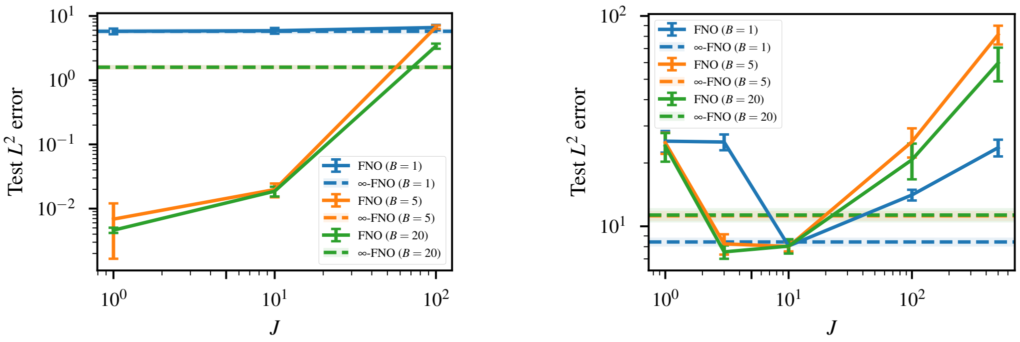


Figure 2: Plot of the MC estimate for the variance of $H[\mathrm{f}](0)$ against our analytical computation (Sec. 3.1).

5.2 Regression tasks

In this task, we're given n pairs of 1D functions $\{\mathrm{f}_i, \mathrm{g}_i\}_{i=1}^n$ evaluated in a grid with $m = 2B_m + 1$ points. We consider FNOs of increasing width $J \in \mathbb{N}^+$, as well as ∞-FNOs, both with increasing kernel band-limits $B \in \{1, 5, 20\}$. These models will be trained on two datasets: (a) A operator



(a) Synthetic data. ∞-FNO with $B = 5$ (▪▪) and $B = 20$ (▪▪) overlap.

(b) 1D Burgers' equation ($\nu = 0.002$). ∞-FNO with $B = 5$ (▪▪) and $B = 20$ (▪▪) overlap.

Figure 3: Results for the regression experiments. Mean and std. of test L^2 loss as a function of width J for different band-limits B.

generated by a randomly-initialized ground truth FNO Z_{true} with band-limit $B = 5$ and width $J = 1$. We sample $n = 100$ input functions $\mathrm{f}_i\colon \mathbb{T} \to \mathbb{R}$ with uniformily-distributed outputs $\mathcal{U}(-1, 1)$ and band-limit $B_m = 5$. (b) 1D Burgers' equation dataset from Takamoto et al. (2022) with $\nu = 0.002$. The task is to predict the end state ($t = 2$) given the initial condition ($t = 0$). Due to memory constraints, we subsample the total dataset data to $n = 100$ functions and a grid size of $m = 103$.

The hyperparameters of the ∞-FNO are estimated using L-BFGS, while the parameters of the FNOs are optimized with Adam using a step size of 0.001. We evaluate all models using 5-fold cross-validation and report the average and standard deviation of the empirical L^2 norm of the prediction error. For ∞-FNOs, we use the posterior mean as the prediction.

In general, we do not expect close agreement between the predictive performance of ∞-FNOs and finite-width FNOs, as the former corresponds to a Bayesian estimate while the latter are trained by minimizing an empirical risk, nonetheless, as observed in Figs. 3a and 3b, there is consistency between the gap of hyperparameters in the same model class.

In the synthetic case, as we know the band-limits of the ground truth operator, Fig. 3a shows that the models are only able to accurately predict the output when their band-limits exceed that of the ground truth.

6 Related works

Infinite limits of stochastic NNs. The study of infinite-width Bayesian neural networks began with the seminal work of Neal (1995) and was later extended to deep architectures (Lee et al., 2018; Yang, 2019; Matthews et al., 2018). Our analysis builds on the ideas developed by Matthews et al. (2018). From the outset, these infinite-width models were considered ”disappointing” (Neal, 1995), a view reinforced by findings that neither the Bayesian limit nor the neural tangent kernel limit learns features from data (Aitchison, 2020). However, recent work shows these models still reflect the different inductive biases of their finite-width counterparts (Novak et al., 2019), and that alternative initialization distributions can enable feature learning in the infinite-width setting (Yang and Hu, 2021).

Bayesian neural operators. Several works have investigated approximate Bayesian uncertainty quantification in finite-width neural operators using function-valued Gaussian processes. Magnani et al. (2022, 2024) both employ last-layer Laplace approximations to construct GP approximations of the posterior distribution. In addition, Magnani et al. (2022) considers the case where the kernel $\mathbf{K}$ of the integral operator $A_{\mathbf{K}}$ follows a Matérn GP prior. However, their analysis is restricted to the finite-width regime on compact subsets of Euclidean space, whereas our work focuses on the flat torus.

Kernel methods for operator learning. Batlle et al. (2024) propose the use of kernel methods for operator learning, leveraging operator-valued kernels and the representer theorem in their corresponding RKHS. Their results are promising and highlight the potential of kernel-based approaches in this domain. Our contribution introduces an additional way to construct operator-valued kernels based on neural operators, enabling new kernel-based models for operator learning.

7 Discussion

In this work, we formalized the concept of infinite-width Bayesian neural operators, established their existence (Theorem 3.1), and described how to compute their associated covariance functions (Section 4). We validated these results empirically (Section 5.1) and further assessed the performance of these models in a regression setting (Section 5.2).

Our contributions lay a foundation for future investigations, particularly in bridging the gap between SGD-trained finite-width neural operators and their infinite-width counterparts. Addressing this challenge will require extending the neural tangent kernel framework (Jacot et al., 2018; Lee et al., 2019) to settings involving Hilbert space-valued functions. Moreover, while we focused on the ubiquitous FNO architecture, deriving covariance functions for other architectures, such as the graph neural operator (Kovachki et al., 2023), remains an open direction.

Limitations. Our current implementation for computing the required kernel quantities scales with cubically in both the evaluation grid size and the number of training functions. We anticipate that future work can improve computational efficiency by leveraging advances from the Gaussian process literature to improve scalability and efficiency (Borovitskiy et al., 2020; Gilboa et al., 2015).

Acknowledgments and Disclosure of Funding

YZ acknowledges support by the Engineering and Physical Sciences Research Council with grant number EP/S021566/1. YS was supported by Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) through the Jovem Cientista do Nosso Estado Program (E-26/201.375/2022 (272760)) and by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) through the Productivity in Research Scholarship (306695/2021-9, 305159/2025-9). DM was supported by the Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) (SEI-260003/000709/2023) and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (404336/2023-0, 305692/2025-9).

References

Laurence Aitchison. Why bigger is not always better: on finite and infinite neural networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020. URL https://proceedings.mlr.press/v119/aitchison20a.html.

Pau Batlle, Matthieu Darcy, Bamdad Hosseini, and Houman Owhadi. Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496, 2024. URL https://doi.org/10.1016/j.jcp.2023.112549.

Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Matérn Gaussian processes on Riemannian manifolds. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://papers.nips.cc/paper/2020/hash/92bf5e6240737e0326ea59846a83e076-Abstract.html.

Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2009. URL https://proceedings.neurips.cc/paper/2009/hash/5751ec3e9a4feab575962e78e006250d-Abstract.html.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/27e8e17134dd7083b050476733207ea1-Abstract.html.

Elad Gilboa, Yunus Saatçi, and John P. Cunningham. Scaling multidimensional inference for structured Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 2015. URL http://dx.doi.org/10.1109/TPAMI.2013.192.

Insu Han, Amir Zandieh, Jaehoon Lee, Roman Novak, Lechao Xiao, and Amin Karbasi. Fast neural kernel embeddings for general activations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/e7be1f4c6212c24919cd743512477c13-Abstract-Conference.html.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research (JMLR)*, 17, 2016. URL http://jmlr.org/papers/v17/11-315.html.

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. *Journal of Machine Learning Research (JMLR)*, 24, 2023. URL http://jmlr.org/papers/v24/21-1524.html.

Jaehoon Lee, Jascha Sohl-Dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=B1EA-M-0Z.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0d1a9651497a38d8b1c3871c84528bd4-Abstract.html.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS Journal of Data Science*, 1(3), 2024. URL https://doi.org/10.1145/3648506.

Emilia Magnani, Nicholas Krämer, Runa Eschenhagen, Lorenzo Rosasco, and Philipp Hennig. Approximate Bayesian neural operators: Uncertainty quantification for parametric PDEs. 2022. URL https://arxiv.org/abs/2208.01565.

Emilia Magnani, Marvin Pförtner, Tobias Weber, and Philipp Hennig. Linearization turns neural operators into function-valued Gaussian processes. 2024. URL https://arxiv.org/abs/2406.05072.

Alexander G. de G. Matthews, Mark Rowland, Jiri Hron, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. 2018. URL https://arxiv.org/abs/1804.11271.

Edith Mourier. L-random elements and l'-random elements in Banach spaces. In *Contributions to Probability Theory*, pages 231–242. University of California Press, December 1956. URL https://doi.org/10.1525/9780520350670-017.

