# **Discretization Error of Fourier Neural Operators**

Margaret Trautner
Caltech

**Samuel Lanthaler** University of Vienna

Andrew M. Stuart
Caltech
astuart@caltech.edu

trautner@caltech.edu

samuel.lanthaler@univie.ac.at

## **Abstract**

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

#### 1 Introduction

While most learning architectures are designed to approximate maps between finite-dimensional spaces, operator learning is a method that approximates maps between infinite-dimensional function spaces. These maps appear commonly in scientific machine learning applications such as surrogate modeling of partial differential equations (PDEs) or model discovery from data. Fourier Neural Operators (FNOs) are a type of operator learning architecture that parameterize the model directly in function space, naturally generalizing deep neural networks (DNNs) [14]. In particular, each hidden layer of an FNO assigns a trainable integral kernel that acts on the hidden states by convolution in addition to the usual affine weights and biases of a DNN. Taking advantage of the duality between convolution and multiplication under Fourier transforms, these convolutional kernels are represented by Fourier multiplier matrices, whose components are optimized during training alongside the regular weights and biases acting in physical space. FNOs have proven to be an effective and popular operator learning method in several PDE application areas including weather forecasting [20], biomedical shape optimization [23], and constitutive modeling [3]. It is thus of interest to study their theoretical properties.

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

$$\Psi^{\dagger} - \Psi_{FNO}^{N} = \underbrace{\left[\Psi^{\dagger} - \Psi_{FNO}\right]}_{\text{model discrepancy}} + \underbrace{\left[\Psi_{FNO} - \Psi_{FNO}^{N}\right]}_{\text{discretization error}}.$$
 (1.1)

Here,  $\Psi^{\dagger}$  is the true map to be approximated by a data-driven model,  $\Psi_{FNO}$  is the continuum FNO map, and  $\Psi^{N}_{FNO}$  is the discretized version of the FNO. In previous analyses of the universal approximation properties of the FNO, the discretization error component is ignored completely; only the continuum definition of the FNO is used [10, 11]. While this approach to universal approximation is mathematically sound, it leaves the discretization components of the error unquantified in practice. Understanding and controlling this discretization error is as important for this model as bounding the model discrepancy error arising from sources such as limited data, optimization, and model capacity. In this paper, we analyze the discretization error both in theory and experimentally.

Aliasing error depends on the regularity, or smoothness, of the input function in the Sobolev sense; this is well known in Fourier analysis. Thus, to bound the error for an entire FNO implementation, regularity must be maintained as the state passes through the layers of the network, including the nonlinear activation function. In particular, regularity-preserving properties of compositions of nonlinear functions are required. Bounds of this type are given by Moser [18] and form a key component of the proofs in this work [18]. Because the smooth GeLU (Gaussian Error Linear Unit) [6] activations preserve regularity, while the non-differentiable ReLU activations do not, the analysis in this paper is confined to the former and extends to other smooth activation functions.

#### 1.1 Contributions

In this paper, we make the following contributions.

- (C1) We bound the discretization error that results from implementing the continuum FNO on a grid.
- (C2) We validate this theory concerning the discretization error of the FNO with numerical experiments.
- (C3) We propose an adaptive subsampling algorithm for faster operator learning training.

In Section 2 we set up the framework for our theoretical results. Section 3 studies the discretization error of the FNO in theory, making contribution (C1). In Section 4 we present numerical experiments that illustrate the theory and propose an algorithm for adaptively refining the discretization during training, making contributions (C2, C3). We conclude in Section 5. The appendices include a self-contained background on aliasing error as well as additional proofs and technical details.

#### 1.2 Related work

Neural networks have been very successful in approximating solutions of partial differential equations using data. Several approaches are used for such models, including physics-informed neural networks (PINNs), constructive networks, and operator learning models. In the case of PINNs, a standard feed-forward machine learning architecture is trained with a loss function involving a constraint of satisfying the underlying PDE [21]. Another approach to applying machine learning to PDEs is to construct approximating networks from classical PDE-solver methods. For example, in [8, 7, 9, 16], ReLU neural networks are shown to replicate polynomial approximations and continuous, piecewise-linear elements used in finite element methods exactly. Both of these two approaches to approximating PDE solution maps require a choice of discretization within the model to approximate an infinite-dimensional operator.

Operator learning is a branch of machine learning that aims to approximate maps between function spaces, which include solution maps defined by partial differential equations (PDEs) [11]. Several operator learning architectures exist, including DeepONet [17], Fourier Neural Operators (FNO) [14], PCA-Net [2], and random features models [19]. Our paper focuses on FNOs, which directly parameterize the model in Fourier space through an integral kernel and allow for changes in discretization in both the input and the output functions, potentially allowing for non-uniform grids [15]. In addition, FNO takes advantage of the computational speedup of the FFT to gain additional model capacity with less evaluation time. A key advantage of the FNO is that it is a *discretization-invariant* operator in the sense that its definition involves no discretization and its implementation can be trivially used on various discretizations with no change of parameter values.

Error analysis for operator learning begins with establishing universal approximation: results which guarantee that, for a class of possible maps, a particular choice of model architecture, and a desired

maximum error, there exists a parameterization of the model that gives at most that error. Universal approximation results are established for a variety of architectures including ReLU neural networks [4], DeepONet [13], FNO [10], and a general class of neural operators [11]. Following universal approximation, model size bounds give a worst-case bound on the model parameter sizes required to achieve a certain error threshold for particular classes of problems. These have been established for FNO [10, 12], but the analysis uses only the continuum definition of the FNO and ignores the fact that in practice the FNO implementation must work with a discretized version. In this work, we close the error gap by quantifying and bounding the error that results from discretizing the continuum FNO.

Perhaps the most conceptually similar work to ours is that of Bartolucci et al. [1], which addresses the fact that discretizations of neural operators deviate from their continuum counterparts. This work introduces an "alias-free" neural operator that bypasses inconsistencies resulting from discretization. In practice, this research direction has led to operator learning frameworks such as Convolutional Neural Operators (CNO), which are not strictly alias free, but reduce aliasing errors via spatial upsampling [22]. These prior works have empirically shown the benefits and importance of carefully controlling discretization errors in operator learning. Prior work has also examined the effects of changing the number of spectral modes in the implemented FNO and an algorithm to optimize training with variable modes [5]. In this work, we also propose an adaptive subsampling algorithm that varies the resolution of the data used in training in a manner designed to minimize training time.

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

#### 2 Notation

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

$$H^{s}(\mathbb{T}^{d}) = \left\{ f : \mathbb{T}^{d} \to \mathbb{R}^{m} \mid \sum_{k \in \mathbb{T}^{d}} (1 + |k|^{2s}) |\widehat{f}(k)|^{2} < \infty \right\}$$
 (2.1)

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

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

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

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

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

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

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

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

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

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

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

$$\lim_{N \to \infty} \frac{1}{N^{d/2}} \|u\|_{\ell^2(n \in [N]^d)} = \|u\|_{L^2(\mathbb{T}^d)}. \tag{2.2}$$

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

**Definition 2.1 (Fourier Neural Operator).** Let A and U be two Banach spaces of real vector-valued functions over domain  $\mathbb{T}^d$ . Assume input functions  $a \in A$  are  $\mathbb{R}^{d_a}$ -valued while the output functions  $u \in \mathcal{U}$  are  $\mathbb{R}^{d_u}$ -valued. The neural operator architecture  $\Psi_{\theta} : A \to \mathcal{U}$  is

$$\Psi_{\theta} = \mathcal{Q} \circ \mathsf{L}_{T-1} \circ \cdots \circ \mathsf{L}_{0} \circ \mathcal{P},$$
  
$$v_{t+1} = \mathsf{L}_{t} v_{t} = \sigma_{t} (W_{t} v_{t} + \mathcal{K}_{t} v_{t} + b_{t}), \quad t = 0, 1, \dots, T - 1,$$

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

$$(\mathcal{K}_t v_t)(x) = \sum_{k \in [[K]]^d} \sum_{j=1}^{d_t} (P_t^{(k)})_j \langle e^{2\pi i \langle k, \cdot \rangle}, (v_t)_j \rangle_{L^2(\mathbb{T}^d; \mathbb{C})} e^{2\pi i \langle k, x \rangle} \in \mathbb{R}^{d_{t+1}}. \tag{2.3}$$

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

In the error analysis in the following section, we are interested in the discrepancy between taking the inner product in equation (2.3) on a grid instead of on a continuum – the errors due to *aliasing*. The above continuum definition is assumed in learning theory analysis of the FNO, but in practice, a discretized approximation is used. We consider the other parameters, including the mode count K, to be fixed and intrinsic to the FNO model considered, irrespective of which grid it is approximated on.

