

# Are Neural Operators Really Neural Operators? Frame Theory Meets Operator Learning

F. Bartolucci and E. de Bézenac and B. Raonic and R. Molinaro and  
S. Mishra and R. Alaifari

Research Report No. 2023-21  
May 2023

Seminar für Angewandte Mathematik  
Eidgenössische Technische Hochschule  
CH-8092 Zürich  
Switzerland

---

---

# Are Neural Operators Really Neural Operators? Frame Theory Meets Operator Learning

---

**Francesca Bartolucci**  
TU Delft

**Emmanuel de Bézenac**  
ETH Zurich

**Bogdan Raonic**  
ETH Zurich

**Roberto Molinaro**  
ETH Zurich

**Siddhartha Mishra**  
ETH Zurich

**Rima Alaifari**  
ETH Zurich

## Abstract

Recently, there has been significant interest in operator learning, i.e. learning mappings between infinite-dimensional function spaces. This has been particularly relevant in the context of learning partial differential equations from data. However, it has been observed that proposed models may not behave as operators when implemented on a computer, questioning the very essence of what *operator learning* should be. We contend that in addition to defining the operator at the continuous level, some form of continuous-discrete equivalence is necessary for an architecture to genuinely learn the underlying operator, rather than just discretizations of it. To this end, we propose to employ *frames*, a concept in applied harmonic analysis and signal processing that gives rise to exact and stable discrete representations of continuous signals. Extending these concepts to operators, we introduce a unifying mathematical framework of *Representation equivalent Neural Operator (ReNO)* to ensure operations at the continuous and discrete level are equivalent. Lack of this equivalence is quantified in terms of aliasing errors. We analyze various existing operator learning architectures to determine whether they fall within this framework, and highlight implications when they fail to do so.

## 1 Introduction

Operators are mappings between infinite-dimensional function spaces. Prominent examples are *solution operators* for ordinary and partial differential equations (PDEs) [8] which map function space inputs such as initial and boundary data to the function space valued PDE solution. They are also the natural mathematical framework for inverse problems, both in the context of PDEs [10] and in (medical) imaging [2], where the object of interest is the *inverse operator* which maps observables to the underlying material property/image that needs to be inferred or reconstructed.

Given the prohibitive cost of traditional physics based algorithms for approximating operators, particularly those associated with PDEs, increasing attention is being paid in recent years to machine-learning based *operator approximation*. Such techniques for learning operators from data fall under the genre of *operator learning*. A (by no means comprehensive) list of examples for operator learning architectures include operator networks [5], DeepONets [19], Graph neural operators [17], multipole neural operators [18], PCA-Nets [3], Fourier neural operators (FNO) [16], VIDON [20], spectral neural operators [9], LOCA [12], NOMAD [23] and transformer based operator learning models [4].

However, there is still a lack of clarity on what constitutes *operator learning*? Clearly, an operator learning framework, or a *neural operator* in the nomenclature of [14], should be able to *process functions as inputs and outputs*. On the other hand, functions are continuous objects and in practice, one may not have access to functions either at the input or output level. Instead, one can only access

functions and perform computations with them through their *discrete representations* such as point values on a grid, cell averages or in general, coefficients of an underlying basis. Hence, in practice, neural operators have to map discrete inputs into discrete outputs. This leads to a dichotomy, i.e., neural operators need functions as inputs/outputs but only have access to their discrete representations. Consequently, at any finite resolution, possible mismatches between the discretizations and the continuous versions of neural operators can lead to inconsistencies in the underlying function spaces. These inconsistency errors can propagate through the networks and may mar the performance of these algorithms. Moreover, important properties of the underlying operator, such as symmetries and conservation laws, hold at the continuous level. Inconsistent discretizations may not preserve these structural properties of the operator, leading to symmetry breaking etc. with attendant adverse consequences for operator approximation.

Addressing this inconsistency between neural operators and their discretizations is the central point of this article. To this end, we revisit and adapt existing notions of *continuous-discrete equivalence (CDE)* in signal processing, applied harmonic analysis and numerical analysis, by resorting to *frame theory* and the associated questions of sampling and interpolation. Roughly speaking, continuous-discrete equivalence aims at finding mathematical frameworks in which the continuous objects (functions) can be completely (uniquely and stably) recovered from their discrete representations (point evaluations, basis coefficients, etc.) at any resolution. Consequently, working with discrete values is tantamount to accessing the underlying continuous object. This equivalence is leveraged to define a *class of neural operators*, for which the discrete input-output representations or their continuous function space realizations are *equivalent*. More concretely, our contributions are

- We provide a novel, very general unifying mathematical formalism to characterize a class of neural operators such that an appropriately defined notion of continuous-discrete equivalence holds for them. These neural operators are termed as *Representation equivalent Neural Operators* or ReNOs. Our definition results in an automatically consistent function space formulation for ReNOs.
- We analyze existing operator learning architectures to find whether they are ReNOs or not.
- Synthetic numerical experiments are presented to illustrate the continuous-discrete equivalence for ReNOs and point out the practical consequences for not respecting it.

## 2 Continuous-Discrete Equivalences

We start by revisiting concepts on the equivalence of functions and their discrete representations. This is the very essence of *frame theory*, widely used in signal processing and applied harmonic analysis.