Radford M. Neal. *Bayesian Learning for Neural Networks.* PhD thesis, University of Toronto, 1995.

Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Jiri Hron, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=B1g30j0qF7.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. 2022. URL https://arxiv.org/abs/2202.11214.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://papers.neurips.cc/paper_files/paper/2022/hash/0a9747136d411fb83f0cf81820d44afb-Abstract-Datasets_and_Benchmarks.html.

Christopher Williams. Computing with infinite networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1996. URL https://proceedings.neurips.cc/paper/1996/hash/ae5e3ce40e0404a45ecacaaf05e5f735-Abstract.html.

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://papers.neurips.cc/paper_files/paper/2019/hash/5e69fda38cda2060819766569fd93aa5-Abstract.html.

Greg Yang and Edward J. Hu. Tensor programs IV: Feature learning in infinite-width neural networks. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. URL https://proceedings.mlr.press/v139/yang21c.html.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contributions consist of i) a major theorem (Theorem 3.1) characterizing the infinite-width limit of Neural Operators (NOs), and ii) closed-form equations (Section 4) for computing the kernel induced by NO architectures, which allow for practical implementations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have included the limitations of our work in the Discussion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate ”Limitations” section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main assumptions are placed inside the theorems, the main text includes a proof sketch of our main result, and the full proof for it and related lemmas is provided in the supplemental material.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The main text include a summary of how the dataset was generated and how to compute the necessary quantities for implementation. The supplementary material include further details on hyper-parameters and initializations alongside the full source code for the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will provide code as part of the supplemental material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe this in the experimental section and include more detail in the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We include mean and standard error for experiment 5.1, representing the different sample used for estimation and experiment 5.2 contain mean and standard deviation of representing each of the 5-fold splits.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer ”Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have included the specs of the machine used for experiments in the main text and in the supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn’t make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work deals with theoretical results and synthetic data, thus, we believe that it conforms to the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

Justification: This work deals with theoretical results and synthetic data, we believe there is no direct path for direct negative application of this work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The models used in this work do not pose such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The authors of frameworks used for implementation are credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

Infinite Neural Operators: Gaussian processes on functions (Supplemental Materials)

A Covariance Function Computation

In this section, we will work out in detail the computations of Section 4, first for the Fourier neural operator (FNO) case and later for the toroidal Matérn operator.

A.1 Fourier neural operator

Under the direct parametrization of the integral kernel operator, the coefficients of the kernel's Fourier series (FS) are parametrized and randomly sampled at initialization. Therefore, our first step is to derive what the Gaussian process distribution of a band-limited function with i.i.d. Gaussian FS coefficients is.

Fourier series. Given a function on the d_x-dimensional torus $\mathbf{f}(\cdot)\colon \mathbb{T}^{d_x} \to \mathbb{R}^d$, $\mathbf{f} = (\mathrm{f}_1, \ldots, \mathrm{f}_d)$, it can be represented in terms of a Fourier series:

$$\mathrm{f}_p(\boldsymbol{x}) = \sum_{\boldsymbol{s}\in\mathbb{Z}^{d_x}} \mathrm{FS}_{\boldsymbol{s}}[\mathrm{f}_p]\ \psi_{-\boldsymbol{s}}(\boldsymbol{x}), \tag{36}$$

for $p \in \{1, \ldots, d\}$, where,

$$\mathrm{FS}_{\boldsymbol{s}}[\mathrm{f}_p] = (2\pi)^{-d_x} \int_{[-\pi,\pi]^{d_x}} \mathrm{f}_p(\boldsymbol{t})\ \psi_{\boldsymbol{s}}(\boldsymbol{t})\,\mathrm{d}\boldsymbol{t}, \tag{37}$$

$$\psi_{\boldsymbol{s}}(\boldsymbol{x}) = \exp[-i \cdot \boldsymbol{s}^\intercal \boldsymbol{x}], \tag{38}$$

and $i = \sqrt{-1}$ is the imaginary unit.

Note that, as f_p is a real-valued function, we also have that $\mathrm{FS}_{\boldsymbol{s}}[\mathrm{f}_p] = \overline{\mathrm{FS}_{-\boldsymbol{s}}}[\mathrm{f}_p]$, where $\bar{z}$ is the complex conjugate.

Gaussian distributed band-limited functions. Consider the sequence $\hat{\boldsymbol{f}}\colon [-B, \ldots, B]^{d_x} \to \mathbb{C}$ defined as:

$$\Re\hat{f}_{\mathbf{0}} \sim \mathcal{N}\big(0, \sigma^2\big), \qquad \Im\hat{f}_{\mathbf{0}} = 0, \tag{39}$$

$$\Re\hat{f}_{\boldsymbol{s}} \sim \mathcal{N}\big(0, \sigma^2/2\big), \qquad \Im\hat{f}_{\boldsymbol{s}} \sim \mathcal{N}\big(0, \sigma^2/2\big), \tag{40}$$

$$\Re\hat{f}_{-\boldsymbol{s}} = \Re\hat{f}_{\boldsymbol{s}}, \qquad \Im\hat{f}_{-\boldsymbol{s}} = -\Im\hat{f}_{\boldsymbol{s}}, \tag{41}$$

where $\Re z$ and $\Im z$ are the real and imaginary parts of the complex number z, respectively, and all random variables are independent of each other, except the conjugate duals $\hat{f}_{\boldsymbol{s}}$ and $\hat{f}_{-\boldsymbol{s}}$. For $\boldsymbol{s} \neq \mathbf{0}$, the equations above can also be expressed as:

$$\begin{bmatrix} \Re\hat{f}_{\boldsymbol{s}} \\ \Im\hat{f}_{\boldsymbol{s}} \\ \Re\hat{f}_{-\boldsymbol{s}} \\ \Im\hat{f}_{-\boldsymbol{s}} \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \frac{1}{2} \begin{bmatrix} \sigma^2 & 0 & \sigma^2 & 0 \\ 0 & \sigma^2 & 0 & -\sigma^2 \\ \sigma^2 & 0 & \sigma^2 & 0 \\ 0 & -\sigma^2 & 0 & \sigma^2 \end{bmatrix} \right). \tag{42}$$

With this in mind, the expectation of the product of two elements is:

$$\mathbb{E}[\hat{f}_{\boldsymbol{s}} \cdot \hat{f}_{\boldsymbol{s}'}] = \mathbb{E}[\Re\hat{f}_{\boldsymbol{s}}\Re\hat{f}_{\boldsymbol{s}'}] - \mathbb{E}[\Im\hat{f}_{\boldsymbol{s}}\Im\hat{f}_{\boldsymbol{s}'}] + i\mathbb{E}[\Re\hat{f}_{\boldsymbol{s}'}\Im\hat{f}_{\boldsymbol{s}}] + i\mathbb{E}[\Re\hat{f}_{\boldsymbol{s}}\Im\hat{f}_{\boldsymbol{s}'}] \tag{43}$$

$$= \mathbb{E}[\Re\hat{f}_{\boldsymbol{s}}\Re\hat{f}_{\boldsymbol{s}'}] - \mathbb{E}[\Im\hat{f}_{\boldsymbol{s}}\Im\hat{f}_{\boldsymbol{s}'}] \tag{44}$$

$$= \begin{cases} \mathbb{E}[\Re\hat{f}_{\boldsymbol{s}}\Re\hat{f}_{\boldsymbol{s}}] - \mathbb{E}[\Im\hat{f}_{\boldsymbol{s}}\Im\hat{f}_{\boldsymbol{s}}] & \text{if } \boldsymbol{s}' = \boldsymbol{s}, \\ \mathbb{E}[\Re\hat{f}_{\boldsymbol{s}}\Re\hat{f}_{\boldsymbol{s}}] + \mathbb{E}[\Im\hat{f}_{\boldsymbol{s}}\Im\hat{f}_{\boldsymbol{s}}] & \text{if } \boldsymbol{s}' = -\boldsymbol{s}, \\ 0 & \text{otherwise;} \end{cases} \tag{45}$$

$$= \begin{cases} \sigma^2 & \text{if } \boldsymbol{s}' = \boldsymbol{0} \text{ and } \boldsymbol{s} = \boldsymbol{0}, \\ \sigma^2/2 - \sigma^2/2 & \text{if } \boldsymbol{s}' = \boldsymbol{s}, \\ \sigma^2/2 + \sigma^2/2 & \text{if } \boldsymbol{s}' = -\boldsymbol{s}, \\ 0 & \text{otherwise;} \end{cases} \tag{46}$$

$$= \begin{cases} \sigma^2 & \text{if } \boldsymbol{s}' = -\boldsymbol{s}, \\ 0 & \text{otherwise.} \end{cases} \tag{47}$$

Thus, we can define the Gaussian process $\mathrm{f}\colon \mathbb{T}^{d_x} \to \mathbb{R}$ through a Fourier series representation:

$$\mathrm{f}(\boldsymbol{x}) = \sum_{\boldsymbol{s} \in \{-B,\ldots,B\}^{d_x}} \hat{f}_{\boldsymbol{s}}\ \psi_{-\boldsymbol{s}}(\boldsymbol{x}). \tag{48}$$