# 3 Main results

Let  $\mathcal{A}$  and  $\mathcal{U}$  be the input and output Banach spaces in the FNO definition 2.1, and let the FNO model hyperparameters be fixed. Given a setting of the trainable FNO parameters  $\theta$ , let  $\Psi_{\text{FNO}}: \mathcal{A} \mapsto \mathcal{U}$  be the FNO obtained using the definition. This definition does not involve a discretization. Thus, any implementation of the FNO with the same hyperparameters and trainable  $\theta$  must be another map, denoted  $\Psi_{\text{FNO}}^N: \mathcal{A} \mapsto \mathcal{U}$ , that evaluates the  $L^2$  inner product in equation (2.3) numerically on grid

points  $X^{(N)}$  rather than at every point  $x \in \mathbb{T}^d$  as  $\Psi_{\text{FNO}}$  does. In particular,  $\Psi^N_{FNO}$  exchanges the operator  $\mathcal{K}_t$  defined in (2.3) for  $\mathcal{K}_t^N$  such that

$$(\mathcal{K}_{t}^{N} v_{t}^{N})(x_{n}) = \sum_{k \in [[K]]^{d}} \sum_{j=1}^{d_{t}} (P_{t}^{(k)})_{j} \mathsf{DFT}((v_{t}^{N})_{j})(k) e^{2\pi i \langle k, x_{n} \rangle} \in \mathbb{R}^{d_{t+1}}, \tag{3.1}$$

where DFT is the discrete Fourier transform. We refer to the output of each internal layer L as a *state value*. Starting from the same input  $a \in \mathcal{A}$ ,  $\Psi_{\text{FNO}}(a)$  and  $\Psi^N_{\text{FNO}}(a)$  will have different state values, denoted  $v_t$  and  $v_t^N$ , respectively, as outputs of internal layer  $\mathsf{L}_{t-1}$  for t>0 despite having the exact same model parameters. This difference is important because in proofs concerning the FNO, only  $\Psi_{\text{FNO}}$  is considered, but  $\Psi_{\text{FNO}}$  is an *unimplementable* object in practice. If  $\Psi^\dagger$  is the map of interest to be approximated using an FNO model, the overall approximation error of the implemented  $\Psi^N_{\text{FNO}}$  can be split into a contribution due to the numerical discretization and another contribution due to model discrepancy, as shown in (1.1). Theorem 3.2 bounds the discretization error component. The result takes into account both the initial errors that occur with each approximated inner product and their magnified effects as they propagate through the layers of the model. Despite this nonlinear propagation, we show that the approximate  $L^2$  norm of the error after any number of layers decreases like  $N^{-s}$ , where s describes the Sobolev regularity of the input.

To prove Theorem 3.2, we assume the following.

**Assumptions 3.1.** For a fixed FNO with T layers:

- (A1) There exists some  $B \ge 1$  such that  $\sigma_t, \sigma_p$ , and  $\sigma_q$  possess continuous derivatives up to order s which are bounded by B.
- (A2) Input set  $A \subset H^s(\mathbb{T}^d)$ .
- (A3)  $1 \le K < \frac{N}{2}$ .
- (A4)  $s > \frac{d}{2}$ .
- (A5) There exists some  $M \geq 1$  such that FNO parameters  $P_t$ ,  $W_t$ , and  $b_t$  are each bounded above by M in the following norms:  $\|P_t\|_F \leq M$ ,  $\|W_t\|_2 \leq M$ , and  $|b_t| \leq M$  for all  $t \in [0, \ldots, T-1]$ . Furthermore,  $\mathcal{P}$  and  $\mathcal{Q}$  are bounded and smooth with  $\|\mathcal{P}\|_{NN} \leq M$ , and  $\|\mathcal{Q}\|_{NN} \leq M$ .

The above assumptions are easily satisfied in practice. We investigate the consequences of violating Assumption (A1) in the numerical experiments. The main result is the following theorem concerning the behavior of the error with respect to the size of the discretization.

**Theorem 3.2.** Let Assumptions 3.1 hold. Let  $A_c$  be a compact set in A. Let  $v_t(a) := L_{t-1} \circ \cdots \circ L_0 \circ \mathcal{P}(a)$  with  $\mathcal{P}$  and each L as defined in Definition 2.1. Similarly, let  $v_t^N(a) := L_{t-1}^N \circ \cdots \circ L_0^N \circ \mathcal{P}(a)$  where  $L_j^N v_j^N = \sigma_j(W_j v_j^N + \mathcal{K}_j^N v_j^N + b_j)$  for  $\mathcal{K}_j^N$  defined in (3.1) for each  $0 \le j \le t-1$ . Then

$$\sup_{a \in \mathcal{A}_c} \frac{1}{N^{d/2}} \| v_t(a) - v_t^N(a) \|_{\ell^2(n \in [N]^d)} \le C N^{-s}$$
(3.2)

where the constant C depends on B, M, d, s, t, and  $A_c$ .

The proof and exact form of the constant C in the above theorem are detailed in the supplementary materials.

We can also state the following variant of Theorem 3.2, which shows that the same convergence rate is obtained at the continuous level, when  $v_t^N(x_n)$  is replaced by a trigonometric polynomial interpolant:

**Theorem 3.3.** Let  $p_t^N(a)(x) = \sum_{k \in [[N]]^d} \mathsf{DFT}(v_t^N(a))(k) e^{2\pi i \langle k, x \rangle}$  denote the interpolating trigonometric polynomial of  $\{v_t^N(a)(x_n)\}_{n \in [N]^d}$ . Let the assumptions of Theorem 3.2 hold. Then,

$$\sup_{a \in \mathcal{A}_c} \|v_t(a) - p_t^N(a)\|_{L^2(\mathbb{T}^d)} \le C' N^{-s}. \tag{3.3}$$

Here, C' depends on B, M, d, s, t, and A.

**Remark 3.4.** A consequence of Theorem 3.3 is that  $\sup_{a \in \mathcal{A}_c} \|\Psi_{FNO}(a) - \Psi^N_{FNO}(a)\|_{L^2(\mathbb{T}^d)}$  also has a convergence rate of  $N^{-s}$  since under Assumptions 3.1  $\mathcal{Q}$  is Lipschitz and  $\mathcal{Q}, \mathcal{P}$  preserve regularity.

The exact form of the constant C' may be found in the proof in the appendix. A key element in the proof of Theorem 3.2 is to provide a bound on the Sobolev norm of the ground truth state  $||v_t||_{H^s}$  at each layer. The following lemma accomplishes this for a single layer. The proof may be found in the appendix.

**Lemma 3.5.** *Under Assumptions 3.1, the following bounds hold:* 

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

for some constant c dependent on d and s, where  $\sigma^* := \max\{\max_{0 \le t \le T-1} \sigma_t(0), 1\}$ .

The result of Theorem 3.2 guarantees that the discretization error converges as grid resolution increases. The algebraic decay rate in a discrete  $L^2$  norm is determined by the regularity of the input; this in turn builds on Lemma 3.5 which ensures that the regularity of the state is preserved through each layer of the FNO.

# 4 Numerical experiments

In this section we present and discuss results from numerical experiments that empirically validate the results of Theorem 3.2. The  $L^2$  error at each layer decreases like  $N^{-s}$  where s governs the input regularity and N is the discretization used to perform convolutions in the FNO implementation. For each FNO model in this section, we use a computation of a discrete FNO on a high resolution grid as the "ground truth;" this is standard practice in numerical analysis when the true solution is unobtainable. We compare states at each layer resulting from inputs of lower resolution with the state resulting from the ground truth. To obtain evaluations of  $v_{\ell}$  at higher discretizations than N, the inverse Fourier transform operation is interpolated to additional grid points using trigonometric polynomial interpolation; Theorem 3.3 justifies this practice.

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

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

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

In Subsection 4.1 we discuss experiments on FNOs with random weights, and in Subsection 4.2 we discuss experiments on trained FNOs. In the random weights experiments, we present a few interesting experimental findings; namely, using ReLU activations or non-periodic position encodings negatively affects the discretization error decay as the theory predicts. In the trained network experiments, we explore the example of learning a gradient map to show that the model cannot learn an output with less regularity than the input. Finally, in Subsection 4.3, we propose an application of discretization subsampling to speed up operator learning training by leveraging adaptive grid sizes.

#### 4.1 Experiments with random weights

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

The default model has randomly initialized iid  $\mathcal{U}(-\frac{1}{\sqrt{d_t}},\frac{1}{\sqrt{d_t}})$  weights (uniformly distributed) for the affine and bias terms, where  $d_t$  is the layer width, and iid  $\mathcal{U}(0,\frac{1}{d_t^2})$  spectral weights. Initializing

the weights this way is the standard default for FNO. This model uses the GeLU activation function standard in FNO. Next we examine the use of ReLU activation instead of GeLU. Finally, we investigate non-periodic positional encoding.

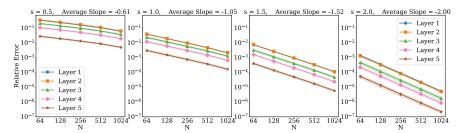


Figure 1: Relative error versus N and s for an FNO with default weight initialization.

**Discretization error for random weights models** The relative error of the state at each layer versus the discretization for inputs of varying regularity may be seen for the default model, the ReLU model, and the non-periodic position encoding model in Figures 1,2, and 3 respectively. In these figures, from left to right,  $s \in \{0.5, 1, 1.5, 2\}$  where  $v_0 \in H^{s-}$ . The uncertainty shading indicates two standard deviations from the mean over five inputs to the FNO.

As can be seen in Figure 1 for the model with the default weight initialization, the empirical behavior of the error matches the behavior expected from Theorem 3.2. One question that arises from Figure 1 is why the error decreases as the number of layers increases; this is an effect of the magnitude of the weights. When the model weights are multiplied by 10, then the error begins to increase with the number of layers. A figure illustrating this phenomenon may be found in the supplementary materials, where additional weight initializations are explored as well.

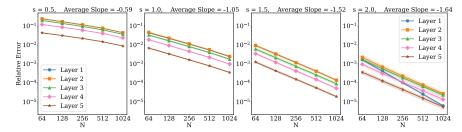


Figure 2: Relative error versus N and s for a default FNO with a ReLU activation.