**Equivalence between functions and their point samples.** For simplicity of exposition, we start with univariate functions  $f \in L^2(\mathbb{R})$  and the following question: Can a function be uniquely and stably recovered from its values, *sampled* from equispaced grid points  $(f(nT))_{n \in \mathbb{Z}}$ ? The classical *Whittaker-Shannon-Kotel'nikov (WSK) sampling theorem* [24], which lies at the heart of digital-to-analog conversion, answers this question in the affirmative when the underlying function  $f \in \mathcal{B}_\Omega$ , i.e., it belongs to the *Paley-Wiener space* of *band-limited functions*  $\mathcal{B}_\Omega = \{f \in L^2(\mathbb{R}) : \text{supp } \hat{f} \subseteq [-\Omega, \Omega]\}$ , for some  $\Omega > 0$ ,  $\hat{f}$  the Fourier transform of  $f$  and *sampling rate*  $1/T \geq 2\Omega$ , with  $2\Omega$  termed as the *Nyquist rate*. The corresponding reconstruction formula is,

$$f(x) = 2T\Omega \sum_{n \in \mathbb{Z}} f(nT) \text{sinc}(2\Omega(x - nT)), \quad \text{sinc}(x) = \sin(\pi x)/(\pi x). \quad (2.1)$$

We note that the sequence of functions  $\{\phi_n(x) = \text{sinc}(2\Omega x - n)\}_{n \in \mathbb{Z}}$  constitutes an *orthonormal basis* for  $\mathcal{B}_\Omega$  and denote by  $\mathcal{P}_{\mathcal{B}_\Omega} : L^2(\mathbb{R}) \rightarrow \mathcal{B}_\Omega$  the *orthogonal projection operator* onto  $\mathcal{B}_\Omega$ ,

$$\mathcal{P}_{\mathcal{B}_\Omega} f = \sum_{n \in \mathbb{Z}} \langle f, \phi_n \rangle \phi_n = \sum_{n \in \mathbb{Z}} f\left(\frac{n}{2\Omega}\right) \phi_n, \quad (2.2)$$

where the last equality is a consequence of  $\mathcal{B}_\Omega$  being a reproducing kernel Hilbert space with kernel  $\mathcal{K}_\Omega(x, y) = \text{sinc}(2\Omega(x - y))$ . Hence,  $\mathcal{P}_{\mathcal{B}_\Omega} f$  corresponds to the right-hand side of (2.1) for  $T = 1/2\Omega$  and formula (2.1) is exact if and only if  $f \in \mathcal{B}_\Omega$ , i.e.,  $f = \mathcal{P}_{\mathcal{B}_\Omega} f \iff f \in \mathcal{B}_\Omega$ .

What happens when we sample a function at a sampling rate below the Nyquist rate, i.e. when  $1/T < 2\Omega$ ? Alternatively, what is the effect of approximating a function  $f \notin \mathcal{B}_\Omega$  by  $\mathcal{P}_{\mathcal{B}_\Omega} f$ ? Again, sampling theory provides an answer in the form of the *aliasing error*:

**Definition 2.1.** [1, §5] The *aliasing error function*  $\varepsilon(f)$  and the corresponding *aliasing error* of  $f \in L^2(\mathbb{R})$  for sampling at the rate  $2\Omega$  are given by

$$\varepsilon(f) = f - \mathcal{P}_{\mathcal{B}_\Omega} f, \quad \text{and} \quad \|\varepsilon(f)\|_2 = \|f - \mathcal{P}_{\mathcal{B}_\Omega} f\|_2.$$

If the aliasing error  $\varepsilon(f)$  is zero, i.e. if  $f \in \mathcal{B}_\Omega$ , we say that there is a *continuous-discrete equivalence (CDE)* between  $f$  and its samples  $\{f(n/2\Omega)\}_{n \in \mathbb{Z}}$ .

**Continuous-Discrete Equivalence in Hilbert spaces.** Next, we generalize the above described concept of continuous-discrete equivalence to any separable Hilbert space  $\mathcal{H}$ , with inner product  $\langle \cdot, \cdot \rangle$  and norm  $\|\cdot\|$ . A countable sequence of vectors  $\{f_i\}_{i \in I}$  in  $\mathcal{H}$  is a *frame* for  $\mathcal{H}$  if there exist constants  $A, B > 0$  such that for all  $f \in \mathcal{H}$

$$A\|f\|^2 \leq \sum_{i \in I} |\langle f, f_i \rangle|^2 \leq B\|f\|^2. \quad (2.3)$$

Clearly, an orthonormal basis for  $\mathcal{H}$  is an example of a frame with  $A = B = 1$ . The bounded operator  $T: \ell^2(I) \rightarrow \mathcal{H}$ ,  $T(\{c_i\}_{i \in I}) = \sum_{i \in I} c_i f_i$ , is called *synthesis operator* and its adjoint  $T^*: \mathcal{H} \rightarrow \ell^2(I)$ ,  $T^* f = \{\langle f, f_i \rangle\}_{i \in I}$ , which extracts frame coefficients from an underlying function is called *analysis operator*. By composing  $T$  and  $T^*$ , we obtain the *frame operator*  $S := TT^*$ , which is an invertible, self-adjoint and positive operator. Furthermore, the pseudo-inverse of the synthesis operator is given by  $T^\dagger: \mathcal{H} \rightarrow \ell^2(I)$ ,  $T^\dagger f = \{\langle f, S^{-1} f_i \rangle\}_{i \in I}$ . With these concepts, one can introduce the most prominent result in frame theory [6], the *frame decomposition theorem*, which shows that every element in  $\mathcal{H}$  can be *uniquely and stably* reconstructed from its frame coefficients by means of the reconstruction formula

$$f = TT^\dagger f = \sum_{i \in I} \langle f, S^{-1} f_i \rangle f_i = \sum_{i \in I} \langle f, f_i \rangle S^{-1} f_i, \quad (2.4)$$

where the series converge unconditionally. Formula (2.4) is clearly a generalization of reconstruction formula (2.1). However, it is worth pointing out that, in general, the coefficients in (2.4) are not necessarily point samples of the underlying function  $f$ , but more general frame coefficients.