We compute the covariance function of f as:

$$\mathrm{c}_{\mathrm{f}}(\boldsymbol{x}) = \mathbb{E}[\mathrm{f}(\boldsymbol{x}) \cdot \mathrm{f}(\boldsymbol{x}')] \tag{49}$$

$$= \mathbb{E}\left[\sum_{\boldsymbol{s} \in \{-B,\ldots,B\}^{d_x}} \hat{f}_{\boldsymbol{s}}\ \psi_{-\boldsymbol{s}}(\boldsymbol{x}) \sum_{\boldsymbol{s}' \in \{-B,\ldots,B\}^{d_x}} \hat{f}_{\boldsymbol{s}'}\ \psi_{-\boldsymbol{s}'}(\boldsymbol{x}')\right] \tag{50}$$

$$= \sum_{\boldsymbol{s},\boldsymbol{s}' \in \{-B,\ldots,B\}^{d_x}} \mathbb{E}[\hat{f}_{\boldsymbol{s}}\ \psi_{-\boldsymbol{s}}(\boldsymbol{x})\ \hat{f}_{\boldsymbol{s}'}\ \psi_{-\boldsymbol{s}'}(\boldsymbol{x}')] \tag{51}$$

$$= \sum_{\boldsymbol{s},\boldsymbol{s}' \in \{-B,\ldots,B\}^{d_x}} \mathbb{E}[\hat{f}_{\boldsymbol{s}} \cdot \hat{f}_{\boldsymbol{s}'}]\psi_{-\boldsymbol{s}}(\boldsymbol{x})\ \psi_{-\boldsymbol{s}'}(\boldsymbol{x}') \tag{52}$$

$$= \sigma^2 \sum_{\boldsymbol{s} \in \{-B,\ldots,B\}^{d_x}} \psi_{-\boldsymbol{s}}(\boldsymbol{x})\ \psi_{\boldsymbol{s}}(\boldsymbol{x}') \tag{53}$$

$$= \sigma^2 \sum_{\boldsymbol{s} \in \{-B,\ldots,B\}^{d_x}} \psi_{-\boldsymbol{s}}(\boldsymbol{x} - \boldsymbol{x}'). \tag{54}$$

A.1.1 Covariance after convolution $\mathrm{c}_{A_{\mathbf{k}^\intercal}}$

Let us place a centered Gaussian distribution on the Fourier series of the band-limited convolution kernel $\mathbf{k}\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}^d$, so that:

$$\mathbf{C}_{\mathbf{k}}((\boldsymbol{z} - \boldsymbol{x}), (\boldsymbol{z}' - \boldsymbol{x}')) = \sigma^2 \boldsymbol{I}_d \sum_{\boldsymbol{s} \in \{-B,\ldots,B\}^{d_x}} \psi_{-\boldsymbol{s}}((\boldsymbol{z} - \boldsymbol{x}) - (\boldsymbol{z}' - \boldsymbol{x}')) \tag{55}$$

$$= \sigma^2 \boldsymbol{I}_d \sum_{\boldsymbol{s} \in \{-B,\ldots,B\}^{d_x}} \psi_{-\boldsymbol{s}}(\boldsymbol{z} - \boldsymbol{x})\psi_{\boldsymbol{s}}(\boldsymbol{z}' - \boldsymbol{x}') \tag{56}$$

$$= \sigma^2 \boldsymbol{I}_d \sum_{\boldsymbol{s} \in \{-B,\ldots,B\}^{d_x}} \psi_{-\boldsymbol{s}}(\boldsymbol{z} - \boldsymbol{z}')\psi_{\boldsymbol{s}}(\boldsymbol{x} - \boldsymbol{x}') \tag{57}$$

So, let us consider the quantity $\mathrm{c}_{A_{\mathbf{k}}}[\mathbf{f}_1, \mathbf{f}_2]$ for arbitrary functions $\mathbf{f}_1$ and $\mathbf{f}_2$:

$$A_{\mathrm{c}_{\mathbf{k}}}[\mathbf{f}_2^\intercal \mathbf{f}_1](\boldsymbol{z}, \boldsymbol{z}') = \iint_{\mathcal{X}} \mathbf{f}_2^\intercal(\boldsymbol{x}')\mathbf{C}_{\mathbf{k}}((\boldsymbol{z} - \boldsymbol{x}), (\boldsymbol{z}' - \boldsymbol{x}'))\mathbf{f}_1(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \tag{58}$$

$$= \iint_{\mathcal{X}} \mathbf{f}_2^\intercal(\boldsymbol{x}')\Big(\sigma^2 \boldsymbol{I}_d \sum_{\boldsymbol{s} \in \{-B,\ldots,B\}^{d_x}} \psi_{-\boldsymbol{s}}(\boldsymbol{z} - \boldsymbol{z}')\psi_{\boldsymbol{s}}(\boldsymbol{x} - \boldsymbol{x}')\Big)\mathbf{f}_1(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \tag{59}$$

$$= \sigma^2 \sum_{\boldsymbol{s} \in \{-B,\dots,B\}^{d_x}} \psi_{-\boldsymbol{s}}(\boldsymbol{z}-\boldsymbol{z}') \iint_{\mathcal{X}} \psi_{\boldsymbol{s}}(\boldsymbol{x}-\boldsymbol{x}')\mathbf{f}_2^{\intercal}(\boldsymbol{x}')\mathbf{f}_1(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \tag{60}$$

$$= \sigma^2 \sum_{\boldsymbol{s} \in \{-B,\dots,B\}^{d_x}} \psi_{-\boldsymbol{s}}(\boldsymbol{z}-\boldsymbol{z}')(2\pi)^{2d_x}\,\mathrm{FS}_{[\boldsymbol{s},-\boldsymbol{s}]}[\mathbf{f}_2^{\intercal}\mathbf{f}_1] \tag{61}$$

$$= \sigma^2(2\pi)^{2d_x} \sum_{\boldsymbol{s} \in \{-B,\dots,B\}^{d_x}} \psi_{-\boldsymbol{s}}(\boldsymbol{z}-\boldsymbol{z}')\,\mathrm{FS}_{-\boldsymbol{s}}[\mathbf{f}_2]^{\intercal}\,\mathrm{FS}_{\boldsymbol{s}}[\mathbf{f}_1]. \tag{62}$$

A.2 Toroidal Matérn operator

Definition A.1 (Matérn family of kernels on a closed manifold)**.** The Matérn family of kernels c with lengthscale ℓ in a d-dimensional closed manifold $\mathcal{M}$ are described as:

$$\mathrm{c}(a,b;\nu,\ell) = \sum_{k=1}^{\infty} \hat{\mathrm{c}}(\lambda_k;\nu,\ell)\cdot\phi_k(a)\cdot\phi_k(b), \tag{63}$$

$$\hat{\mathrm{c}}(\lambda_k;\nu,\ell) = \begin{cases} \exp\left[-\frac{\ell^2}{2}\lambda_k\right] & \text{if } \nu=\infty, \\ \left(\frac{2\nu}{\ell^2}+\lambda_k\right)^{-\nu-\frac{d}{2}} & \text{otherwise.} \end{cases}, \tag{64}$$

where, λ_k and ϕ_k are the k-th eigenvalues and eigenfunctions, respectively, of the Laplace-Beltrami operator of the manifold $\mathcal{M}$.

For a 1-dimensional flat torus, an orthonormal eigensystem for the Laplace-Beltrami operator is:

$$\lambda_k = \lfloor k/2 \rfloor^2; \qquad \phi_k(x) = \begin{cases} 1/\sqrt{2}\pi & \text{if } k=1, \\ \cos\left(\sqrt{\lambda_k}x\right)/\sqrt{\pi} & \text{if } k=2n, \\ \sin\left(\sqrt{\lambda_k}x\right)/\sqrt{\pi} & \text{if } k=2n+1. \end{cases} \tag{65}$$

Additionally, for a d_x-dimensional flat torus, an orthonormal eigensystem for the Laplace-Beltrami operator is given by the sum and product of the 1-dimensional eigensystem such that, given an index $\boldsymbol{k}=[k_1,\dots,k_{d_x}]$, we have that $\lambda_{\boldsymbol{k}} = \sum_{j=1}^{d_x}\lambda_{k_j}$ and $\phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^{d_x}\phi_{k_j}(x_j)$.

Expression of 1-dimensional toroidal kernel using complex exponentials. For convenience, we will rewrite the series expansion of this kernel to use exponentials of complex numbers.