The results shown in Figure 2 justify the use of the GeLU activation function, which belongs to  $C^{\infty}$ , over the ReLU activation function, which is only Lipschitz. The figure shows that the benefit of having sufficiently smooth inputs is negated by the ReLU activation: the error decay is limited. Note that this effect does not occur for the first layer since at that point ReLU has been applied once, and the Fourier transform is not applied to the output of an activation function until the second layer. Additionally, we do not observe the effect of ReLU until the input has regularity greater than s=1.5 since the ReLU activation function has regularity of s=1.5

A similar effect to the ReLU model occurs when non-periodic positional encoding information is appended to the input, as is standard in practical FNO usage; see Figure 3. Since this grid data has a jump discontinuity across the boundary of  $[0,1]^d$ , it has regularity of s=0.5, so the convergence rate never achieves  $N^{-1}$ . These results suggest caution when using positional encoding information with smooth input data; periodic positional encodings may be preferred.

#### 4.2 Experiments with trained networks

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

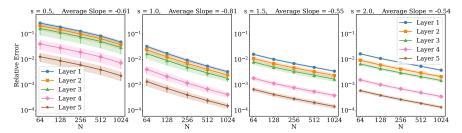


Figure 3: Relative error versus N and s for a default FNO with non-periodic position encoding appended to the input.

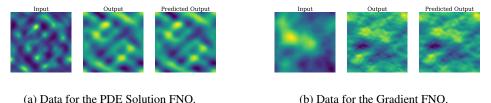


Figure 4: Visualization of the input and output data for the trained model examples.

map of the following PDE:

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

$$\nabla \cdot (\nabla \chi A) = \nabla \cdot A, \quad y \in \mathbb{T}^2 \tag{4.1}$$

$$\chi \text{ is } 1 - \text{periodic}, \ \int_{\mathbb{T}^2} \chi \ \mathrm{d}y = 0.$$
 (4.2)

(b) Data for the Gradient FNO.

Here, the input  $A: \mathbb{T}^2 \mapsto \mathbb{R}^{2 \times 2}$  is symmetric positive definite at every point in the domain  $\mathbb{T}^2$  and is bounded and coercive. For the output data we take the first component of  $\chi: \mathbb{T}^2 \mapsto \mathbb{R}^2$ . In our experiments the model is trained to < 5% relative  $L^2$  test error. A visualization of the data is in Figure 4a.

The error versus discretization analysis can be seen in Figure 5. The error decreases slightly faster than predicted by the theory; a potential explanation is that the trained model itself has a smoothing effect that is not exploited in our analysis.

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

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

The error versus discretization for inputs of various smoothness is shown in Figure 6. The error decreases according the the smoothness of the input despite the smoothness-decreasing properties of the data. Indeed, the model does produce more regular predicted outputs than the true gradient, as can be seen in Figure 4b where the predicted output is visibly smoother than the true output.

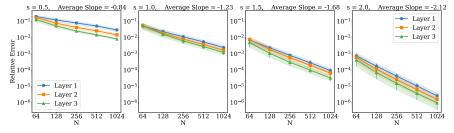


Figure 5: Error versus discretization for inputs of varying regularity for the FNO trained on data corresponding to a PDE solution.

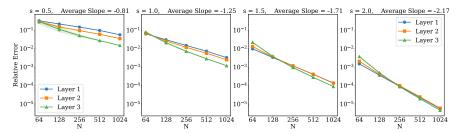


Figure 6: Error versus discretization for inputs of varying regularity for the FNO trained on data corresponding to a gradient map.

#### 4.3 Speeding up training via adaptive subsampling

The fact that the FNO architecture and its parametrization are independent of the numerical discretization allows for increased flexibility. Specifically, it is possible to adaptively choose an optimal discretization for a given objective. Furthermore, the error decomposition (1.1) invites an exploitation to optimize computational training time.

The idea of the proposed approach is that, during training, it is not necessary to compute outputs to a numerical accuracy that is substantially better than the model discrepancy. This suggests an adaptive choice of the numerical discretization, where we use a coarse grid during the early phase of training and refine the grid later. In practice, we realize this idea by introducing a

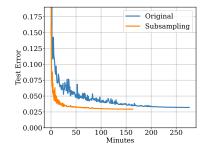


Figure 7: Adaptive grid refinement leads to greater training efficiency.

subsampling scheduler. The subsampling scheduler tracks a validation error on held out data and adaptively changes the numerical resolution via suitable subsampling of the training data. Starting from a coarse resolution, we iteratively double the grid size once the validation error plateaus.

We train FNO for the elliptic PDE (4.1) with and without the subsampling scheduler; details are contained in the supplementary materials. Compared to training without subsampling, training with a subsampling scheduler requires the same number of forward and backward passes over the network for the training and test set, plus an additional overhead due to the validation set. Since we are mainly interested in the training time, our choice of adding validation samples, rather than performing a training/validation split of the original training samples, ensures that computational timings are not skewed in favor of subsampling. Over the course of training, we iterate through the following grid sizes: 32x32, 64x64, and 128x128. Our criterion for a plateau is that the validation error has not improved for 40 training epochs. The results of training with and without subsampling scheduler for the PDE solution model (4.1) are shown in Figure 7. We observe that training time can be substantially reduced with subsampling. The presentation of the algorithm here is preliminary and serves to emphasize the importance of considering discretization error in the model. The success of this simple example points to the potential benefits of developing adaptive numerical methods for model evaluation within operator learning.

## 5 Conclusions

In this paper, we analyze the discretization error that results from implementation of Fourier Neural Operators (FNOs). We bound the  $L^2$  norm of the error in Theorem 3.2, proving an upper bound that decreases asymptotically as  $N^{-s}$ , where N is the discretization in each dimension, and s is the input regularity. We show empirically that FNOs with random weights chosen as the default FNO weights for training behave almost exactly as the theory predicts. Furthermore, our theory and experiments justify the use of the GeLU activation function in FNO over ReLU, as the former preserves regularity. Additional analyses on trained models show that the error behaves less predictably in relation to our theory in the low-discretization regime. Finally, we use the decomposition of model error and discretization error to propose an adaptive subsampling algorithm for decreasing training time with operator learning. As FNOs become a more common tool in scientific machine learning, understanding the various sources of error is critical. By bounding FNO discretization error and demonstrating its behavior in numerical experiments, we understand its effect on learning and the potential to minimize computational costs by an adaptive choice of numerical resolution.

# **Acknowledgments and Disclosure of Funding**

SL is supported by Postdoc.Mobility grant P500PT-206737 from the Swiss National Science Foundation. The work of AMS is supported by a Department of Defense Vannevar Bush Faculty Fellowship, and by the SciAI Center, funded by the Office of Naval Research (ONR), under Grant Number N00014-23-1-2729. MT is supported by the Department of Energy Computational Science Graduate Fellowship under award number DE-SC00211. All code and data for this work is available at https://github.com/mtrautner/BoundFN0. The authors are grateful to Nicholas Nelsen for helpful discussions on FNO implementation. The computations presented here were conducted in the Resnick High Performance Computing Center, a facility supported by Resnick Sustainability Institute at the California Institute of Technology.

#### References

- [1] F. Bartolucci, E. de Bézenac, B. Raonić, R. Molinaro, S. Mishra, and R. Alaifari. Are neural operators really neural operators? frame theory meets operator learning. *arXiv preprint arXiv:2305.19913*, 2023.
- [2] 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.
- [3] K. Bhattacharya, N. Kovachki, A. Rajan, A. M. Stuart, and M. Trautner. Learning homogenization for elliptic operators. *SIAM Journal on Numerical Analysis*, 2024.
- [4] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- [5] R. J. George, J. Zhao, J. Kossaifi, Z. Li, and A. Anandkumar. Incremental spatial and spectral learning of neural operators for solving large-scale pdes. *arXiv e-prints*, pages arXiv–2211, 2022.
- [6] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016.
- [7] L. Herrmann, C. Schwab, and J. Zech. Deep neural network expression of posterior expectations in Bayesian PDE inversion. *Inverse Problems*, 36(12):125011, 32, 2020. ISSN 0266-5611,1361-6420. doi: 10.1088/1361-6420/abaf64. URL https://doi.org/10.1088/1361-6420/abaf64.
- [8] L. Herrmann, J. A. Opschoor, and C. Schwab. Constructive deep relu neural network approximation. *Journal of Scientific Computing*, 90(2):75, 2022.
- [9] L. Herrmann, C. Schwab, and J. Zech. Neural and spectral operator surrogates: unified construction and expression rate bounds, 2024.
- [10] N. Kovachki, S. Lanthaler, and S. Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021.
- [11] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [12] N. B. Kovachki, S. Lanthaler, and A. M. Stuart. Operator learning: Algorithms and analysis. *arXiv preprint arXiv:2402.15715*, 2024.
- [13] 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.
- [14] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations*, 2021.

- [15] Z. Li, D. Z. Huang, B. Liu, and A. Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.
- [16] M. Longo, J. A. Opschoor, N. Disch, C. Schwab, and J. Zech. De rham compatible deep neural network fem. *Neural Networks*, 165:721–739, 2023.
- [17] 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.
- [18] J. Moser. A rapidly convergent iteration method and non-linear partial differential equations-i. Annali della Scuola Normale Superiore di Pisa-Scienze Fisiche e Matematiche, 20(2):265–315, 1966.
- [19] N. H. Nelsen and A. M. Stuart. The random feature model for input-output maps between banach spaces. *SIAM Journal on Scientific Computing*, 43(5):A3212–A3243, 2021.
- [20] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.
- [21] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [22] B. Raonic, R. Molinaro, T. De Ryck, T. Rohner, F. Bartolucci, R. Alaifari, S. Mishra, and E. de Bézenac. Convolutional neural operators for robust and accurate learning of pdes. *Advances in Neural Information Processing Systems*, 36, 2024.
- [23] T. Zhou, X. Wan, D. Z. Huang, Z. Li, Z. Peng, A. Anandkumar, J. F. Brady, P. W. Sternberg, and C. Daraio. Ai-aided geometric design of anti-infection catheters. *Science Advances*, 10(1): eadj1741, 2024.