In general, it may not be possible to access all the frame coefficients to reconstruct a function in a Hilbert space. Instead, just like in the case of reconstructing functions from point samples, one will need to consider approximations to this idealized situation. This is best encapsulated by the notion of a *frame sequence*, i.e., a countable sequence  $\{v_i\}_{i \in I} \in \mathcal{H}$  which is a frame for its closed linear span, i.e. for  $\text{span}\{v_i : i \in I\}$ . With this notion, we are in a position to generalize aliasing errors and the CDE, to arbitrary Hilbert spaces. Let  $\mathcal{H}$  be a separable Hilbert space and let  $\{v_i\}_{i \in I}$  be a frame sequence for  $\mathcal{H}$  with  $\mathcal{V} = \overline{\text{span}\{v_i : i \in I\}}$  and frame operator  $S: \mathcal{V} \rightarrow \mathcal{V}$ . Then, the orthogonal projection of  $\mathcal{H}$  onto  $\mathcal{V}$  is given by  $\mathcal{P}_\mathcal{V} f = \sum_{i \in I} \langle f, v_i \rangle S^{-1} v_i$ . Thus, formula (2.4) holds and the function  $f$  can be uniquely and stably recovered from its frame coefficients if and only if  $f \in \mathcal{V}$ . If  $f \notin \mathcal{V}$ , reconstructing  $f$  from the corresponding frame coefficients results in an *aliasing error*:

**Definition 2.2.** The *aliasing error function*  $\varepsilon(f)$  and the resulting *aliasing error*  $\|\varepsilon(f)\|$  of  $f \in \mathcal{H}$  for the frame sequence  $\{v_i\}_{i \in I} \subseteq \mathcal{V}$  are given by

$$\varepsilon(f) = f - \mathcal{P}_\mathcal{V} f, \quad \|\varepsilon(f)\| = \|f - \mathcal{P}_\mathcal{V} f\|.$$

If the aliasing error  $\varepsilon(f)$  is zero, i.e. if  $f \in \mathcal{V}$ , we say that there is a *continuous-discrete equivalence (CDE)* between  $f$  and its frame coefficients  $\{\langle f, v_i \rangle\}_{i \in I}$ .

### 3 Representation equivalent Neural Operators (ReNOs).

The concept of neural operators and operator learning [14, 19] can be encapsulated in a pair  $(U, u)$ . Here,  $U$  is the operator, i.e. the map between infinite-dimensional function spaces, and  $u$  is its

discretization. As we may have access to different discrete representations associated to the input and output functions in the form of frame coefficients (e.g. point samples evaluated on different grids),  $u$  should be able to handle varying inputs and outputs *appropriately*, in a way that is *consistent* with the continuous operator  $U$ . Our next aim is to provide a precise mathematical formulation of this issue. To this end, we need to significantly adapt the concepts of CDE and associated aliasing errors, defined for functions in the previous section, to operators.

### 3.1 Representation equivalent Operators

Let  $U : \text{Dom } U \subseteq \mathcal{H} \rightarrow \mathcal{K}$  be an operator between two separable Hilbert spaces, and let  $\Psi = \{\psi_i\}_{i \in I}$  and  $\Phi = \{\phi_k\}_{k \in K}$  be frame sequences for  $\mathcal{H}$  and  $\mathcal{K}$ , respectively, with synthesis operators  $T_\Psi$  and  $T_\Phi$ . We denote their closed linear spans by  $\mathcal{M}_\Psi := \overline{\text{span}\{\psi_i : i \in I\}}$  and  $\mathcal{M}_\Phi := \overline{\text{span}\{\phi_k : k \in K\}}$ . We note that by classical frame theory [6], the pseudo-inverses  $T_\Psi^\dagger$  and  $T_\Phi^\dagger$ , initially defined on  $\mathcal{M}_\Psi$  and  $\mathcal{M}_\Phi$ , respectively, can in fact be extended to the entire Hilbert spaces, i.e.  $T_\Psi^\dagger : \mathcal{H} \rightarrow \ell^2(I)$  and  $T_\Phi^\dagger : \mathcal{K} \rightarrow \ell^2(K)$ . Given any mapping  $u : \ell^2(I) \rightarrow \ell^2(K)$ , we can build the operator  $T_\Phi \circ u \circ T_\Psi^\dagger : \mathcal{H} \rightarrow \mathcal{K}$ , whose definition clearly depends on the choices of the frame sequences that we make on the continuous level. In other words, any mapping  $u$  can be interpreted as a discrete representation of an underlying continuous operator, which in general, may differ from the operator  $U$ , that is of interest here. Hence, in analogy to Definitions 2.1 and 2.2, we can define the aliasing error of  $U$  relative to the discrete representation  $u$  as,

**Definition 3.1.** The *aliasing error operator*  $\varepsilon(U, u, \Psi, \Phi)$  of  $U$  relative to  $u$  and to the choice of frame sequences  $\Psi$  and  $\Phi$  is the operator  $\varepsilon(U, u, \Psi, \Phi) : \text{Dom } U \subseteq \mathcal{H} \rightarrow \mathcal{K}$  given by

$$\varepsilon(U, u, \Psi, \Phi) = U - T_\Phi \circ u \circ T_\Psi^\dagger, \quad (3.1)$$

and the corresponding *aliasing error* is  $\|\varepsilon(U, u, \Psi, \Phi)\|$ , with  $\|\cdot\|$  denoting the operator norm.

By definition, the aliasing error is zero if and only if  $U = T_\Phi \circ u \circ T_\Psi^\dagger$ , or equivalently if the diagram in Figure 1 commutes, i.e. the black and the blue directed paths in the diagram lead to the same result. If the aliasing error is zero, we say that  $(U, u, \Psi, \Phi)$  satisfies a *continuous-discrete equivalence (CDE)*, implying that accessing the discrete representation  $u$  is exactly the same as accessing the underlying continuous operator  $U$ . See **SM D** for an illustrative example of an aliasing error operator.

Equipped with this notion of continuous-discrete equivalence, we can now introduce the concept of *Representation equivalent Operator (ReO)*. To this end, we denote by  $u$  a map defined on pairs  $(\Psi, \Phi)$  of frame sequences for  $\mathcal{H}$  and  $\mathcal{K}$  and whose values are mappings from  $\ell^2(I)$  into  $\ell^2(K)$ , i.e.  $u(\Psi, \Phi) : \ell^2(I) \rightarrow \ell^2(K)$ .

**Definition 3.2.** We say that  $(U, u)$  is a *Representation equivalent Operator (ReO)* if the aliasing error operator  $\varepsilon(U, u(\Psi, \Phi), \Psi, \Phi)$  is identically zero for every pair  $(\Psi, \Phi)$  of frame sequences that satisfy

$$\text{Dom } U \subseteq \mathcal{M}_\Psi \quad \text{and} \quad \text{Ran } U \subseteq \mathcal{M}_\Phi, \quad (3.2)$$

or equivalently, if the diagram in Figure 1 with the choice  $u = u(\Psi, \Phi)$  commutes.