Start by noting that:

$$\phi_{2n}(a)\phi_{2n}(b) = \frac{1}{\pi}\cos(na)\cos(nb), \qquad \phi_{2n+1}(a)\phi_{2n+1}(b) = \frac{1}{\pi}\sin(na)\sin(nb), \tag{66}$$

Therefore,

$$\phi_{2n}(a)\phi_{2n}(b) + \phi_{2n+1}(a)\phi_{2n+1}(b) = \frac{1}{\pi}\cos(na)\cos(nb) + \frac{1}{\pi}\sin(na)\sin(nb) \tag{67}$$

$$= \frac{1}{\pi}\cos(n(a-b))) \tag{68}$$

$$= \frac{1}{2\pi}(\exp[in(a-b)] + \exp[-in(a-b)]) \tag{69}$$

$$= \frac{1}{2\pi}(\exp[ina]\exp[-inb] + \exp[-ina]\exp[inb]) \tag{70}$$

$$= \frac{1}{2\pi}(\psi_{-n}(a)\psi_n(b) + \psi_n(a)\psi_{-n}(b)) \tag{71}$$

Now, we can rewrite the Matérn kernel expression as:

$$\mathrm{c}(a,b;\nu,\ell) = \sum_{k=1}^{\infty} \hat{\mathrm{c}}(\lambda_k;\nu,\ell)\cdot\phi_k(a)\cdot\phi_k(b) \tag{72}$$

$$= \frac{1}{2\pi}\hat{\mathrm{c}}(\lambda_1;\nu,\ell) + \sum_{n=1}^{\infty}\hat{\mathrm{c}}(\lambda_{2n};\nu,\ell)(\phi_{2n}(a)\phi_{2n}(b) + \phi_{2n+1}(a)\phi_{2n+1}(b)) \tag{73}$$

$$= \frac{1}{2\pi}\hat{\mathrm{c}}(0;\nu,\ell) + \sum_{n=1}^{\infty} \frac{1}{2\pi}\hat{\mathrm{c}}\big(n^2;\nu,\ell\big)(\psi_{-n}(a)\psi_n(b) + \psi_n(a)\psi_{-n}(b)) \tag{74}$$

$$= \frac{1}{2\pi}\hat{\mathrm{c}}(0;\nu,\ell) + \sum_{n=1}^{\infty} \frac{1}{2\pi}\hat{\mathrm{c}}\big(n^2;\nu,\ell\big)\psi_{-n}(a)\psi_n(b) + \sum_{n=1}^{\infty} \frac{1}{2\pi}\hat{\mathrm{c}}\big(n^2;\nu,\ell\big)\psi_n(a)\psi_{-n}(b) \tag{75}$$

$$= \frac{1}{2\pi}\hat{\mathrm{c}}(0;\nu,\ell) + \sum_{n=-1}^{-\infty} \frac{1}{2\pi}\hat{\mathrm{c}}\big(n^2;\nu,\ell\big)\psi_n(a)\psi_{-n}(b) + \sum_{n=1}^{\infty} \frac{1}{2\pi}\hat{\mathrm{c}}\big(n^2;\nu,\ell\big)\psi_n(a)\psi_{-n}(b) \tag{76}$$

$$= \frac{1}{2\pi}\hat{\mathrm{c}}(0)\psi_0(a)\psi_{-0}(b;\nu,\ell) + \sum_{n\in\mathbb{Z}\setminus\{0\}} \frac{1}{2\pi}\hat{\mathrm{c}}\big(n^2;\nu,\ell\big)\psi_n(a)\psi_{-n}(b) \tag{77}$$

$$= \frac{1}{2\pi}\sum_{n\in\mathbb{Z}} \hat{\mathrm{c}}\big(n^2;\nu,\ell\big)\psi_n(a)\psi_{-n}(b). \tag{78}$$

Product kernel for $\mathbb{T}^{d_x}$ In order to have one lengthscale per dimension, we will make a tensor product kernel where the kernel of a d_x-dimensional torus $\mathbb{T}^{d_x}$ is the product of the 1-d toroidal kernel for each dimension:

$$\mathrm{c}(\boldsymbol{a},\boldsymbol{b};\nu,\boldsymbol{\ell}_j) = \prod_{j=1}^{d_x} \mathrm{c}(a_j,b_j;\nu,\ell_j) \tag{79}$$

$$= \prod_{j=1}^{d_x} \frac{1}{2\pi}\sum_{n\in\mathbb{Z}} \hat{\mathrm{c}}\big(n^2;\nu,\ell_j\big)\psi_n(a_j)\psi_{-n}(b_j) \tag{80}$$

$$= (2\pi)^{-d_x}\prod_{j=1}^{d_x}\sum_{n\in\mathbb{Z}} \hat{\mathrm{c}}\big(n^2;\nu,\ell_j\big)\psi_n(a_j)\psi_{-n}(b_j) \tag{81}$$

$$= (2\pi)^{-d_x}\sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}}\prod_{j=1}^{d_x} \hat{\mathrm{c}}\big(n_j^2;\nu,\ell_j\big)\psi_{n_j}(a_j)\psi_{-n_j}(b_j) \tag{82}$$

$$= (2\pi)^{-d_x}\sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \psi_{\boldsymbol{n}}(\boldsymbol{a})\psi_{-\boldsymbol{n}}(\boldsymbol{b})\prod_{j=1}^{d_x} \hat{\mathrm{c}}\big(n_j^2;\nu,\ell_j\big). \tag{83}$$

A.2.1 $\nu = \infty$ lets Matérn kernel be a product kernel

Notice that when $\nu = \infty$ and $\ell_j = \ell$, we have

$$\mathrm{c}(\boldsymbol{a},\boldsymbol{b};\nu,\boldsymbol{\ell}) = (2\pi)^{-d_x}\sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \psi_{\boldsymbol{n}}(\boldsymbol{a})\psi_{-\boldsymbol{n}}(\boldsymbol{b})\prod_{j=1}^{d_x} \exp\big[-\ell^2 n_j^2/2\big]. \tag{84}$$

$$= (2\pi)^{-d_x}\sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \psi_{\boldsymbol{n}}(\boldsymbol{a})\psi_{-\boldsymbol{n}}(\boldsymbol{b})\exp\left[-\sum_{j=1}^{d_x}\ell^2 n_j^2/2\right]. \tag{85}$$

$$= (2\pi)^{-d_x}\sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \psi_{\boldsymbol{n}}(\boldsymbol{a})\psi_{-\boldsymbol{n}}(\boldsymbol{b})\exp\left[-\frac{\ell^2}{2}\sum_{j=1}^{d_x} n_j^2\right]. \tag{86}$$

$$= (2\pi)^{-d_x}\sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \psi_{\boldsymbol{n}}(\boldsymbol{a})\psi_{-\boldsymbol{n}}(\boldsymbol{b})\hat{\mathrm{c}}\Big(\sum_{j=1}^{d_x} n_j^2;\nu,\ell\Big). \tag{87}$$

With the proper rearrangement, we can see that this fits the definition of a Matérn kernel in the $\mathbb{T}^{d_x}$, as the eigenvalues of its Beltrami-Laplace operator can be expressed as the sum of the eigenvalues for the 1-dimensional flat torus $\mathbb{T}$.

A.2.2 Covariance after convolution $\mathrm{c}_{A_{\mathbf{k}}}$

Let us place a centered factored Matérn prior in the convolution kernel $\mathbf{k}\colon \mathcal{Z} \times \mathcal{X} \to \mathbb{R}^d$, so that:

$$\mathbf{C}_{\mathbf{k}}(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{z}', \boldsymbol{x}') = \mathrm{c}(\boldsymbol{z}, \boldsymbol{z}'; \nu_z, \boldsymbol{\ell}_z)\ \mathrm{c}(\boldsymbol{x}, \boldsymbol{x}'; \nu_x, \ell_x)\boldsymbol{I}_d. \tag{88}$$

When clear from context, we will suppress the dependency on the hyper-parameters of the Matérn kernel.

So, let us consider the quantity $\mathrm{c}_{A_{\mathbf{k}}}[\mathbf{f}_1, \mathbf{f}_2]$ for arbitrary functions $\mathbf{f}_1$ and $\mathbf{f}_2$:

$$A_{\mathrm{c}_{\mathbf{k}^{\mathsf{T}}}}[\mathbf{f}_2^{\mathsf{T}}\mathbf{f}_1](\boldsymbol{z}, \boldsymbol{z}')$$

$$= \iint_{\mathcal{X}} \mathbf{f}_2^{\mathsf{T}}(\boldsymbol{x}')\mathbf{C}_{\mathbf{k}}(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{z}', \boldsymbol{x}')\mathbf{f}_1(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \tag{89}$$

$$= \iint_{\mathcal{X}} \mathbf{f}_2^{\mathsf{T}}(\boldsymbol{x}')\Big(\mathrm{c}(\boldsymbol{z}, \boldsymbol{z}'; \nu_z, \boldsymbol{\ell}_z)\mathrm{c}(\boldsymbol{x}, \boldsymbol{x}'; \nu_x, \ell_x)\boldsymbol{I}_d\Big)\mathbf{f}_1(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \tag{90}$$

$$= \mathrm{c}(\boldsymbol{z}, \boldsymbol{z}') \iint_{\mathcal{X}} \mathbf{f}_2^{\mathsf{T}}(\boldsymbol{x}')\mathbf{f}_1(\boldsymbol{x})\mathrm{c}(\boldsymbol{x}, \boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \tag{91}$$

$$= \mathrm{c}(\boldsymbol{z}, \boldsymbol{z}') \sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} (2\pi)^{-d_x} \iint_{\mathcal{X}} \mathbf{f}_2^{\mathsf{T}}(\boldsymbol{x}')\mathbf{f}_1(\boldsymbol{x})\psi_{\boldsymbol{n}}(\boldsymbol{a})\psi_{-\boldsymbol{n}}(\boldsymbol{b})\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \prod_{j=1}^{d_x} \hat{\mathrm{c}}\big(n_j^2; \nu, \ell_j\big) \tag{92}$$

$$= \mathrm{c}(\boldsymbol{z}, \boldsymbol{z}'; \nu_z, \boldsymbol{\ell}_z)\ (2\pi)^{d_x} \sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \mathrm{FS}_{[\boldsymbol{n},-\boldsymbol{n}]}[\mathbf{f}_2^{\mathsf{T}}\mathbf{f}_1] \prod_{j=1}^{d_x} \hat{\mathrm{c}}\big(n_j^2; \nu, \ell_j\big) \tag{93}$$

$$= (2\pi)^{d_x}\mathrm{c}(\boldsymbol{z}, \boldsymbol{z}'; \nu_z, \boldsymbol{\ell}_z) \sum_{\boldsymbol{n}\in\mathbb{Z}^{d_x}} \mathrm{FS}_{-\boldsymbol{n}}[\mathbf{f}_2]^{\mathsf{T}}\,\mathrm{FS}_{\boldsymbol{n}}[\mathbf{f}_1] \prod_{j=1}^{d_x} \hat{\mathrm{c}}\big(n_j^2; \nu, \ell_j\big) \tag{94}$$