### **Appendices**

# A Trigonometric interpolation and aliasing

In this section, we present a self-contained analysis of aliasing errors for  $v \in H^s(\mathbb{T}^d)$ . These results are straightforward and well known in numerical analysis, but we give a clear exposition here as background as it is difficult to find a succinct and widely-available reference. The primary goal is to state and prove Proposition A.6, which controls the difference between a function defined over  $\mathbb{T}^d$  and the trigonometric interpolation of a function defined on a grid. In the following,  $N \in \mathbb{Z}_{>0}$ . We recall that  $X^{(N)}$  is a set of equidistant grid points on the torus  $\mathbb{T}^d$ ,

$$X^{(N)} = \{x_n \in \mathbb{T}^d \mid x = n/N, \ n \in [N]^d\}.$$

We note that the discrete Fourier transform gives rise to a natural correspondence between grid values and Fourier modes,

$$\{v(x_n)\}_{n\in[N]^d} \leftrightarrow \{\widehat{v}_k\}_{k\in[[N]]^d},\tag{A.1}$$

where

$$\widehat{v}_k = \frac{1}{N^d} \sum_{n \in [N]^d} v(x_n) e^{-2\pi i \langle k, x_n \rangle} =: \mathsf{DFT}(v)(k). \tag{A.2}$$

We begin with the following observation:

**Lemma A.1.** Let N be given. Then,

$$\frac{1}{N^d} \sum_{k \in [[N]]^d} e^{2\pi i \langle k, x_m - x_n \rangle} = \delta_{mn}, \quad \forall m, n \in [N]^d, \tag{A.3}$$

$$\frac{1}{N^d} \sum_{n \in [N]^d} e^{2\pi i \langle k - k', x_n \rangle} = \delta_{kk'}, \quad \forall k, k' \in [[N]]^d. \tag{A.4}$$

*Proof.* This follows from an elementary calculation, which we briefly recall here. For d=1, the claim follows by noting that  $x_n=n/N$ , and using the identity

$$\sum_{\ell=0}^{N-1} q^{\ell} = \begin{cases} \frac{q^N - 1}{q - 1}, & (q \neq 1), \\ N, & (q = 1), \end{cases}$$
 (A.5)

with  $q=e^{2\pi i(m-n)/N}$  and  $q=e^{2\pi i(k-k')/N}$ , respectively. Indeed, assuming d=1 and denoting  $-K:=\min[[N]]$ , then the above identity implies, for example,

$$\sum_{k \in [[N]]} e^{2\pi i k(x_m - x_n)} = \sum_{k \in [[N]]} \left[ \underbrace{e^{2\pi i (m - n)/N}}_{=:q} \right]^k = \sum_{k \in [[N]]} q^k = q^{-K} \sum_{\ell=0}^{N-1} q^{\ell}.$$

If  $q \neq 1$ , then  $q^N = e^{2\pi i(m-n)} = 1$ . By (A.5), this implies that the last sum is 0. On the other hand, if q = 1, then the last sum is trivially = N. We finally note that, for  $m, n \in [N]$ , we have q = 1 if and only if m = n, implying that

$$q^{-K} \sum_{\ell=0}^{N} q^{\ell} = N \delta_{mn}.$$

Thus,

$$\sum_{k \in [[N]]} e^{2\pi i k(x_m - x_n)} = N \delta_{mn},$$

and (A.3) follows. The argument for (A.4) is analogous. For d>1, the sum over  $[[N]]^d=[[N]]\times\cdots\times[[N]]$  is split into sums along each dimension, and the same argument is applied for each of the d components, yielding the claim also for d>1.

A trigonometric polynomial  $p: \mathbb{T}^d \mapsto \mathbb{R}^m$  is a function of the form

$$p(x) = \sum_{k \in [[N]]^d} c_k e^{2\pi i \langle k, x \rangle}$$
(A.6)

with  $c_k \in \mathbb{C}^m$  chosen to make p(x)  $\mathbb{R}^m$ -valued at each  $x \in \mathbb{T}^d$ . We note that the discrete and continuous  $L^2$ -norms are equivalent for trigonometric polynomials:

**Lemma A.2.** Let N be a positive integer. If p(x) is a trigonometric polynomial, then

$$\frac{1}{N^{d/2}} \|p\|_{\ell^2(n \in [N]^d)} = \|p\|_{L^2(\mathbb{T}^d)}.$$

Proof. We have

$$||p||_{L^{2}(\mathbb{T}^{d})}^{2} = \int_{\mathbb{T}^{d}} |p(x)|^{2} dx = \sum_{k,k' \in [[N]]^{d}} c_{k} \overline{c}_{k'} \underbrace{\int_{\mathbb{T}^{d}} e^{2\pi i \langle k-k',x \rangle} dx}_{=\delta_{kk'}} = \sum_{k \in [[N]]^{d}} |c_{k}|^{2},$$

and

$$\frac{1}{N^d} ||p||_{\ell^2(n \in [N]^d)}^2 = \frac{1}{N^d} \sum_{n \in [N]^d} |p(x_n)|^2$$

$$= \sum_{k,k' \in [[N]]^d} c_k \overline{c}_{k'} \underbrace{\frac{1}{N^d} \sum_{n \in [N]^d} e^{2\pi i \langle k - k', x_n \rangle}}_{=\delta_{kk'}}$$

$$= \sum_{k \in [[N]]^d} |c_k|^2.$$

This proves the claim.

Let  $v: \mathbb{T}^d \to \mathbb{R}$  be a function with grid values  $\{v(x_n)\}_{n \in [N]^d}$ . Let  $\mathsf{DFT}(v)(k)$  denote the coefficients of the discrete Fourier transform defined by (A.1). Then

$$p(x) := \sum_{k \in [[N]]^d} \mathsf{DFT}(v)(k) e^{2\pi i \langle k, x \rangle},\tag{A.7}$$

П

is the trigonometric polynomial associated to v. The next lemma shows that p(x) interpolates v(x). **Lemma A.3.** The trigonometric polynomial p(x) defined by (A.7) interpolates v(x) at the grid points, i.e., we have  $p(x_n) = v(x_n)$  for all  $n \in [N]^d$ .

*Proof.* Fix  $n \in [N]^d$ . Then

$$\begin{split} p(x_n) &= \sum_{k \in [[N]]^d} \mathsf{DFT}(v)(k) e^{2\pi i \langle k, x_n \rangle} \\ &= \sum_{k \in [[N]]^d} \left\{ \frac{1}{N^d} \sum_{m \in [N]^d} v(x_m) e^{-2\pi i \langle k, x_m \rangle} \right\} e^{2\pi i \langle k, x_n \rangle} \\ &= \sum_{m \in [N]^d} v(x_m) \left\{ \frac{1}{N^d} \sum_{k \in [[N]]^d} e^{2\pi i \langle k, x_n - x_m \rangle} \right\} \\ &= \sum_{m \in [N]^d} v(x_m) \delta_{mn} \\ &= v(x_m) \end{split}$$

where we have made use of (A.3) to pass to the fourth line.

The following trigonometric polynomial interpolation estimate for functions in Sobolev spaces  $H^s(\mathbb{T}^d)$  will be useful in stating our main proposition:

**Lemma A.4.** Let  $v \in H^s(\mathbb{T}^d)$  for s > d/2. Let p denote the interpolating trigonometric polynomial given by (A.7). Then

$$v(x) - p(x) = \sum_{k \in \mathbb{Z}^d \setminus [[N]]^d} \widehat{v}(k) e^{2\pi i \langle k, x \rangle} - \sum_{k \in [[N]]^d} \left\{ \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} \widehat{v}(k+\ell N) \right\} e^{2\pi i \langle k, x \rangle}. \tag{A.8}$$

Furthermore, there exists a constant  $c_{s,d} > 0$ , such that

$$||v - p||_{L^{2}(\mathbb{T}^{d})} \le c_{s,d} ||v||_{H^{s}(\mathbb{T}^{d})} N^{-s}.$$
(A.9)

**Remark A.5.** The first sum on the right-hand side of (A.8) is the  $L^2$ -orthogonal Fourier projection of v onto the complement of  $\mathrm{span}\{e^{2\pi i \langle k,x\rangle} \mid k \in [[N]]^d\}$ . The second sum in (A.8) is known as an "aliasing" error; It arises because two Fourier modes are indistinguishable on the discrete grid whenever  $k-k' \in \mathbb{NZ}^d$ , i.e.  $e^{2\pi i \langle k,x_n\rangle} = e^{2\pi i \langle k',x_n\rangle}$  for all  $n \in [N]^d$ .