In other words, we require that the discrete representations  $u(\Psi, \Phi)$  are all equivalent, meaning that they uniquely determine the same underlying operator  $U$ , whenever a continuous-discrete equivalence property holds at the level of the function spaces. If  $(U, u)$  is a ReO, then any frame sequences  $\Psi, \Psi' \subseteq \mathcal{H}, \Phi, \Phi' \subseteq \mathcal{K}$  that satisfy (3.2) result in the following commutative diagram:

$$\begin{array}{ccc} & \ell^2(I) & \xrightarrow{u(\Psi, \Phi)} & \ell^2(K) \\ & \nearrow T_\Psi^\dagger & & \searrow T_\Phi \\ \mathcal{H} & \xrightarrow{U} & & \mathcal{K} \\ & \searrow T_{\Psi'}^\dagger & & \nearrow T_{\Phi'} \\ & \ell^2(I') & \xrightarrow{u(\Psi', \Phi')} & \ell^2(K') \end{array}$$

Thus, there is an equivalence of discrete representations for the same underlying operator  $U$ , justifying the name *Representation equivalent Operators*.

$$\begin{array}{ccc} \mathcal{H} & \xrightarrow{U} & \mathcal{K}, \\ \downarrow T_{\Psi}^{\dagger} & & T_{\Phi}^{\dagger} \uparrow \\ \ell^2(I) & \xrightarrow{u} & \ell^2(K) \end{array}$$

Figure 1

$$\begin{array}{ccc} \mathcal{H} & \xrightarrow{U} & \mathcal{K} \\ T_{\Psi}^{\dagger} \uparrow & & \downarrow T_{\Phi}^{\dagger} \\ \ell^2(I) & \xrightarrow{u} & \ell^2(K) \end{array}$$

Figure 2

$$\begin{array}{ccc} \mathcal{H}_{\ell} & \xrightarrow{\mathcal{G}_{\ell}} & \mathcal{H}_{\ell+1} \\ \downarrow T_{\Psi_{\ell}}^{\dagger} & & T_{\Psi_{\ell+1}}^{\dagger} \uparrow \\ \ell^2(I_{\ell}) & \xrightarrow{\mathfrak{g}_{\ell}(\Psi_{\ell}, \Psi_{\ell+1})} & \ell^2(I_{\ell+1}) \end{array}$$

Figure 3

**Remark 3.3.** A possible way to choose the mapping  $u$  in the above definition is to set,

$$u(\Psi, \Phi) = T_{\Phi}^{\dagger} \circ U \circ T_{\Psi}. \quad (3.3)$$

We observe that this definition of  $u(\Psi, \Phi)$  is such that the diagram in Figure 2 commutes with  $u = u(\Psi, \Phi)$ . Furthermore, the aliasing error operator reads  $\varepsilon(U, u(\Psi, \Phi), \Psi, \Phi) = U - \mathcal{P}_{\mathcal{M}_{\Phi}} \circ U \circ \mathcal{P}_{\mathcal{M}_{\Psi}}$ . This follows by the fact that the compositions  $T_{\Psi} \circ T_{\Psi}^{\dagger}$  and  $T_{\Phi} \circ T_{\Phi}^{\dagger}$  are the orthogonal projections onto the closed subspaces  $\mathcal{M}_{\Psi}$  and  $\mathcal{M}_{\Phi}$ , respectively. Therefore, the pair  $(U, u)$  with  $u$  defined by (3.3) is a ReO. Note that to avoid any aliasing error, the discrete representation of  $U$  has to depend on the chosen frame sequences, i.e. inevitably,  $u$  must depend on  $\Psi$  and  $\Phi$ .

### 3.2 Representation equivalent Neural Operators

The aim in *operator learning* is to learn non-linear operators  $U: \mathcal{H} \rightarrow \mathcal{K}$  between separable Hilbert spaces from input-output function pairs  $\{u_i, U(u_i)\}_{i=1}^N$ . As with neural networks which approximate functions mapping finite-dimensional spaces with compositions of *hidden layers*, we will postulate a similar construction for our operator learning architecture. To this end, we will consider the following composition,

$$\mathcal{G} = \mathcal{G}_L \circ \mathcal{G}_{L-1} \circ \dots \circ \mathcal{G}_1 \quad (3.4a)$$

of linear and non-linear operators

$$\mathcal{G}_{\ell}: \mathcal{H}_{\ell} \rightarrow \mathcal{H}_{\ell+1}, \quad \ell = 1, \dots, L, \quad (3.4b)$$

between separable Hilbert spaces. For each  $1 \leq \ell \leq L$ ,  $\mathcal{G}_{\ell}$  is termed a *hidden layer*.

Note that the form (3.4) of the operator  $\mathcal{G}$  is very general. A particular example is provided by the neural operators (Eqn. 5) of [14], where each hidden layer is of the form  $\mathcal{G}_{\ell} = \sigma(W_{\ell} + K_{\ell} + b_{\ell})$ , with  $W_{\ell}$  a local (pointwise) linear map (matrix),  $K_{\ell}$  a non-local linear operator, for instance a kernel-integral operator and  $b_{\ell}$  a bias function. The non-linearity is introduced through the activation function  $\sigma$  which acts pointwise. However, we prefer to keep the form (3.4) as it can represent much more general operators, for instance when the non-linearity acts in a non-local manner.