B Proofs

In this section, we include the proofs for Lemma 3.2, Theorem 3.1, and a short lemma on the well-defined-ness of the activation operator.

B.1 Well-defined-ness of the point-wise element-wise activation operator

Lemma B.1 *Let $(\mathcal{X}, \Sigma, \mu_{\mathcal{X}})$ be a finite measure space, i.e. $\mu_{\mathcal{X}}(\mathcal{X}) \leq \infty$, and $\sigma\colon \mathbb{R} \to \mathbb{R}$ a Borel measurable function such that*

$$\sup_{x\in\mathbb{R}} \frac{|\sigma(x)|}{1+|x|} < C \tag{95}$$

for some constant $C \in \mathbb{R}$. Then, the operator $\sigma[\mathrm{f}]\colon L^2(\mathcal{X}) \to L^2(\mathcal{X}) = \sigma \circ \mathrm{f}$ is well defined.

Proof. Remember that a function f is in $L^2(\mathcal{X})$ if, and only if,

$$\int_{\mathcal{X}} |\mathrm{f}(\boldsymbol{x})|^2\,\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}) < \infty. \tag{96}$$

Now, from the linear boundedness condition, we know that for any $\mathrm{f} \in L^2(\mathcal{X})$ and any $\boldsymbol{x} \in \mathcal{X}$:

$$|\sigma(\mathrm{f}(\boldsymbol{x}))| < C(1 + |\mathrm{f}(\boldsymbol{x})|), \tag{97}$$

by squaring both sides and taking integrals,

$$\int_{\mathcal{X}} |\sigma(\mathrm{f}(\boldsymbol{x}))|^2\,\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}) < C^2 \int_{\mathcal{X}} (1 + |\mathrm{f}(\boldsymbol{x})|)^2\,\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}) \tag{98}$$

Now, note that the constant function 1 is in $L^2(\mathcal{X})$ since $\int_{\mathcal{X}} 1\,\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}) = \mu_{\mathcal{X}}(\mathcal{X}) < \infty$ and that $|\mathrm{f}(\cdot)|$ is in $L^2\mathcal{X}$. Thus, from linearity, $1 + |\mathrm{f}(\cdot)|$ is also in $L^2\mathcal{X}$ and $\int_{\mathcal{X}} 1 + |\mathrm{f}(\boldsymbol{x})|\,\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}) < \infty$. Therefore,

$$\int_{\mathcal{X}} |\sigma(\mathrm{f}(\boldsymbol{x}))|^2\,\mathrm{d}\mu_{\mathcal{X}}(\boldsymbol{x}) < \infty. \tag{99}$$

□

B.2 Compositionality of covariance functions

Lemma 3.2. *Let $B_1 \colon L^2\big(\mathcal{X}; \mathbb{R}^d\big) \to L^2\big(\mathcal{X}; \mathbb{R}^J\big)$ be a random operator and $B_2 \colon L^2\big(\mathcal{X}; \mathbb{R}^J\big) \to L^2(\mathcal{X})$ be a centered function-valued Gaussian process. If the following assumptions hold:*

- *For all $\mathbf{f} \in L^2\big(\mathcal{X}; \mathbb{R}^d\big)$ and $\boldsymbol{x} \in \mathcal{X}$, each component of $B_1[\mathbf{f}](\boldsymbol{x}) \in \mathbb{R}^J$ is independent and identically distributed such that the covariance function $\mathbf{C}_{B_1}[\mathbf{f}, \mathbf{g}] = \mathrm{c}_{B_1}[\mathbf{f}, \mathbf{g}]\boldsymbol{I}_J$;*
- *The covariance function of B_2 can expressed, for all $\mathbf{f}, \mathbf{g} \in L^2\big(\mathcal{X}; \mathbb{R}^J\big)$ as $\mathrm{c}_{B_2}[\mathbf{f}, \mathbf{g}] = \mathrm{c}_{B_2}\big[\frac{1}{J}\mathbf{g}^\intercal \mathbf{f}\big]$ and the function $\mathrm{h} \mapsto \mathrm{c}_{B_2}[\mathrm{h}]$ is a continuous map from $L^2(\mathcal{X} \times \mathcal{X})$ to itself.*

Then, $B_2 \circ B_1$ converges in distribution to a function-valued Gaussian process as $J \to \infty$, and

$$\mathrm{c}_{B_2 \circ B_1}[\mathbf{f}_1, \mathbf{f}_2] = \mathrm{c}_{B_2}[\mathrm{c}_{B_1}[\mathbf{f}_1, \mathbf{f}_2]]. \tag{100}$$

Proof. Consider a set of size $N \in \mathbb{N}^+$, $\{(\mathbf{f}_n, \mathrm{h}_n)\}_{n=1}^N \subset L^2\big(\mathcal{X}; \mathbb{R}^d\big) \times L^2(\mathcal{X})$, then define the N-dimensional vector:

$$\boldsymbol{z} \coloneqq [\langle \mathrm{h}_1, (B_2 \circ B_1)[\mathbf{f}_1]\rangle, \ldots, \langle \mathrm{h}_N, (B_2 \circ B_1)[\mathbf{f}_N]\rangle]^\intercal \in \mathbb{R}^N. \tag{101}$$

Additionally, define the function:

$$\bar{\mathrm{c}}_{B_1}[\mathbf{f}_j, \mathbf{f}_k] \colon L^2\big(\mathcal{X}; \mathbb{R}^d\big) \times L^2\big(\mathcal{X}; \mathbb{R}^d\big) \to L^2(\mathcal{X} \times \mathcal{X}) = \frac{1}{J} B_1[\mathbf{f}_k]^\intercal B_1[\mathbf{f}_j]. \tag{102}$$

Then, the conditional random variable $\boldsymbol{z} \mid \{\bar{\mathrm{c}}_{B_1}[\mathbf{f}_i, \mathbf{f}_j]\}_{i,j=1}^N$ is Gaussian distributed with zero mean and covariance:

$$\mathrm{cov}(z_i, z_j \mid \bar{\mathrm{c}}_{B_1}[\mathbf{f}_i, \mathbf{f}_j]) = \langle \mathrm{h}_j, \langle \mathrm{h}_i, \mathrm{c}_{B_2}[\bar{\mathrm{c}}_{B_1}[\mathbf{f}_i, \mathbf{f}_j]]\rangle\rangle. \tag{103}$$

We want to show that every $\boldsymbol{z}$ converges in distribution to a Gaussian distribution when $J \to \infty$, thus, it is useful to remember the following facts:

- **Multivariate Levy's continuity theorem.** A sequence of random variables $\{\boldsymbol{x}_j\}_{j=1}^\infty$ converges to another one $\boldsymbol{x}_\infty$ if and only if the sequence of characteristic functions $\phi_{\boldsymbol{x}_j}(\boldsymbol{t}) = \mathbb{E}[\exp(i \cdot \boldsymbol{t}^\intercal \boldsymbol{x}_j)]$, where $i = \sqrt{-1}$, converges point-wise to $\phi_{\boldsymbol{x}_\infty}$.
- **Characteristic function of a N-dimensional Gaussian distribution.** If $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, then $\phi_{\boldsymbol{x}}(\boldsymbol{t}) = \exp(\boldsymbol{t}^\intercal \boldsymbol{\Sigma} \boldsymbol{t})$ and $\phi_{\boldsymbol{x}}(\boldsymbol{t}) \leq 1$, for all $\boldsymbol{t} \in \mathbb{R}^N$.
- **Strong law of large numbers.** As $J \to \infty$, the random element $\mathbf{K}[\mathbf{f}_i, \mathbf{f}_j]$ converges strongly to the constant $\mathrm{c}_{B_1}[\mathbf{f}_i, \mathbf{f}_j]$, for all $\mathbf{f}_i$.
- **Portmanteau theorem.** Given a sequence of random elements in $\mathcal{H}$ converging in distribution $\{x_i\}_{i=1}^\infty \to x_\infty$, then, $\lim_{i \to \infty} \mathbb{E}[f(x_i)] = \mathbb{E}[f(x_\infty)]$, for all bounded and continuous functions $\mathrm{f} \colon \mathcal{H} \to \mathbb{R}$.