*Proof.* Since  $v \in H^s(\mathbb{T}^d)$  has Sobolev smoothness s for s > d/2, it can be shown that the Fourier series of v is uniformly convergent. In particular, we may write

$$v(x) = \sum_{k' \in \mathbb{Z}^d} \widehat{v}(k') e^{2\pi i \langle k', x \rangle}$$

for

$$\widehat{v}(k') = \int_{\mathbb{T}^d} v(x) e^{-2\pi i \langle k', x \rangle} \, \mathrm{d}x.$$

First, substitution of  $v(x_n) = \sum_{k' \in \mathbb{Z}^d} \widehat{v}(k') e^{2\pi i \langle k', x_n \rangle}$  into DFT(v)(k) yields

$$\mathsf{DFT}(v)(k) = \frac{1}{N^d} \sum_{n \in [N]^d} \left\{ \sum_{k' \in \mathbb{Z}^d} \widehat{v}(k') e^{2\pi i \langle k', x_n \rangle} \right\} e^{-2\pi i \langle k, x_n \rangle}$$
$$= \sum_{k' \in \mathbb{Z}^d} \widehat{v}(k') \left\{ \frac{1}{N^d} \sum_{n \in [N]^d} e^{2\pi i \langle k' - k, x_n \rangle} \right\}$$

We now note that

$$\frac{1}{N^d} \sum_{n \in [N]^d} e^{2\pi i \langle k' - k, x_n \rangle} = \begin{cases} 0, & (k' \not\equiv k \mod N), \\ 1, & (k' \equiv k \mod N), \end{cases}$$

as a consequence of the trigonometric identity (A.4). Letting  $k' = k + \ell N$ , i.e. k' for which the sum inside the braces does not vanish, it follows that

$$\mathsf{DFT}(v)(k) = \sum_{\ell \in \mathbb{Z}^d} \widehat{v}(k + \ell N).$$

Thus,

$$\begin{split} v(x) - p(x) &= \sum_{k \in \mathbb{Z}^d} \widehat{v}(k) e^{2\pi i \langle k, x \rangle} - \sum_{k \in [[N]]^d} \mathsf{DFT}(v)(k) e^{2\pi i \langle k, x \rangle} \\ &= \sum_{k \in \mathbb{Z}^d \backslash [[N]]^d} \widehat{v}(k) e^{2\pi i \langle k, x \rangle} + \sum_{k \in [[N]]^d} \left\{ \widehat{v}(k) - \mathsf{DFT}(v)(k) \right\} e^{2\pi i \langle k, x \rangle} \\ &= \sum_{k \in \mathbb{Z}^d \backslash [[N]]^d} \widehat{v}(k) e^{2\pi i \langle k, x \rangle} - \sum_{k \in [[N]]^d} \left\{ \sum_{\ell \in \mathbb{Z}^d \backslash \{0\}} \widehat{v}(k + \ell N) \right\} e^{2\pi i \langle k, x \rangle}. \end{split}$$

We proceed to bound the last two terms. For the first term, we have by Parseval's theorem,

$$\begin{split} \left\| \sum_{k \in \mathbb{Z}^d \setminus [[N]]^d} \widehat{v}(k) e^{2\pi i \langle k, x \rangle} \right\|_{L^2(\mathbb{T}^d)}^2 &= \sum_{k \in \mathbb{Z}^d \setminus [[N]]^d} |\widehat{v}(k)|^2 \\ &\leq \frac{1}{(1 + (N/2)^{2s})} \sum_{k \in \mathbb{Z}^d} (1 + |k|^{2s}) |\widehat{v}(k)|^2 \\ &\leq 4^s N^{-2s} \|v\|_{H^s(\mathbb{T}^d)}^2, \end{split}$$

where  $\|v\|_{H^s(\mathbb{T}^d)}^2 = \sum_{k \in \mathbb{Z}^d} (1+|k|^{2s})|\widehat{v}(k)|^2$ , and for the second term

$$\begin{split} \Big\| \sum_{k \in [[N]]^d} \Big\{ \sum_{\ell \in \mathbb{Z}^d \backslash \{0\}} \widehat{v}(k+\ell N) \Big\} e^{2\pi i \langle k, x \rangle} \Big\|_{L^2(\mathbb{T}^d)}^2 \\ &= \sum_{k \in [[N]]^d} \Big| \sum_{\ell \in \mathbb{Z}^d \backslash \{0\}} \widehat{v}(k+\ell N) \Big|^2 \\ &\leq \sum_{k \in [[N]]^d} \Big( \sum_{\ell \in \mathbb{Z}^d \backslash \{0\}} (1+|k+\ell N|^{2s})^{-1} \Big) \\ &\qquad \times \Big( \sum_{\ell \in \mathbb{Z}^d \backslash \{0\}} (1+|k+\ell N|^{2s}) |\widehat{v}(k+\ell N)|^2 \Big). \end{split}$$

The final inequality is obtained via Cauchy-Schwarz. We note that for  $k \in [[N]]^d$ , we have  $|k|_{\infty} \le N/2$ , and hence, for any integer vector  $\ell \ne 0$ , we obtain

$$|k + \ell N| \ge |k + \ell N|_{\infty} \ge |\ell|_{\infty} N - |k|_{\infty} \ge |\ell|_{\infty} N - \frac{N}{2} \ge \frac{N}{2} |\ell|_{\infty} \ge \frac{N}{2\sqrt{d}} |\ell|.$$
 (A.10)

We can now bound

$$\sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} (1 + |k + \ell N|^{2s})^{-1} \le \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} \left(\frac{N}{2\sqrt{d}}\right)^{-2s} |\ell|^{-2s} \tag{A.11a}$$

 $\leq c_{d,s} N^{-2s},\tag{A.11b}$ 

where  $c_{d,s}:=(4d)^s\sum_{\ell\in\mathbb{Z}^d\setminus\{0\}}|\ell|^{-2s}<\infty$  is finite, since s>d/2 implies that the last series converges. Substitution of this bound in the estimate above implies,

$$\begin{split} \left\| \sum_{k \in [[N]]^d} \left\{ \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} \widehat{v}(k+\ell N) \right\} e^{2\pi i \langle k, x \rangle} \right\|_{L^2(\mathbb{T}^d)}^2 \\ & \leq c_{d,s} N^{-2s} \sum_{k \in [[N]]^d} \left( \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} (1 + |k + \ell N|^{2s}) |\widehat{v}(k+\ell N)|^2 \right) \\ & \leq c_{d,s} N^{-2s} \|v\|_{H^s(\mathbb{T}^d)}^2. \end{split}$$

Combining the above estimates, we conclude that

$$||v-p||_{L^2} \le c_{d,s} ||v||_{H^s(\mathbb{T}^d)} N^{-s}$$

where we have re-defined  $c_{d,s}:=2^s+(4d)^{s/2}\sum_{\ell\in\mathbb{Z}^d\setminus\{0\}}|\ell|^{-2s}.$ 

We can now state the main outcome of this section:

**Proposition A.6.** Let  $v \in H^s(\mathbb{T}^d)$  be given for s > d/2 and let  $\{v^N(x_n)\}_{n \in [N]^d}$  be any grid values. Let  $p^N(x) = \sum_{k \in [[N]]^d} \mathsf{DFT}(v^N)(k) e^{2\pi i \langle k, x \rangle}$  be the interpolating trigonometric polynomial of  $v^N$ . Then,

$$||v - p^N||_{L^2(\mathbb{T}^d)} \le \frac{1}{N^{d/2}} ||v - v^N||_{\ell^2(n \in [N]^d)} + c_{d,s} ||v||_{H^s(\mathbb{T}^d)} N^{-s}.$$

*Proof.* Let  $p(x) = \sum_{k \in [[N]]^d} \mathsf{DFT}(v)(k) e^{2\pi i \langle k, x \rangle}$  be the interpolating trigonometric polynomial given the point-values  $\{v(x_n)\}_{n \in [N]^d}$ . Then,

$$||v - p^N||_{L^2(\mathbb{T}^d)} \le ||v - p||_{L^2(\mathbb{T}^d)} + ||p - p^N||_{L^2(\mathbb{T}^d)}.$$
 (A.12)

By Lemma A.4, we have

$$||v-p||_{L^2(\mathbb{T}^d)} \le c_{d,s} ||v||_{H^s(\mathbb{T}^d)} N^{-s}.$$

By Lemma A.2, and since  $p(x_n) = v(x_n)$ ,  $p^N(x_n) = v^N(x_n)$  by Lemma A.3, we have

$$||p - p^{N}||_{L^{2}(\mathbb{T}^{d})} = \frac{1}{N^{d/2}} ||p(x_{n}) - p^{N}(x_{n})||_{\ell^{2}(n \in [N]^{d})}$$
$$= \frac{1}{N^{d/2}} ||v(x_{n}) - v^{N}(x_{n})||_{\ell^{2}(n \in [N]^{d})}.$$

Substitution in (A.12) gives the claimed bound.

## **B** Discretization error derivation

In this section, we derive the error breakdown within each FNO layer. This error breakdown is used in the proofs of subsequent sections. Within a single layer, we define the following quantities to track the error origin and propagation, noting that, for values of  $m_t$  that will vary with layer t,  $\mathcal{E}_t^{(j)}: X^{(N)} \to \mathbb{R}^{m_t}, j = 0, 3$  and  $\mathcal{E}_t^{(j)}: [[K]]^d \to \mathbb{C}^{m_t}, j = 1, 2$ .

0. 
$$\mathcal{E}_{t}^{(0)}(x_{n}) = v_{t}^{N}(x_{n}) - v_{t}(x_{n}), \quad x_{n} \in X^{(N)}.$$

1. 
$$\mathcal{E}_t^{(1)}(k) = \frac{1}{N^d} \sum_{n \in [N]^d} v_t(x_n) e^{-2\pi i \langle k, x_n \rangle} - \int_{\mathbb{T}^d} v_t(x) e^{-2\pi i \langle k, x \rangle} \, \mathrm{d}x, \quad k \in [[K]]^d.$$

2. 
$$\mathcal{E}_t^{(2)}(k) = \frac{1}{N^d} \sum_{n \in [N]^d} \mathcal{E}_t^{(0)}(x_n) e^{-2\pi i \langle k, x_n \rangle}, \quad k \in [[K]]^d.$$

3. 
$$\mathcal{E}_t^{(3)}(x_n) = \sum_{k \in [[K]]^d} P_t^{(k)} \left( \mathcal{E}^{(1)}(k) + \mathcal{E}^{(2)}(k) \right) e^{2\pi i \langle k, x_n \rangle}, \quad x_n \in X^{(N)}$$

4. 
$$\mathcal{E}_{t+1}^{(0)}(x_n) = \sigma \left( W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t + W_t \mathcal{E}_t^{(0)}(x_n) + \mathcal{E}_t^{(3)}(x_n) \right) - \sigma (W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t) = v_{t+1}^N(x_n) - v_{t+1}(x_n), \quad x_n \in X^{(N)}$$

Here,  $\mathcal{E}_t^{(0)}$  is the initial error in the inputs to FNO layer t,  $\mathcal{E}^{(1)}$  is the aliasing error,  $\mathcal{E}_t^{(2)}$  is the initial error  $\mathcal{E}_t^{(0)}$  after the discrete Fourier transform, and  $\mathcal{E}_t^{(3)}$  is the error after the operation of the kernel  $\mathcal{K}_t$ . Finally, the initial error for the next layer is given by  $\mathcal{E}_{t+1}^{(0)}$  in terms of the error quantities of the previous layer. Intuitively, the quantity  $\mathcal{E}^{(1)}$  is the source of the error within each layer since it depends only on the ground truth  $v_t$ . All other error quantities are propagation of existing error from previous layers. We provide an exact derivation of these quantities in the following.