In practice, it is not possible to directly work with the continuous form (3.4), whether at the level of accessing inputs/outputs or performing computations on a digital computer. Rather, we have to rely on a discretization of (3.4). Given the notion of *Representation equivalent Operators* above, wherein the continuous form operators can be uniquely and stably recovered from its discretizations, we particularize this concept to neural operators. To this end, we consider a finite sequence of separable Hilbert spaces  $\{\mathcal{H}_{\ell}\}_{\ell=1}^{L+1}$  in (3.4) and for every  $\ell = 1, \dots, L+1$ , we denote by  $\Psi_{\ell} = \{\psi_{\ell,i}\}_{i \in I_{\ell}}$  a *frame sequence* for  $\mathcal{H}_{\ell}$ . The choice of frame sequences for each  $\mathcal{H}_{\ell}$  corresponds to the choice of accessible discrete representations of functions in the underlying function space  $\mathcal{H}_{\ell}$ . For instance, if  $\mathcal{H}_{\ell} = \mathcal{B}_{\Omega}$ , the space of bandlimited functions with bandwidth  $\Omega > 0$ , and we can access the pointwise evaluations of any function  $f \in \mathcal{H}_{\ell}$  on the uniform grid  $\{\frac{n}{2\Omega}\}_{n \in \mathbb{Z}}$ , then  $\Psi_{\ell}$  may be chosen as the orthonormal basis  $\{\text{sinc}(2\Omega x - n)\}_{n \in \mathbb{Z}}$ .

Now let  $(\mathcal{G}, \mathfrak{g})$  be a pair of the form

$$\mathcal{G} = \mathcal{G}_L \circ \mathcal{G}_{L-1} \circ \dots \circ \mathcal{G}_1, \quad \mathfrak{g} = \mathfrak{g}_L \circ \mathfrak{g}_{L-1} \circ \dots \circ \mathfrak{g}_1, \quad (3.5)$$

where, for every  $\ell = 1, \dots, L$ ,  $\mathcal{G}_{\ell}: \mathcal{H}_{\ell} \rightarrow \mathcal{H}_{\ell+1}$  is a (possibly) non-linear operator and  $\mathfrak{g}_{\ell}(\Psi_{\ell}, \Psi_{\ell+1}): \text{Ran } T_{\Psi_{\ell}}^{\dagger} \subseteq \ell^2(I_{\ell}) \rightarrow \text{Ran } T_{\Psi_{\ell+1}}^{\dagger} \subseteq \ell^2(I_{\ell+1})$  for each choice of frame sequences  $\Psi_{\ell} \subseteq \mathcal{H}_{\ell}, \Psi_{\ell+1} \subseteq \mathcal{H}_{\ell+1}$ .

**Definition 3.4.** The pair  $(\mathcal{G}, \mathfrak{g})$  is called a *Representation equivalent Neural Operator (ReNO)* if it is a Representation equivalent Operator at each hidden layer, meaning that for every  $\ell = 1, \dots, L$  the aliasing error operator  $\varepsilon(\mathcal{G}_\ell, \mathfrak{g}_\ell(\Psi_\ell, \Psi_{\ell+1}), \Psi_\ell, \Psi_{\ell+1})$  is identically zero for every pair  $(\Psi_\ell, \Psi_{\ell+1})$  of frame sequences that satisfy  $\text{Dom } \mathcal{G}_\ell \subseteq \mathcal{M}_{\Psi_\ell}$  and  $\text{Ran } \mathcal{G}_\ell \subseteq \mathcal{M}_{\Psi_{\ell+1}}$ . Or equivalently, the diagram in Figure 3 is commutative for every  $\ell$ .

**Remark 3.5.** If the aliasing error operator  $\varepsilon(\mathcal{G}_\ell, \mathfrak{g}_\ell(\Psi_\ell, \Psi_{\ell+1}), \Psi_\ell, \Psi_{\ell+1})$  is identically zero (as required in Definition 3.4), then the assumption that  $\mathfrak{g}_\ell(\Psi_\ell, \Psi_{\ell+1})$  maps  $\text{Ran } T_{\Psi_\ell^\dagger}$  into  $\text{Ran } T_{\Psi_{\ell+1}^\dagger}$  implies that  $\mathfrak{g}_\ell(\Psi_\ell, \Psi_{\ell+1})$  is of the form (3.3), i.e.  $\mathfrak{g}_\ell(\Psi_\ell, \Psi_{\ell+1}) = T_{\Psi_{\ell+1}^\dagger}^\dagger \circ \mathcal{G}_\ell \circ T_{\Psi_\ell}$ . This readily shows that there is only one recipe for constructing ReNOs (see SM B for the proof). In particular, every map  $\mathfrak{g}_\ell$  satisfies

$$\mathfrak{g}(\Psi'_\ell, \Psi'_{\ell+1}) = T_{\Psi'_{\ell+1}^\dagger}^\dagger \circ T_{\Psi_{\ell+1}} \circ \mathfrak{g}(\Psi_\ell, \Psi_{\ell+1}) \circ T_{\Psi_\ell}^\dagger \circ T_{\Psi'_\ell}, \quad (3.6)$$

whenever the frame sequences satisfy (3.2). Thus, Eqn. (3.6) characterizes the discretized versions of ReNO under changes of frame sequences.

Definition 3.4 requires that at each hidden layer the underlying operator satisfies a continuous-discrete equivalence property and accessing a discretized version is exactly the same as accessing the underlying continuous operator whenever the frame sequences fulfill (3.2) at each layer. Requiring this CDE at each hidden layer suffices to enforce it at the level of the operator (3.4) itself:

**Proposition 3.6.** If the pair  $(\mathcal{G}, \mathfrak{g})$  is a ReNO, then the operator  $\mathcal{G}$  in (3.4) is uniquely determined by its discretized version  $\mathfrak{g}$  in (3.5) since the aliasing error (3.1) is identically zero.

As the proof of Proposition 3.6, presented in SM A, also shows, if each hidden layer in the operator (3.4) has an aliasing error (3.1), then these errors may propagate through the network and increase with increasing number of layers.

**Remark 3.7.** The goal in operator learning is to learn an operator  $U: \mathcal{H} \rightarrow \mathcal{K}$  between separable Hilbert spaces  $\mathcal{H}$  and  $\mathcal{K}$  from a discrete representation of its sampled values  $\{u_i, U(u_i)\}_{i=1}^N \in \mathcal{H} \times \mathcal{K}$ . The above discussion indicates it is crucial to maintain an exact correspondence between the continuous and the discrete representations in the training set  $\{u_i, U(u_i)\}_{i=1}^N$  as well. In other words, no aliasing should be present in order to learn the target operator correctly. Mathematically, this can be formulated in terms of accessing the discrete representation  $\{T_\Psi^\dagger(u_i), T_\Phi^\dagger(U(u_i))\}_{i=1}^N$  of  $\{u_i, U(u_i)\}_{i=1}^N$  for underlying frame sequences  $\Psi \subseteq \mathcal{H}$  and  $\Phi \subseteq \mathcal{K}$  that satisfy (3.2).