Thus, we begin with the characteristic function of the variable $\boldsymbol{z}$:

$$\phi_{\boldsymbol{z}}(\boldsymbol{t}) = \mathbb{E}[\exp(i \cdot \boldsymbol{t}^\intercal \boldsymbol{z})], \tag{104}$$

by the tower rule, we can write

$$= \mathbb{E}\big[\mathbb{E}[\exp(i \cdot \boldsymbol{t}^\intercal \boldsymbol{z}) \mid \{\bar{\mathrm{c}}_{B_1}[\mathbf{f}_j, \mathbf{f}_k]\}_{j,k=1}^N]\big] = \mathbb{E}[\exp(\boldsymbol{t}^\intercal \boldsymbol{\Sigma} \boldsymbol{t})], \tag{105}$$

where, $[\boldsymbol{\Sigma}]_{jk} = \mathrm{cov}(z_j, z_k \mid \bar{\mathrm{c}}_{B_1}[\mathbf{f}_j, \mathbf{f}_k])$ is a random variable.

Now, because of the continuity of inner products and the assumption that c_{B_2} is continuous, we know that the mapping $\mathrm{h} \mapsto \mathrm{cov}(z_j, z_k \mid \bar{\mathrm{c}}_{B_1}[\mathbf{f}_j, \mathbf{f}_k] = \mathrm{h})$ is continuous. With this we take the limit:

$$\lim_{J \to \infty} \phi_{\boldsymbol{z}}(\boldsymbol{t}) = \lim_{J \to \infty} \mathbb{E}[\exp(\boldsymbol{t}^\intercal \boldsymbol{\Sigma} \boldsymbol{t})]; \tag{106}$$

using the portmanteau theorem, we get that:

$$\lim_{J \to \infty} \phi_{\boldsymbol{z}}(\boldsymbol{t}) = \mathbb{E}\left[\exp\left(\boldsymbol{t}^\intercal \begin{bmatrix} \Big\langle \mathrm{h}_1, \Big\langle \mathrm{h}_1, \mathrm{c}_{B_2}\Big[\lim\limits_{J \to \infty} \bar{\mathrm{c}}_{B_1}[\mathbf{f}_1, \mathbf{f}_1]\Big]\Big\rangle\Big\rangle & \cdots \\ \vdots & \ddots \end{bmatrix} \boldsymbol{t}\right)\right], \tag{107}$$

and, finally, by the strong law of large numbers, we write the expectation as:

$$\lim_{J\to\infty} \phi_{\boldsymbol{z}}(\boldsymbol{t}) = \exp\left(\boldsymbol{t}^{\intercal} \begin{bmatrix} \langle \mathrm{h}_1, \langle \mathrm{h}_1, \mathrm{c}_{B_2}[\mathrm{c}_{B_1}[\mathbf{f}_1, \mathbf{f}_1]]\rangle\rangle & \cdots \\ \vdots & \ddots \end{bmatrix} \boldsymbol{t}\right). \tag{108}$$

Therefore, we have shown that $\boldsymbol{z}$ converges to a centered Gaussian distribution with:

$$\operatorname{cov}(z_j, z_k) = \operatorname{cov}(z_j, z_k \mid \bar{\mathrm{c}}_{B_1}[\mathbf{f}_j, \mathbf{f}_k] = \mathrm{c}_{B_1}[\mathbf{f}_j, \mathbf{f}_k]) = \langle \mathrm{h}_k, \langle \mathrm{h}_j, \mathrm{c}_{B_2}[\mathrm{c}_{B_1}[\mathbf{f}_j, \mathbf{f}_k]]\rangle\rangle \tag{109}$$

Since the set $\{\mathbf{f}_n, \mathrm{h}_n\}_{n=1}^{N}$ is arbitrary, we have shown that $B_2 \circ B_1$ is a centered function-valued Gaussian process with covariance function $\mathrm{c}_{B_2}[\mathrm{c}_{B_1}[\cdot,\cdot]]$. □

B.3 Infinite-width neural operators are Gaussian processes

Definition B.1 (Iterated convergence in distribution)**.** Let $X_{\boldsymbol{i}}$ be a random variable for each $\boldsymbol{i} = [i_1, \cdots, i_k] \subset \mathbb{N}^{+}$. The iterated limit

$$\lim_{i_k\to\infty}\left(\lim_{i_{k-1}\to\infty}\left(\cdots \lim_{i_1\to\infty}\left(X_{\boldsymbol{i}}\right)\cdots\right)\right) \tag{110}$$

whenever exists, is defined as the iterated limit in distribution. That is, suppose there are random variables $X_{[\infty, i_2, \cdots, i_k]}, X_{[\infty,\infty,\cdots,i_k]}, \cdots, X_{[\infty,\infty,\cdots,\infty]}$ such that for every $i_2, \cdots, i_k$

$$X_{[i_1, i_2, i_3, \cdots, i_{k-1}, i_k]} \xrightarrow{d} X_{[\infty, i_2, i_3, \cdots, i_{k-1}, i_k]} \qquad \text{as } i_1 \to \infty. \tag{111}$$

$$X_{[\infty, i_2, i_3, \cdots, i_{k-1}, i_k]} \xrightarrow{d} X_{[\infty, \infty, i_3, \cdots, i_{k-1}, i_k]} \tag{112}$$

$$\vdots$$

$$X_{[\infty, \infty, \infty, \cdots, \infty, i_k]} \xrightarrow{d} X_{[\infty, \infty, \infty, \cdots, \infty, \infty]} \tag{113}$$

Then we define the iterated limit of $X_{\boldsymbol{i}}$ as

$$\lim_{i_k\to\infty}\left(\lim_{i_{k-1}\to\infty}\left(\cdots \lim_{i_1\to\infty}\left(X_{\boldsymbol{i}}\right)\cdots\right)\right) = X_{[\infty,\infty,\cdots,\infty]} \tag{114}$$

Theorem 3.1. *Let $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ be a measurable space and let $\mathcal{H}(\mathcal{X};\mathbb{R}^J) \subset L^2(\mathcal{X};\mathbb{R}^J)$ be an RKHS for any $J \in \mathbb{N}^{+}$. Then, for a given depth $D \in \mathbb{N}^{+}$, consider a vector positive integers $\boldsymbol{J} = [J_0, J_1, \ldots, J_{D-1}, 1]^{\intercal} \in \mathbb{N}^{D+1}$ and a $\boldsymbol{J}$-indexed neural operators $Z_{\boldsymbol{J}}^{(D)}$ of depth D:*

$$Z_{\boldsymbol{J}}^{(D)} := H^{(D)} \circ \sigma \circ Z_{\boldsymbol{J}}^{(D-1)} \in (\mathcal{X} \to \mathbb{R}^{J_0}) \to (\mathcal{X} \to \mathbb{R}), \tag{115}$$

where,

$$Z_{\boldsymbol{J}}^{(1)} := H^{(1)} \in L^2(\mathcal{X};\mathbb{R}^{J_0}) \to \mathcal{H}(\mathcal{X};\mathbb{R}^{J_1}), \textit{ and} \tag{116}$$

$$H^{(\ell)} := (A_{\mathbf{K}^{(\ell)}} + \boldsymbol{W}^{(\ell)}) \in L^2(\mathcal{X};\mathbb{R}^{J_{\ell-1}}) \to \mathcal{H}(\mathcal{X};\mathbb{R}^{J_\ell}), \tag{117}$$

with $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{J_\ell \times J_{\ell-1}}$, and $\mathbf{K}^{(\ell)} \in \mathcal{H}(\mathcal{X}\times\mathcal{X};\mathbb{R}^{J_\ell \times J_{\ell-1}})$.
When all parameters are independently distributed a priori *according to*

$$\boldsymbol{W}^{(\ell)} \sim \mathcal{N}(\mathbf{0}, \sigma_\ell^2/J_{\ell-1}\boldsymbol{I}), \textit{ and } \mathbf{K}^{(\ell)} \sim \mathrm{GP}(\mathbf{0}, \mathrm{c}_{\mathrm{k}^{(\ell)}}/J_{\ell-1}\boldsymbol{I}), \qquad \textit{for } \ell \in \{1, \ldots, d\}, \tag{118}$$

then, the iterated limit $\lim\limits_{J_{D-1}\to\infty} \cdots \lim\limits_{J_1\to\infty} Z_{\boldsymbol{J}}^{(D)}$, in the sense of Definition B.1, is equal to a function-valued GP $Z_{\infty}^{(D)} \sim \mathrm{GP}(0, \mathrm{c}_\infty)$, where $\mathrm{c}_\infty[\mathbf{f}, \mathbf{g}]$ is available in closed-form.