Let  $\mathcal{E}_t^{(0)}$  be the error in the inputs to FNO layer t such that

$$\mathcal{E}_t^{(0)}(x_n) = v_t^N(x_n) - v_t(x_n), \quad x_n \in X^{(N)}.$$

Let  $\mathcal{F}(v_t)(k) = \int_{\mathbb{T}^d} v_t(x) e^{-2\pi i \langle k, x \rangle} dx$  denote the Fourier transform and DFT as in equation (A.2). Then for  $k \in [[K]]^d$ ,

$$\mathsf{DFT}(v_t^N)(k) = \frac{1}{N^d} \sum_{n \in [N]^d} v_t(x_n) e^{-2\pi i \langle k, x_n \rangle} + \frac{1}{N^d} \sum_{n \in [N]^d} \mathcal{E}_t^{(0)}(x_n) e^{-2\pi i \langle k, x_n \rangle}$$
$$= \mathcal{F}(v_t)(k) + \mathcal{E}_t^{(1)}(k) + \mathcal{E}_t^{(2)}(k)$$

where  $\mathcal{E}_t^{(1)}$  is the error resulting from computing the Fourier transform of  $v_t$  on a discrete grid rather than all of  $\mathbb{T}^d$ , i.e.

$$\mathcal{E}_t^{(1)}(k) = \frac{1}{N^d} \sum_{n \in \lceil N \rceil^d} v_t(x_n) e^{-2\pi i \langle k, x_n \rangle} - \int_{\mathbb{T}^d} v_t(x) e^{-2\pi i \langle k, x \rangle} \ \mathrm{d}x$$

and  $\mathcal{E}_t^{(2)}$  is the error  $\mathcal{E}_t^{(0)}$  after the discrete Fourier transform, i.e.

$$\mathcal{E}_t^{(2)}(k) = \frac{1}{N^d} \sum_{n \in [N]^d} \mathcal{E}_t^{(0)}(x_n) e^{-2\pi i \langle k, x_n \rangle}.$$

For  $x_n \in X^{(N)}$ , the output of the discrete kernel integral operator acting on  $v_t^N$  is given by

$$\begin{split} (\mathcal{K}_t^N v_t^N)(x_n) &= \sum_{k \in [[K]]^d} P_t^{(k)} \left( \mathcal{F}(v_t)(k) + \mathcal{E}_t^{(1)}(k) + \mathcal{E}_t^{(2)}(k) \right) e^{2\pi i \langle k, x_n \rangle} \\ &= (\mathcal{K}_t v_t)(x_n) + \mathcal{E}_t^{(3)}(x_n) \end{split}$$

where

$$\mathcal{E}_{t}^{(3)}(x_{n}) = \sum_{k \in [[K]]^{d}} P_{t}^{(k)} \left( \mathcal{E}^{(1)}(k) + \mathcal{E}^{(2)}(k) \right) e^{2\pi i \langle k, x_{n} \rangle}.$$

Finally, the output of layer t is given by

$$v_{t+1}^{N}(x_n) = \sigma \left( W_t(v_t(x_n) + \mathcal{E}_t^{(0)}(x_n)) + (\mathcal{K}_t^N v_t^N)(x_n) + b_t \right)$$
  
=  $\sigma \left( W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t + W_t \mathcal{E}_t^{(0)}(x_n) + \mathcal{E}_t^{(3)}(x_n) \right).$ 

Therefore, the initial error for the next layer is given by

$$\mathcal{E}_{t+1}^{(0)}(x_n) = \sigma \left( W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t + W_t \mathcal{E}_t^{(0)}(x_n) + \mathcal{E}_t^{(3)}(x_n) \right) - \sigma \left( W_t v_t(x_n) + \mathcal{K}_t v_t(x_n) + b_t \right).$$

# C Proofs of approximation theory lemmas

We bound the components described in Appendix B in the following proposition.

**Proposition C.1.** *Under Assumptions 3.1, it holds that* 

$$1. \ \|\mathcal{E}_t^{(1)}\|_{\ell^2(k \in [[K]]^d)} \leq \alpha_{d,s} N^{-s} \|v_t\|_{H^s} \ \text{where} \ \alpha_{d,s} \ \text{is independent of} \ N, v_t;$$

2. 
$$\|\mathcal{E}_t^{(2)}\|_{\ell^2(k\in[[N]]^d)} = N^{-d/2} \|\mathcal{E}_t^{(0)}\|_{\ell^2(n\in[N]^d)};$$

3. 
$$\|\mathcal{E}_{t}^{(3)}\|_{\ell^{2}(n\in[N]^{d})} \leq N^{d/2}\|P_{t}\|_{F}\left(\|\mathcal{E}_{t}^{(1)}\|_{\ell^{2}(k\in[[K]]^{d})} + \|\mathcal{E}_{t}^{(2)}\|_{\ell^{2}(k\in[[K]]^{d})}\right);$$

4. 
$$\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2(n\in[N]^d)} \leq B\left(\|W_t\|_2\|\mathcal{E}_t^{(0)}\|_{\ell^2(n\in[N]^d)} + \|\mathcal{E}_t^{(3)}\|_{\ell^2(n\in[N]^d)}\right)$$
.

*Proof.* Beginning with the definition of  $\mathcal{E}_t^{(1)}(k)$ , we have

$$\|\mathcal{E}_t^{(1)}\|_{\ell^2(k \in [[K]]^d)}^2 = \Big\|\frac{1}{N^d} \sum_{n \in [N]^d} v_t(x_n) e^{-2\pi i \langle k, x_n \rangle} - \int_{\mathbb{T}^d} e^{-2\pi i \langle k, x \rangle} v_t(x) \; \mathrm{d}x \Big\|_{\ell^2(k \in [[K]]^d)}^2.$$

Denote the terms in the above expression  $\widehat{v}_t^N(k)$  and  $\widehat{v}_t(k)$ , respectively. Since  $s>\frac{d}{2}$ ,

$$v_t(x_n) = \sum_{k \in \mathbb{Z}^d} \widehat{v}_t(k) e^{2\pi i \langle k, x_n \rangle},$$

and it follows that

$$\widehat{v}_t^N(k') = \frac{1}{N^d} \sum_{n \in [N]^d} \left( \sum_{k \in \mathbb{Z}^d} \widehat{v}_t(k) e^{2\pi i \langle k, x_n \rangle} \right) e^{-2\pi i \langle k', x_n \rangle}$$

$$= \sum_{k \in \mathbb{Z}^d} \widehat{v}_t(k) \frac{1}{N^d} \sum_{n \in [N]^d} e^{2\pi i \langle k - k', x_n \rangle}$$

$$= \sum_{\ell \in \mathbb{Z}^d} \widehat{v}_t(k' + \ell N).$$

Therefore,

$$\begin{split} \|\mathcal{E}_{t}^{(1)}\|_{\ell^{2}(k \in [[K]]^{d})}^{2} &= \|\widehat{v}_{t}^{N} - \widehat{v}_{t}\|_{\ell^{2}(k \in [[K]]^{d})}^{2} \\ &= \sum_{k \in [[K]]^{d}} \left| \sum_{\ell \in \mathbb{Z}^{d} \setminus \{0\}} \widehat{v}_{t}(k + \ell N) \right|^{2} \\ &\leq \sum_{k \in [[K]]^{d}} \left( \sum_{\ell \in \mathbb{Z}^{d} \setminus \{0\}} \frac{1}{|k + \ell N|^{2s}} \right) \sum_{\ell \in \mathbb{Z}^{d} \setminus \{0\}} |k + \ell N|^{2s} |\widehat{v}_{t}(k + \ell N)|^{2} \end{split}$$

by Cauchy-Schwarz. We bound each component separately. It is clear from Definition 2.1 that

$$\sum_{k \in [[K]]^d} \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} |k + \ell N|^{2s} |\widehat{v}_t(k + \ell N)|^2 \le ||v_t||_{H^s}^2.$$
 (C.1)

To bound the first component independently of k, we note from  $K \leq \frac{N}{2}$  and equation (A.10) that

$$\sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} \frac{1}{|k + \ell N|^{2s}} \le \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} \left(\frac{N}{2\sqrt{d}} |\ell|\right)^{-2s}$$
$$\le \alpha_{d,s}^2 N^{-2s}$$

by equation (A.11), where  $\alpha_{d,s}^2 = (4d)^s \sum_{\ell \in \mathbb{Z}^d \setminus \{0\}} |\ell|^{-2s}$  is finite since  $s \geq \frac{d}{2}$ . We express the final bound as

$$\|\mathcal{E}_t^{(1)}\|_{k \in [[K]]^d} \le \alpha_{d,s} N^{-s} \|v_t\|_{H^s}.$$