## 4 Examples

Equipped with our definition of Representation equivalent Neural Operators (ReNOs), we analyze some existing operator learning architectures to ascertain whether they are neural operators or not.

**Convolutional Neural Networks.** Classical convolutional neural networks (CNN) are based on the convolutional layer  $K$ , involving a discrete kernel  $k \in \mathbb{R}^{2s+1}$  and a discrete input  $c$ :

$$K(c)[m] = (k * c)[m] = \sum_{i=-s}^s c[m-i]k[i].$$

We can then analyze this layer using our framework to ask whether this operation can be associated to some underlying continuous operator. Intuitively, if this is the case, the computations conducted on different discretizations effectively representing the input should be consistent; in the contrary case, no associated continuous operator exists and the convolutional operation is not a ReO.

Consider the case where the discrete input  $c$  corresponds to pointwise evaluation on a grid of some underlying bandlimited function  $f \in \mathcal{B}_\Omega$ , for example  $c[n] = f(\frac{n}{2\Omega})$ ,  $n \in \mathbb{Z}$  with associated orthonormal basis  $\Psi = \{\text{sinc}(2\Omega x - n)\}_{n \in \mathbb{Z}}$ . Consider now an alternate representation of  $f$ , point samples of a grid twice as fine:  $d[n] = f(\frac{n}{4\Omega})$ , with  $\Psi' = \{\text{sinc}(4\Omega x - n)\}_{n \in \mathbb{Z}}$  as basis.

Clearly, even though discrete inputs agree, i.e.  $c[n] = d[2n]$ , this is no longer true for the outputs,  $(k * c)[n] \neq (k * d)[2n]$ . This in turn implies that there exist frame sequences  $\Psi, \Psi'$  such that:

$$T_{\Psi} \circ K_{\Phi}(c) \circ T_{\Psi}^{\dagger} \neq T_{\Psi'} \circ K_{\Phi'}(d) \circ T_{\Psi'}^{\dagger}, \quad (4.1)$$

thereby defying the representation equivalence property of ReOs. As a consequence, CNNs are not ReNOs, in the sense of Definition 3.4. This fact is also corroborated with the experimental analysis, Section 5.

**Fourier Neural Operators (FNO).** FNOs are defined in [16] in terms of layers, which are either lifting or projection layers or *Fourier layers*. As lifting and projection layers do not change the underlying spatial structure of the input function, but only act on the channel width, these linear layers will satisfy Definition 3.4 of ReNO here. Hence, we focus on the Fourier layer of the form,

$$v_{\ell+1}(x) = \sigma(A_{\ell}v_{\ell}(x) + B_{\ell}(x) + \mathcal{K}v_{\ell}(x)), \quad (4.2)$$

with the Fourier operator given by

$$\mathcal{K}v = \mathcal{F}^{-1}(R \odot \mathcal{F})(v).$$

Here,  $\mathcal{F}, \mathcal{F}^{-1}$  are the Fourier and Inverse Fourier transforms.

For simplicity of exposition, we set  $A_{\ell} = B_{\ell} \equiv 0$  and focus on investigating whether the Fourier layer (4.2) satisfies the requirements of a ReNO. Following [16], the discrete form of the Fourier layer is given by  $\sigma(kv)$ , with  $kv = F^{-1}(R \odot F(v))$ , where  $F, F^{-1}$  denote the discrete Fourier transform (DFT) and its inverse.

In **SM C**, we show that the convolution in Fourier space operation for the FNO layer (4.2) satisfies the requirements of a ReNO. However the pointwise activation function  $\sigma(f)$ , applied to a bandlimited input  $f \in \mathcal{B}_{\Omega}$  will not necessarily respect the bandlimits, i.e.,  $\sigma(f) \notin \mathcal{B}_{\Omega}$ . In fact, with popular choices of activation functions such as ReLU,  $\sigma(f) \notin \mathcal{B}_{\omega}$ , for any  $\omega > 0$  (see **SM F** for numerical illustrations). Thus, the Fourier layer operator (4.2) may not respect the continuous-discrete equivalence and can lead to aliasing errors, a fact already identified in [9]. Hence, *FNO may not be a ReNO in the sense of Definition 3.4.*

**Spectral Neural Operators (SNO).** Introduced in [9], this architecture is defined as follows. Let  $K > 0$  and denote by

$$\mathcal{P}_K = \left\{ g(x) = \sum_{k=-K}^K c_k e^{i\pi k x} : (c_k)_{k=-K}^K \in \mathbb{C}^{2K+1} \right\},$$

the space of 2-periodic signals bandlimited to  $\pi K$ . Clearly,  $\Psi_K = \{e^{i\pi k \cdot}\}_{k=-K}^K$  constitutes an orthonormal basis for  $\mathcal{P}_K$ , and the corresponding synthesis operator  $T_{\Psi_K}: \mathbb{C}^{2K+1} \rightarrow \mathcal{P}_K$  and analysis operator  $T_{\Psi_K}^*: \mathcal{P}_K \rightarrow \mathbb{C}^{2K+1}$  are given by

$$T_{\Psi_K}((c_k)_{k=-K}^K) = \sum_{k=-K}^K c_k e^{i\pi k \cdot}, \quad T_{\Psi_K}^* f = \{\langle f, e^{i\pi k \cdot} \rangle\}_{k=-K}^K.$$