Proof. First, we note from Section 3.1 that the covariances $\mathrm{c}_{\boldsymbol{W}^{(\ell)}}[\mathbf{f}, \mathbf{g}]$ and $\mathrm{c}_{A_{\mathbf{K}^{(\ell)}}}[\mathbf{f}, \mathbf{g}]$ are equal to:

$$\mathbf{C}_{\boldsymbol{W}^{(\ell)}}[\mathbf{f}, \mathbf{g}] = \sigma_\ell^2 \frac{1}{J_{\ell-1}} \mathbf{g}^{\intercal}\mathbf{f}\boldsymbol{I}_{J_\ell}, \text{ and,} \mathbf{C}_{A_{\mathbf{K}^{(\ell)}}}[\mathbf{f}, \mathbf{g}] = A_{\mathrm{c}_{\mathrm{k}^{(\ell)}}}\left[\frac{1}{J_{\ell-1}}\mathbf{g}^{\intercal}\mathbf{f}\right]\boldsymbol{I}_{J_\ell}, \tag{119}$$

such that both depend on the empirical covariance $\frac{1}{J_{\ell-1}}\mathbf{g}^\intercal\mathbf{f}$, for all $\mathbf{f},\mathbf{g} \in L^2\big(\mathcal{X};\mathbb{R}^{J_{\ell-1}}\big)$ and $\ell \in \mathbb{N}^+$. Therefore, since $H^{(\ell)}$ is the sum of these two independent function-valued Gaussian processes, we have that $H^{(\ell)} \sim \mathrm{GP}\big(0, \mathrm{c}^{(\ell|\ell-1)}\boldsymbol{I}_{J_\ell}\big)$ such that:

$$\mathrm{c}^{(\ell|\ell-1)}[\mathbf{f},\mathbf{g}] = \mathrm{c}^{(\ell|\ell-1)}[\mathbf{g}^\intercal\mathbf{f}/J_{\ell-1}] = A_{\mathrm{c}_{\mathrm{k}^{(\ell)}}}[\mathbf{g}^\intercal\mathbf{f}/J_{\ell-1}] + \sigma_\ell^2\mathbf{g}^\intercal\mathbf{f}/J_{\ell-1} \tag{120}$$

With this in mind, we proceed the proof by induction on the depth D.

Base case. For the base case $D = 1$, we consider the operator $Z_{\boldsymbol{J}}^{(1)}$. Therefore, there are no limits to consider in this step. Nonetheless, as discussed in the previous paragraph, this quantity is a function-valued GP with covariance:

$$\mathbf{C}^{(1)}[\mathbf{f},\mathbf{g}] = \mathrm{c}^{(1|0)}[\mathbf{f},\mathbf{g}]\boldsymbol{I}_{J_1} = A_{\mathrm{c}_{\mathrm{k}^{(1)}}}[\mathbf{g}^\intercal\mathbf{f}/J_0] + \sigma_1^2\mathbf{g}^\intercal\mathbf{f}/J_0. \tag{121}$$

Therefore, our claim is proven.

Inductive step. Our inductive hypothesis says that, for a specific $\ell \in \mathbb{N}^+$, we have that the iterated limit $\lim\limits_{J_{\ell-1}\to\infty} \cdots \lim\limits_{J_1\to\infty} Z_{\boldsymbol{J}}^{(\ell)}$ converges in distribution to a $Z_\infty^{(\ell)} \sim \mathrm{GP}\big(\mathbf{0}, \mathrm{c}^{(\ell)}\boldsymbol{I}_{J_\ell}\big)$.

As a first step, we would like to prove that

$$\lim_{J_{\ell-1}\to\infty} \cdots \lim_{J_1\to\infty} H^{(\ell+1)} \circ \sigma \circ Z_{\boldsymbol{J}}^{(\ell)} \tag{122}$$

converges in distribution to

$$H^{(\ell+1)} \circ \sigma \circ Z_\infty^{(\ell)}. \tag{123}$$

Consider an arbitrary set of size $N \in \mathbb{N}^+$,

$$(\mathcal{F},\mathcal{H}) = \{(\mathbf{f}_1,\mathbf{h}_1),\ldots,(\mathbf{f}_N,\mathbf{h}_N)\} \subset L^2\big(\mathcal{X};\mathbb{R}^{J_0}\big) \times L^2\big(\mathcal{X};\mathbb{R}^{J_{\ell+1}}\big), \tag{124}$$

and define the variables $\boldsymbol{z}[\mathcal{F},\mathcal{H}] \in \mathbb{R}^N$ and $\mathbf{Z}_{\boldsymbol{J}}^{(\ell)}[\mathcal{F}] \in L^2\big(\mathcal{X};\mathbb{R}^{N\times J_\ell}\big)$ such that:

$$[\boldsymbol{z}[\mathcal{F},\mathcal{H}]]_n \coloneqq \Big\langle \mathbf{h}_n, H^{(\ell+1)}\Big[\sigma(Z_{\boldsymbol{J}}^{(\ell)}[\mathbf{f}_n])\Big]\Big\rangle, \text{ and,} \tag{125}$$

$$[\mathbf{Z}_{\boldsymbol{J}}^{(\ell)}[\mathcal{F}]]_n \coloneqq Z_{\boldsymbol{J}}^{(\ell)}[\mathbf{f}_n]. \tag{126}$$

By definition, $\boldsymbol{z}[\mathcal{F},\mathcal{H}]$ conditioned on $\mathbf{Z}_{\boldsymbol{J}}^{(\ell)}[\mathcal{F}]$ follows a multivariate centered Gaussian distribution $\mathcal{N}\big(\mathbf{0}, \boldsymbol{\Sigma}\big(\mathbf{Z}_{\boldsymbol{J}}^{(\ell)}[\mathcal{F}]\big)\big)$ with covariance matrix:

$$[\boldsymbol{\Sigma}(\boldsymbol{A})]_{jk} = \big\langle \mathbf{h}_k, \big\langle \mathbf{h}_j, \mathrm{c}_{H^{(\ell+1)}}[\sigma([\boldsymbol{A}]_k)^\intercal\sigma([\boldsymbol{A}]_j)/J_\ell]\boldsymbol{I}_{J_{\ell+1}}\big\rangle\big\rangle. \tag{127}$$

Thus, by the tower rule, the characteristic function of the marginal distribution of $\boldsymbol{z}[\mathcal{F},\mathcal{H}]$ is:

$$\phi_{\boldsymbol{z}[\mathcal{F},\mathcal{H}]}(\boldsymbol{t}) = \mathbb{E}\Big[\mathbb{E}[\exp(i\boldsymbol{t}^\intercal\boldsymbol{z}[\mathcal{F},\mathcal{H}]) \mid \mathbf{Z}_{\boldsymbol{J}}^{(\ell)}[\mathcal{F}]]\Big] = \mathbb{E}\Big[\exp\Big(\boldsymbol{t}^\intercal\boldsymbol{\Sigma}\big(\mathbf{Z}_{\boldsymbol{J}}^{(\ell)}[\mathcal{F}]\big)\boldsymbol{t}\Big)\Big]. \tag{128}$$

Now, consider the point-wise convergence of the characteristic function:

$$\lim_{J_{\ell-1}\to\infty} \cdots \lim_{J_1\to\infty} \phi_{\boldsymbol{z}[\mathcal{F},\mathcal{H}]}(\boldsymbol{t}) =: \phi_\infty(\boldsymbol{t}) \tag{129}$$

Using the portmanteau theorem and continuity of $\boldsymbol{\Sigma}(\cdot)$, we have that:

$$\phi_\infty(\boldsymbol{t}) = \mathbb{E}\Bigg[\exp\Bigg(\boldsymbol{t}^\intercal\boldsymbol{\Sigma}\Big(\lim_{J_{\ell-1}\to\infty} \cdots \lim_{J_1\to\infty} \mathbf{Z}_{\boldsymbol{J}}^{(\ell)}[\mathcal{F}]\Big)\boldsymbol{t}\Bigg)\Bigg]. \tag{130}$$

Now, our inductive hypothesis says that $Z_{\boldsymbol{J}}^{(\ell)}$ converges in distribution to a function-valued Gaussian process $Z_\infty^{(\ell)}$ with each output being i.i.d. With this fact, we can conclude that $\mathbf{Z}_{\boldsymbol{J}}^{(\ell)}[\mathcal{F}]$ also converges in distribution to the corresponding variable: $[\mathbf{Z}_\infty^{(\ell)}[\mathcal{F}]]_n \coloneqq Z_\infty^{(\ell)}[\mathbf{f}_n]$. This means that:

$$\phi_\infty(\boldsymbol{t}) = \mathbb{E}\Big[\exp\Big(\boldsymbol{t}^\intercal\boldsymbol{\Sigma}\Big(\mathbf{Z}_\infty^{(\ell)}[\mathcal{F}]\Big)\boldsymbol{t}\Big)\Big], \tag{131}$$

which is the characteristic function of a variable defined as:

$$[\tilde{\boldsymbol{z}}[\mathcal{F},\mathcal{H}]]_n := \Big\langle \mathbf{h}_n, H^{(\ell+1)}\Big[\sigma(Z_\infty^{(\ell)}[\mathbf{f}_n])\Big]\Big\rangle. \tag{132}$$

Therefore, $\boldsymbol{z}[\mathcal{F},\mathcal{H}]$ iteratively converges in distribution to $\tilde{\boldsymbol{z}}[\mathcal{F},\mathcal{H}]$, as $J_\ell \to \infty$ for every $\ell \leq \ell$. Since the set $(\mathcal{F},\mathcal{H})$ is arbitrary, we can conclude that $(H^{(\ell+1)} \circ \sigma \circ Z_{\boldsymbol{J}}^{(\ell)})$ also converges in distribution to $(H^{(\ell+1)} \circ \sigma \circ Z_\infty^{(\ell)})$ as a random operator.