For  $\mathcal{E}_t^{(2)}(k)$  we have the definition

$$\mathcal{E}_{t}^{(2)}(k) = \frac{1}{N^{d}} \sum_{n \in [N]^{d}} \mathcal{E}_{t}^{(0)}(x_{n}) e^{-2\pi i \langle k, x_{n} \rangle}.$$

By Parseval's Theorem, we have

$$\|\mathcal{E}_t^{(2)}\|_{\ell^2(k\in[[N]]^d)}^2 = \frac{1}{N^d} \|\mathcal{E}_t^{(0)}\|_{\ell^2(n\in[N]^d)}^2. \tag{C.2}$$

For  $P_t \in \mathbb{R}^{d_{v_{t+1}} \times K^d \times d_{v_t}}$  we define the tensor Frobenius norm  $\|P_t\|_F^2 = \sum_{k \in [[K]]^d} \|P_t^{(k)}\|_F^2$ .

$$\begin{split} \|\mathcal{E}_{t}^{(3)}\|_{\ell^{2}(n\in[N]^{d})}^{2} &= \sum_{n\in[N]^{d}} \left| \sum_{k\in[[K]]^{d}} P_{t}^{(k)} \left( \mathcal{E}_{t}^{(1)}(k) + \mathcal{E}_{t}^{(2)}(k) \right) e^{2\pi i \langle k, x_{n} \rangle} \right|^{2} \\ &\leq N^{d} \left| \sum_{k\in[[K]]^{2}} \left| P_{t}^{(k)} (\mathcal{E}_{t}^{(1)}(k) + \mathcal{E}_{t}^{(2)}(k)) \right|^{2} \\ &\leq N^{d} \sum_{k\in[[K]]^{d}} \|P_{t}^{(k)}\|_{F}^{2} \sum_{k\in[[K]]^{d}} \left| \mathcal{E}_{t}^{(1)}(k) + \mathcal{E}_{t}^{(2)}(k) \right|^{2} \\ &= N^{d} \|P_{t}\|_{F}^{2} \|\mathcal{E}_{t}^{(1)} + \mathcal{E}_{t}^{(2)}\|_{\ell^{2}(k\in[[K]]^{d})}^{2} \\ \|\mathcal{E}_{t}^{(3)}\|_{\ell^{2}(n\in[N]^{d})} \leq N^{d/2} \|P_{t}\|_{F} \left( \|\mathcal{E}_{t}^{(1)}\|_{\ell^{2}(k\in[[K]]^{d})} + \|\mathcal{E}_{t}^{(2)}\|_{\ell^{2}(k\in[[K]]^{d})} \right) \end{split}$$

Finally, we have

$$\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^{2}(n\in[N]^{d})}^{2} = \sum_{n\in[N]^{d}} \left| \sigma(W_{t}v_{t} + \mathcal{K}_{t}v_{t} + b_{t} + W_{t}\mathcal{E}_{t}^{(0)}(x_{n}) + \mathcal{E}_{t}^{(3)}(x_{n})) - \sigma(W_{t}v_{t} + \mathcal{K}_{t}v_{t} + b_{t}) \right|^{2}$$

$$\leq \sum_{n\in[N]^{d}} B^{2} \left| W_{t}\mathcal{E}_{t}^{(0)}(x_{n}) + \mathcal{E}_{t}^{(3)}(x_{n}) \right|^{2}$$

$$\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^{2}(n\in[N]^{d})} \leq B\left(\|W_{t}\|_{2}\|\mathcal{E}_{t}^{(0)}\|_{\ell^{2}(n\in[N]^{d})} + \|\mathcal{E}_{t}^{(3)}\|_{\ell^{2}(n\in[N]^{d})}\right)$$

where  $\|\cdot\|_2$  is the matrix-2 norm. Recall B bounds derivatives of  $\sigma$  in Assumptions 3.1.

The results of Proposition C.1 allow us to easily prove the following lemma.

**Lemma C.2.** Under Assumptions 3.1, the following bound holds:

$$\frac{1}{N^{d/2}} \|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2(n \in [N]^d)} \le BM \left( \frac{2}{N^{d/2}} \|\mathcal{E}_t^{(0)}\|_{\ell^2(n \in [N]^d)} + \alpha_{d,s} N^{-s} \|v_t\|_{H^s} \right) \tag{C.3}$$

where  $\alpha_{d,s}$  is a constant dependent only on d and s.

*Proof.* From Proposition C.1, and shortening the notation  $\ell^2(n \in [N]^d)$  to  $\ell^2$ ,

$$\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^{2}} \leq B\left(\|W_{t}\|_{2}\|\mathcal{E}_{t}^{(0)}\|_{\ell^{2}} + N^{d/2}\|P_{t}\|_{F}\left(\alpha_{d,s}N^{-s}\|v_{t}\|_{H^{s}} + N^{-d/2}\|\mathcal{E}_{t}^{(0)}\|_{\ell^{2}}\right)\right)$$

Combining terms gives

$$\|\mathcal{E}_{t+1}^{(0)}\|_{\ell^{2}} \leq B\left(\left(\|W_{t}\|_{2} + \|P_{t}\|_{F}\right)\|\mathcal{E}_{t}^{(0)}\|_{\ell^{2}} + \alpha_{d,s}N^{d/2-s}\|P_{t}\|_{F}\|v_{t}\|_{H^{s}}\right). \tag{C.4}$$

Replacing  $||W_t||_2$  and  $||P_t||_F$  with M and rescaling gives

$$\frac{1}{N^{d/2}} \|\mathcal{E}_{t+1}^{(0)}\|_{\ell^2(n \in [N]^d)} \leq BM\left(\frac{2}{N^{d/2}} \|\mathcal{E}_t^{(0)}\|_{\ell^2(n \in [N]^d)} + \alpha_{d,s} N^{-s} \|v_t\|_{H^s}\right).$$

# D Proofs of regularity theory lemmas

The proof of Lemma 3.5 relies on another result for bounding the  $H^s$  norm of compositions of functions, which is largely taken from the lemma of Moser [18, sec. 2, p. 273] without assuming an  $L^\infty$  norm of v less than 1. We state a proof here for completeness.

**Lemma D.1.** Assume  $\varphi: \mathbb{T}^d \to \mathbb{T}^d$  possesses continuous derivatives up to order r which are bounded by B. Then

$$|\varphi \circ v|_r \le Bc \left(1 + \|v\|_{\infty}^{r-1}\right) \|v\|_{H^r}$$

provided  $v \in H^r(\mathbb{T}^d)$ , where c is a constant dependent on r and d.

Proof. By Faà di Bruno's formula, we have

$$D_x^r(\varphi \circ v(x)) = \sum_{i=1}^r C_{\alpha,r} \frac{d^{\rho} \varphi}{dx^{\rho}}(v(x)) \prod_{i=1}^r (D_x^j v(x))^{\alpha_j}$$
(D.1)

where the sum is over all nonnegative integers  $\alpha_1,\ldots,\alpha_r$  such that  $\alpha_1+2\alpha_2+\cdots+r\alpha_r=r$ , the constant  $C_{\alpha,r}=\frac{r!}{\alpha_1!\alpha_2!2!\alpha_2...\alpha_r!r!\alpha_r}$ , and  $\rho:=\alpha_1+\alpha_2+\cdots+\alpha_r$ .

We seek a bound on square integrals of (D.1). Setting  $v_0 = \frac{d^\rho \varphi}{dx^\rho} v$ ,  $v_\lambda = D_x^\lambda v$ ,  $\alpha_0 = 1$ ,  $p_0 = \infty$ , and  $p_\lambda = \frac{r}{\lambda \alpha_\lambda}$  and noting that  $\sum_{\lambda=0}^r \frac{1}{2p_\lambda} = \frac{1}{2}$ , we have by Hölder's inequality for multiple products that

$$\begin{split} \int_{\mathbb{T}^d} \left| \frac{d^\rho \varphi}{dx^\rho}(v(x)) \prod_{j=1}^r (D_x^j v(x))^{\alpha_j} \right|^2 \, \mathrm{d}x &\leq \int_{\mathbb{T}^d} \prod_{\lambda=0}^r |v_\lambda|^{2\alpha_\lambda} \, \mathrm{d}x \leq \prod_{\lambda=0}^r \left( \int_{\mathbb{T}^d} |v_\lambda|^{2\alpha_\lambda p_\lambda} \, \mathrm{d}x \right)^{1/p_\lambda} \\ &= \|v_0\|_\infty^2 \prod_{\lambda=1}^r \left( \int_{\mathbb{T}^d} |v_\lambda|^{2\alpha_\lambda p_\lambda} \, \mathrm{d}x \right)^{1/p_\lambda} \end{split}$$

The first factor is bounded above by  $B^2$  by assumption. By application of Gagliardo-Nirenberg, the second factor may be bounded by

$$\begin{split} \prod_{\lambda=1}^r \left( \int_{\mathbb{T}^d} |D_x^\lambda v|^{2r/\lambda} \; \mathrm{d}x \right)^{\lambda \alpha_\lambda/r} &\leq C^r \prod_{\lambda=1}^r \|v\|_\infty^{2\alpha_\lambda(1-\lambda/r)} \left( \|D_x^r v\|^2 + \|v\|^2 \right)^{\alpha_\lambda \lambda/r} \\ &\leq C^r \|v\|_\infty^{2\rho-2} \|v\|_{H^r}^2 \end{split}$$

since  $\sum_{\lambda} \lambda \alpha_{\lambda} = r$ , and  $\sum_{\lambda} \alpha_{\lambda} = \rho$ . Combining the bounds,

$$\int_{\mathbb{T}^d} \prod_{\lambda=0}^r |v_{\lambda}|^{2\alpha_{\lambda}} \, \mathrm{d}x \le B^2 C^r \|v\|_{\infty}^{2\rho-2} \|v\|_{H^r}^2.$$

If  $||v||_{\infty} < 1$ , we have the bound

$$\int_{\mathbb{T}^d} \prod_{\lambda=0}^r |v_{\lambda}|^{2\alpha_{\lambda}} \, \mathrm{d}x \le B^2 C^r \|v\|_{H^r}^2, \tag{D.2}$$

and otherwise since  $\rho \leq r$ ,

$$\int_{\mathbb{T}^d} \prod_{\lambda=0}^r |v_{\lambda}|^{2\alpha_{\lambda}} \, \mathrm{d}x \le B^2 C^r \|v\|_{\infty}^{2r-2} \|v\|_{H^r}^2. \tag{D.3}$$

Since these bounds hold for any term in the sum D.1, we obtain

$$|\varphi \circ v|_r \le Bc \left(1 + ||v||_{\infty}^{r-1}\right) ||v||_{H^r}$$
 (D.4)

for a different constant c depending on r and d.