A spectral neural operator is defined as the compositional mapping  $T_{\Psi_{K'}} \circ \mathcal{N} \circ T_{\Psi_K}^*$ , where  $\mathcal{N}: \mathbb{C}^{2K+1} \rightarrow \mathbb{C}^{2K'+1}$  is an ordinary feedforward neural network with activation function  $\sigma$ ,

$$\mathcal{N}(x) = W^{(L+1)}\sigma(W^{(L)} \dots \sigma(W^{(2)})\sigma(W^{(1)}x - b^{(1)}) - b^{(2)}) \dots - b^{(L)} - b^{(L+1)} \quad (4.3)$$

for some weights  $W^{(\ell)}$  and biases  $b^{(\ell)}$ ,  $\ell = 1, \dots, L+1$ . It is straightforward to see that this architecture corresponds to the commutative diagram,

$$\begin{array}{ccccccc} \mathcal{P}_K & \xrightarrow{T_{\Psi_K}^*} & \mathbb{C}^{2K+1} & \xrightarrow{\mathcal{N}} & \mathbb{C}^{2K'+1} & \xrightarrow{T_{\Psi_{K'}}} & \mathcal{P}_{K'} \\ T_{\Psi_K} \updownarrow T_{\Psi_K}^* & & \updownarrow \text{Id} & & \updownarrow \text{Id} & & T_{\Psi_{K'}} \updownarrow T_{\Psi_{K'}}^* \\ \mathbb{C}^{2K+1} & \xrightarrow{\text{Id}} & \mathbb{C}^{2K+1} & \xrightarrow{\mathcal{N}} & \mathbb{C}^{2K'+1} & \xrightarrow{\text{Id}} & \mathbb{C}^{2K'+1} \end{array}$$



A discretized version of spectral neural operators simply corresponds to an ordinary feedforward neural network mapping Fourier coefficients to Fourier coefficients. We conclude that for SNOs to be ReNOs with respect to the function spaces  $\mathcal{P}_K, \mathcal{P}'_K$ , we have to enforce that for more general choices of frame sequences, Remark 3.5 is satisfied. Moreover, the architecture of SNOs can be generalized with respect to any frames in finite-dimensional inner-product spaces. Under some appropriate condition on the output spanned by DeepONets [19], they also fall in this class.

## 5 Experimental Analysis

An empirical analysis was conducted to demonstrate the pertinence of the framework presented in Section 3, and to support the theoretical claims made in Section 4. This experiment consists in training each of the models presented of Section 4 on a fixed discretization, and evaluating how they behave when alternate discretizations are considered.

Regression to some given target operator  $Q : \mathcal{H} \rightarrow \mathcal{H}$  is considered, where  $\mathcal{H}$  is the space of bandlimited and periodic functions. The precise construction of the target operator is deferred to **SM F**, as well as other experimental details. Frames  $(\Psi, \Phi)$  for  $\mathcal{H}$  are considered, discrete representations of the training data are derived from these, which are used to train a discrete operator  $u : \ell^2(I) \rightarrow \ell^2(K)$ .

After training, we consider alternate discretizations derived from frame sequences  $(\Psi', \Phi')$ , with their associated mapping  $u' : \ell^2(I') \rightarrow \ell^2(K')$ . Discrepancy between the two is then calculated by computing each mapping on their given discretization, and comparing them by expressing  $u'$  in the basis associated to  $u$ . This idea is made formal introducing the discrete aliasing error map  $\epsilon(u, u') : \ell^2(I) \rightarrow \ell^2(K)$ , defined as:

$$\epsilon(u, u') = u - T_{\Phi}^{\dagger} \circ T_{\Phi'} \circ u' \circ T_{\Psi'}^{\dagger} \circ T_{\Psi} \quad (5.1)$$

**Remark 5.1.** In the case of an orthogonal basis, the discrete aliasing error  $\epsilon(u, u')$  precisely corresponds to the discretization of the aliasing error operator, as  $\epsilon(u, u') = T_{\Phi}^{\dagger} \circ \varepsilon(U, u', \Psi', \Phi') \circ T_{\Psi}$  with  $U = T_{\Phi} \circ u \circ T_{\Psi}^{\dagger}$ . This implies that if we consider ReNOs,  $\epsilon(u, u') = 0$  as long as  $(\Psi', \Phi')$  correspond to frame sequences satisfying the spanning condition in (3.2).

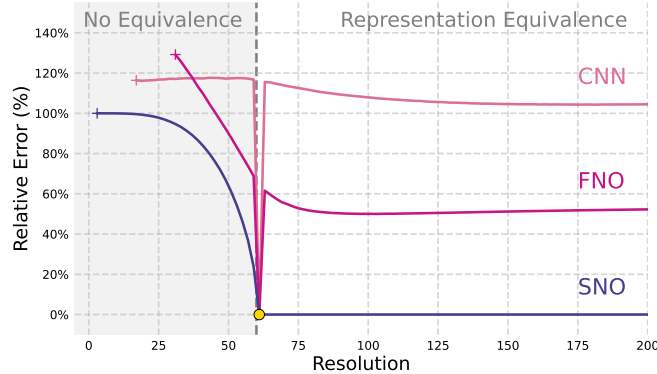


Figure 4: Representation equivalence analysis is conducted by training three classical architectures on a fixed resolution and examining their performance when input resolution is varied. The resolution equivalence zone, located on the right-hand side, denotes the region where discrete representations have an associated frame, while the left-hand side represents the area where this is no longer the case. As predicted by the theory, CNN and FNO are not representation equivalent, while SNO is a ReNO in the representation equivalence region, but failing to do so outside of it.

The results of this experiment are presented in Figure 4. We test 3 models in this experiment, namely CNN, FNO and SNO. All the models are trained on a given resolution ( $61^2$ ) and evaluated on varying resolutions. The results in Figure 4 clearly show that as predicted by our theory, neither CNN nor FNO are representation equivalent, in the sense of Definition 3.4 and changing the resolution, which

amounts to a change of frame, does not keep the operator invariant and causes aliasing errors. On the other hand, as predicted by the theory, SNO is a ReNO as long as the conditions of Definition 3.4 and Remark 3.5 hold. When these conditions no longer hold, SNO also generates aliasing errors. Thus, this experiment clearly demonstrates the practical implications of not respecting representation or continuous-discrete equivalence.