From the induction step, we know that the entries in $(\sigma \circ Z_\infty^{(\ell)})$ are i.i.d. since the entries of $\sigma \circ Z_\infty^{(\ell)}$ are also i.i.d. Therefore, we use Lemma 3.2 to show that $\lim\limits_{J_\ell\to\infty} (H^{(\ell+1)} \circ \sigma \circ Z_\infty^{(\ell)})$ converges in distribution to a function-valued Gaussian process with covariance function

$$\mathbf{C}^{(\ell+1)}[\mathbf{f},\mathbf{g}] = \mathrm{c}_{H^{(\ell+1)}}\big[\mathrm{c}_{(\sigma\circ Z_\infty^{(\ell)})}\big]\boldsymbol{I}_{J_{\ell+1}} = \mathrm{c}^{(\ell+1|\ell)}\big[\mathrm{c}_\sigma\big[\mathrm{c}^{(\ell)}[\mathbf{f},\mathbf{g}]\big]\big]\boldsymbol{I}_{J_{\ell+1}}. \tag{133}$$

Therefore, we just proved by induction that the iterated limit $\lim\limits_{J_{D-1}\to\infty} \cdots \lim\limits_{J_1\to\infty} Z_{\boldsymbol{J}}^{(D)}$ converges in distribution to a $Z_\infty^{(D)} \sim \mathrm{GP}(\mathbf{0}, \mathrm{c}_\infty \boldsymbol{I}_{J_\ell})$ and this covariance function is equal to:

$$\mathrm{c}_\infty[\mathbf{f},\mathbf{g}] = \mathrm{c}^{(d)}[\mathbf{f},\mathbf{g}] = \mathrm{c}^{(d|d-1)}[\mathrm{c}^{(d-1|d-2)}[\cdots \mathrm{c}^{(2|1)}[\mathrm{c}^{(1)}[\mathbf{f},\mathbf{g}]]\cdots] \tag{134}$$

□

C Experimental details

In this section, we describe the setup for our experiments. As previously mentioned, all experiments were run in a desktop machine with a 3.8GHz Intel Core i7-9800X CPU and a 24GB NVIDIA Titan RTX (TU102) GPU. More details for each experiment can be found below.

C.1 Empirical demonstration of results

For both experiments, the input function $\mathrm{f}\colon \mathbb{T} \to \mathbb{R}$ has band-limit $B = 3$, with its output values $\mathrm{f}(x)$ sampled from a uniform distribution $\mathcal{U}(-1,1)$. In other words, we can express this band-limited function as:

$$\mathrm{f}(x) = \frac{1}{7}\sum_{s=-3}^{3} f_s \sum_{s'=-3}^{3} \psi_{s'}\Big(x - \frac{2\pi}{7}s\Big), \tag{135}$$

where each $f_s \sim \mathcal{U}(-1,1)$ is independent and identically distributed.

In the first experiment of Fig. 2, we construct the operator layer H under the usual formulation:

$$H[\mathrm{f}](x)\colon L^2(\mathbb{T}) \to L^2(\mathbb{T}) = A_{\mathrm{k}}[\mathrm{f}](x) + w\mathrm{f}(x), \tag{136}$$

where $w \sim \mathcal{N}(0,1)$ and k follows the band-limited Gaussian process distribution (Section 4.1 and Appendix A.1) with with band-limit $B = 3$ and variance $\sigma^2 = 1/7$. Then, the operator on f is evaluated at zero $H[\mathrm{f}](0)$ with increasing sample sizes.

For the second experiment of Fig. 1, we construct the single-layer neural operator:

$$Z[\mathrm{f}](x)\colon L^2(\mathbb{T}) \to L^2\big(\mathbb{T};\mathbb{R}^J\big) \to L^2(\mathbb{T}) = (\boldsymbol{w}_2^\intercal \circ \mathrm{ReLU} \circ (A_{\mathbf{k}} + \boldsymbol{w}_1))[\mathrm{f}](x), \tag{137}$$

where J is the width of the hidden layer, and $\boldsymbol{w}_2 \sim \mathcal{N}(0, 1/J)$, $\boldsymbol{w}_2 \sim \mathcal{N}(0,1)$, and $\mathbf{k}$ follows an i.i.d. band-limited Gaussian process distribution (Section 4.1 and Appendix A.1) with band-limit $B = 3$ and variance $\sigma^2 = 1/7$. For varying widths $J \in \{1, 10, 100, 1000\}$, we evaluate 10,000 samples of the operator on f at zero $Z[\mathrm{f}](0)$ and show the density of the empirical distribution using kernel density estimation (KDE) with a Gaussian kernel.

These experiments are implemented in the file `experiments/fno_limit.ipynb`.

C.2 Regression

We consider FNOs of increasing width, $J \in \{1, 10, 100\}$ and $J \in \{1, 3, 10, 100, 500\}$ for the synthetic and 1D Burgers' respectively, as well as ∞-FNOs, both with increasing kernel band-limits $B \in \{1, 5, 20\}$. These single-layer neural operators are constructed as:

$$Z_{J,B}[\mathrm{f}](x) \colon L^2(\mathbb{T}) \to L^2\big(\mathbb{T}; \mathbb{R}^J\big) \to L^2(\mathbb{T}) = (\boldsymbol{w}_2^\intercal \circ \mathrm{ReLU} \circ (A_{\mathbf{k}} + \boldsymbol{w}_1))[\mathrm{f}](x), \tag{138}$$

where J is the width of the hidden layer, and $\boldsymbol{w}_2 \sim \mathcal{N}(0, 1/J)$, $\boldsymbol{w}_2 \sim \mathcal{N}(0, 1)$, and $\mathbf{k}$ follow an i.i.d. band-limited Gaussian process distribution (Section 4.1 and Appendix A.1) with variance $\sigma^2 = 1/(2B + 1)$.

The hyperparameters of the ∞-FNO are estimated using L-BFGS, while the parameters of the FNOs are optimized with Adam using a step size of 0.001. We evaluate all models using 5-fold cross-validation and report the average and standard deviation of the empirical L^2 norm of the prediction error. For ∞-FNOs, we use the posterior mean as the prediction.

This experiment is implemented in the file `experiments/train.py`.

Synthetic regression

We start by defining the ground truth Fourier neural operator (FNO) which will generate our training and test data:

$$Z_{\mathrm{true}}[\mathrm{f}](x) \colon L^2(\mathbb{T}) \to L^2(\mathbb{T}) \to L^2(\mathbb{T}) = (w_2 \circ \mathrm{ReLU} \circ (A_{\mathrm{k}} + w_1))[\mathrm{f}](x), \tag{139}$$

where the hidden layer's width is 1 and the band-limit of k is equal to 5. Next, we sample $n = 100$ input functions $\mathrm{f}_i \colon \mathbb{T} \to \mathbb{R}$ with the same band-limit $B = 5$ and uniformily-distributed outputs $\mathcal{U}(-1, 1)$, so that we have:

$$\mathrm{f}_i(x) = \frac{1}{11} \sum_{s=-5}^{5} f_{is} \sum_{s'=-5}^{5} \psi_{s'}\left(x - \frac{2\pi}{11} s\right), \tag{140}$$

where each $f_{is} \sim \mathcal{U}(-1, 1)$ is independent and identically distributed. We then compute $Z_{\mathrm{true}}[\mathrm{f}_i]$ on an equally spaced grid given by $\{-5\frac{2\pi}{11}, \ldots, 5\frac{2\pi}{11}\} \subset \mathbb{R}^{11}$.

1D Burgers' equation

This dataset is provided from PDEBench (Takamoto et al., 2022), which includes solutions to the 1D Burgers' equation:

$$\frac{\partial}{\partial t}\mathrm{u}(t, x) + \frac{1}{2}\frac{\partial}{\partial x}\mathrm{u}^2(t, x) = \frac{\nu}{\pi}\frac{\partial^2}{\partial x^2}\mathrm{u}(t, x), \tag{141}$$

where $x \in (0, 1)$ and $t \in (0, 2]$ are independent variables and ν is the diffusion coefficient.

The regression task is set up with $\nu = 0.002$ and a collection of initial conditions $\{\mathrm{u}(0, \cdot) = \mathrm{f}_i\}_{i=1}^n$ and their respective end states $\{\mathrm{u}(2, \cdot) = \mathrm{g}_i\}_{i=1}^n$. Due to memory constraints when creating the covariance matrices for ∞-FNO, we subsample the original dataset to $n = 100$ functions and a grid size of $m = 103$. The original data can be downloaded at `https://darus.uni-stuttgart.de/api/access/datafile/268193`.