## 6 Discussion

**Summary.** Although a variety of architectures for operator learning are available since the last couple of years, there is still a lack of clarity about what exactly constitutes *operator learning*. Everyone would agree that an operator learning architecture should have functions as inputs and outputs. However, in practice, one does not necessarily have access to functions, either at the input or output level. Rather the access is limited to some form of *discrete representations* of the underlying functions, for instance, point values, cell averages, coefficients with respect to a basis etc. Moreover, one can only perform computations with discrete objects on digital computers. Hence, given the necessity of having to work with discrete representations of functions, it is essential to enforce some relationship between the continuous and discrete representations of the underlying functions or in other words, demand consistency in function space.

Unlike in [14], where the authors advanced a form of *asymptotic consistency* in terms of discretization parameters, we go further to require structure-preserving *continuous-discrete equivalence (CDE)*. To this end, we leverage the well-established concept of CDE for functions in Hilbert spaces in terms of frame theory. The main point about this CDE is the fact that functions, belonging to suitable function spaces, can be uniquely and stably reconstructed, from their frame coefficients. A failure to enforce this CDE results in the so-called aliasing errors that quantitatively measure function space inconsistencies.

We extend these notions of CDE and aliasing errors to operators here. These notions of CDE for operators are then used to define *Representation equivalent Neural Operators (ReNOs)*, see Definition 3.4. Our definition amounts to requiring the CDE for each layer of the approximating operator. This will ensure that there is no aliasing error when the composite operator is considered. Moreover, our framework automatically implies consistency in function spaces and provides a recipe for deriving ReNOs in terms of changing the underlying frames. We also employ our framework to analyze whether or not existing operator learning architectures are ReNOs and also corroborate our results through experiments.

**Related Work.** Our current paper relies heavily on structure preservation from classical numerical analysis [21] and references therein and CDE concepts from signal processing and applied harmonic analysis [24] and references therein. Among the emerging literature on operator learning, [14] was among the first papers to attempt a unifying framework that encompasses many operator learning architectures and codify them through a particular definition of neural operators. We take an analogous route here but with our main point of departure from [14] being that unlike their notion of asymptotic consistency, we require *systematic consistency in function space* by enforcing the representation equivalence for the underlying operator at each layer. Another relevant work is [9] where the authors flag the issue of possible aliasing errors with specific operator learning architectures. We significantly expand on the approach of [9] by providing a rigorous and very general definition for aliasing errors. Finally, our definition of Representation equivalent Neural Operators, leveraging CDEs for operators, is analogous to a similar approach for *computing with functions*, rather than discrete representations, which is the cornerstone of the *Chebfun* project [7] and references therein.

**Limitations and Extensions.** Our aim in this paper was to tackle a fundamental question of what defines a *neural operator*? We have addressed this question here and shown that enforcing some form of CDE is needed for the architecture to genuinely learn the underlying operator rather than just a discrete representation of it. What we have not addressed here are quantitative measures of the error associated with a *Representation equivalent Neural Operator*, introduced in Definition 3.4, in approximating the underlying operator. This is a much broader question than what is addressed here, since sources of error, other than aliasing errors, such as *approximation, training and generalization* errors also contribute to the total error (see [15, 13, 14]).

One interesting direction for further analysis would involve exploring operators and their discretized counterparts that adhere to a relaxed form of CDE. In our formulation, this relaxation could be achieved by allowing the aliasing error to be bounded by a specified value instead of mandating it to be exactly zero.

## References

- [1] J. J. Benedetto. Irregular sampling and frames. *Wavelets: A Tutorial in Theory and Applications*, 2:445–507, 1992.
- [2] M. Bertero, P. Bocacci, and C. De Mol. *Introduction to inverse problems in imaging*. CRC press, 2021.
- [3] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart. Model Reduction And Neural Networks For Parametric PDEs. *The SMAI journal of computational mathematics*, 7:121–157, 2021.
- [4] S. Cao. Choose a transformer: Fourier or galerkin. In *35th conference on neural information processing systems*, 2021.
- [5] T. Chen and H. Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.
- [6] O. Christensen. *Frames and bases: An introductory course*. Springer Science & Business Media, 2008.
- [7] T. A. Driscoll, N. Hale, and L. N. Trefethen. *Chebfun Guide*. Pafnuty Publishers, Oxford, 2014.
- [8] L. C. Evans. *Partial differential equations*, volume 19. American Mathematical Soc., 2010.
- [9] V. Fanaskov and I. Oseledets. Spectral neural operators. *arXiv preprint arXiv:2205.10573*, 2022.
- [10] V. Isakov. *Inverse Problems for Partial Differential Equations*. Springer, 2017.
- [11] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.
- [12] G. Kissas, J. H. Seidman, L. F. Guilhoto, V. M. Preciado, G. J. Pappas, and P. Perdikaris. Learning operators with coupled attention. *Journal of Machine Learning Research*, 23(215):1–63, 2022.
- [13] N. Kovachki, S. Lanthaler, and S. Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22:Art–No, 2021.
- [14] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces. *CoRR*, abs/2108.08481, 2021.
- [15] S. Lanthaler, S. Mishra, and G. E. Karniadakis. Error estimates for DeepONets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022.
- [16] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.
- [17] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar. Neural operator: Graph kernel network for partial differential equations. *CoRR*, abs/2003.03485, 2020.
- [18] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, A. M. Stuart, K. Bhattacharya, and A. Anandkumar. Multipole graph neural operator for parametric partial differential equations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 6755–6766. Curran Associates, Inc., 2020.

- [19] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.
- [20] M. Prasthofer, T. De Ryck, and S. Mishra. Variable input deep operator networks. *arXiv preprint arXiv:2205.11404*, 2022.
- [21] A. Quarteroni and A. Valli. *Numerical approximation of Partial differential equations*, volume 23. Springer, 1994.
- [22] B. Raonić, R. Molinaro, T. Rohner, S. Mishra, and E. de Bezenac. Convolutional neural operators. *arXiv preprint arXiv:2302.01178*, 2023.
- [23] J. H. Seidman, G. Kissas, P. Perdikaris, and G. J. Pappas. NOMAD: Nonlinear manifold decoders for operator learning. *arXiv preprint arXiv:2206.03551*, 2022.
- [24] M. Unser. Sampling-50 years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000.