Now we may prove Lemma 3.5.

**Lemma 3.5.** *Under Assumptions 3.1, the following bounds hold:* 

• 
$$||v_{t+1}||_{\infty} \le \sigma^* + BM(1 + ||v_t||_{\infty} + K^{d/2}||v_t||_{L^2(\mathbb{T}^d)}).$$

• 
$$|v_{t+1}|_s \le BcM^sK^{ds/2}(1+||v_t||_{\infty})^s(1+|v_t|_s)$$

for some constant c dependent on d and s, where  $\sigma^* := \max\{\max_{0 \le t \le T-1} \sigma_t(0), 1\}$ .

*Proof.* First we bound  $\|\mathcal{K}_t v_t\|_{\infty}$ . Recall  $\widehat{v}_t(k) := \int_{\mathbb{T}^d} v_t(x) e^{-2\pi i \langle k, x \rangle} dx$ .

$$\begin{split} \|\mathcal{K}_{t}v_{t}\|_{\infty} &= \|\sum_{k \in [[K]]^{d}} P_{t}^{(k)} \widehat{v}_{t}(k) e^{2\pi i \langle k, x \rangle} \|_{\infty} \\ &\leq \sum_{k \in [[K]]^{d}} \|P_{t}^{(k)}\| \|\widehat{v}_{t}(k)\| \\ &\leq \left(\sum_{k \in [[K]]^{d}} \|P_{t}^{(k)}\|^{2}\right)^{1/2} \|\widehat{v}_{t}\|_{\ell^{2}(k \in [[K]]^{d})} \\ &\leq \|P_{t}\|_{F} K^{d/2} \|\widehat{v}_{t}\|_{\ell^{2}(k \in [[K]]^{d})} \\ &\leq \|P_{t}\|_{F} K^{d/2} \|v_{t}\|_{L^{2}(\mathbb{T}^{d})}. \end{split}$$

Then

$$||W_t v_t + \mathcal{K}_t v_t + b_t||_{\infty} \le ||W_t||_2 ||v_t||_{\infty} + |b_t| + ||P_t||_F K^{d/2} ||v_t||_{L^2(\mathbb{T}^d)},$$

and by Lipschitzness of  $\sigma$  we have

$$||v_{t+1}||_{\infty} \le \sigma^* + BM \left( 1 + ||v_t||_{\infty} + K^{d/2} ||v_t||_{L^2(\mathbb{T}^d)} \right).$$

Next we bound  $|v_{t+1}|_s$ . Letting  $f_t = W_t v_t + \mathcal{K}_t v_t + b_t$ , we see from Lemma D.1 that bounding  $||f_t||_{H^s}$  will give the result.

$$\begin{split} D_x^s(f_t) &= W_t(D_x^s v_t) + \mathcal{K}_t(D_x^s v_t). \\ \int_{\mathbb{T}^d} |D_x^s(f_t)|^2 \, \mathrm{d}x &\leq 2 \left( \int_{\mathbb{T}^d} |W_t(D_x^s v_t)|^2 \, \mathrm{d}x + \int_{\mathbb{T}^d} |\mathcal{K}_t(D_x^s v_t)|^2 \, \mathrm{d}x \right) \end{split}$$

The first integral on the right may be bounded by  $||W_t||_2^2 |v_t|_s^2$ . To bound the second integral,

$$\int_{\mathbb{T}^d} |\mathcal{K}_t(D_x^s v_t)|^2 \, \mathrm{d}x = \int_{\mathbb{T}^d} \left| \sum_{k \in [[K]]^d} P_t^{(k)} \widehat{g}_t(k) e^{2\pi i \langle k, x \rangle} \right|^2 \, \mathrm{d}x$$

where  $\hat{q}_t(k)$  are the Fourier coefficients of  $D_x^s v_t$ . Continuing,

$$\begin{split} \int_{\mathbb{T}^d} |\mathcal{K}_t(D_x^s v_t)|^2 \, \mathrm{d}x & \leq \int_{\mathbb{T}^d} \|P_t\|_F^2 \sum_{k \in [[K]]^d} |\widehat{g}_t(k)|^2 \, \mathrm{d}x \\ & \leq \|P_t\|_F^2 \|D_x^s v_t\|_{L^2}^2, \end{split}$$

giving a bound of

$$|f_t|_s \leq 2M|v_t|_s$$

In the following,  $\lesssim$  denotes inequality up to a constant multiple that does not depend on any of the variables involved. Combining Lemma D.1 and the above bounds, we have

$$\begin{split} |\sigma \circ f_{t}|_{s} &\leq Bc(1 + \|f_{t}\|_{\infty}^{s-1})\|f_{t}\|_{H^{s}} \\ &\leq Bc(1 + (M(1 + \|v_{t}\|_{\infty} + K^{d/2}\|v_{t}\|_{\infty}))^{s-1})(M(1 + \|v_{t}\|_{\infty} + K^{d/2}\|v_{t}\|_{\infty}) + 2M|v_{t}|_{s}) \\ &\lesssim BcM^{s}K^{ds/2}(1 + (1 + \|v_{t}\|_{\infty})^{s-1})(1 + \|v_{t}\|_{\infty} + |v_{t}|_{s}) \\ &\lesssim BcM^{s}K^{ds/2}(1 + \|v_{t}\|_{\infty})^{s-1}(1 + \|v_{t}\|_{\infty})(1 + |v_{t}|_{s}) \\ &\lesssim BcM^{s}K^{ds/2}(1 + \|v_{t}\|_{\infty})^{s}(1 + |v_{t}|_{s}). \end{split}$$

## E Proof of Theorem 3.2

**Theorem 3.2.** Let Assumptions 3.1 hold. Let  $A_c$  be a compact set in A. Let  $v_t(a) := L_{t-1} \circ \cdots \circ L_0 \circ \mathcal{P}(a)$  with  $\mathcal{P}$  and each L as defined in Definition 2.1. Similarly, let  $v_t^N(a) := L_{t-1}^N \circ \cdots \circ L_0^N \circ \mathcal{P}(a)$  where  $L_j^N v_j^N = \sigma_j(W_j v_j^N + \mathcal{K}_j^N v_j^N + b_j)$  for  $\mathcal{K}_j^N$  defined in (3.1) for each  $0 \le j \le t-1$ . Then

$$\sup_{a \in \mathcal{A}_c} \frac{1}{N^{d/2}} \| v_t(a) - v_t^N(a) \|_{\ell^2(n \in [N]^d)} \le C N^{-s}$$
(3.2)

where the constant C depends on B, M, d, s, t, and  $A_c$ .

*Proof.* Temporarily dropping the notational dependence of  $v_t$  and  $v_0$  on a, from Lemma 3.5 we have for t > 1,

$$||v_t||_{\infty} \lesssim \sigma^* \sum_{j=0}^{t-1} (BMK^{d/2})^j + \sum_{j=1}^t (BMK^{d/2})^j + (BMK^{d/2})^t ||v_0||_{\infty}$$
$$|v_t|_s \lesssim \left( \sum_{j=1}^t (BcM^s K^{ds/2})^j \prod_{\ell=t-j}^{t-1} (1 + ||v_\ell||_{\infty})^s \right) + (BcM^s K^{ds/2})^t \left( \prod_{\ell=0}^{t-1} (1 + ||v_\ell||_{\infty})^s \right) |v_0|_s.$$

Denote  $\max\{BMK^{d/2}, B^{1/s}c^{1/s}MK^{d/2}, 1\}$  by  $C_0$ . Since  $\sigma^* \geq 1$ , the bound on  $\|v_t\|_{\infty}$  simplifies to

$$||v_t||_{\infty} \lesssim \sigma^* \sum_{j=1}^t C_0^j + C_0^t ||v_0||_{\infty}$$
  
$$\leq \sigma^* t C_0^t + C_0^t ||v_0||_{\infty}$$

Plugging in this bound to the product in the bound on  $|v_t|_s$ , we have

$$\prod_{\ell=t-j}^{t-1} (1 + \|v_{\ell}\|_{\infty})^{s} \lesssim \prod_{\ell=t-j}^{t-1} (1 + \ell\sigma^{*}C_{0}^{\ell} + C_{0}^{\ell}\|v_{0}\|_{\infty})^{s} 
\lesssim C_{0}^{tsj}(t)^{sj} (\sigma^{*} + \|v_{0}\|_{\infty})^{sj}.$$

Combining these two bounds, we attain the following bound on  $|v_t|_s$  for  $t \ge 1$ .

$$|v_t|_s \lesssim \left(\sum_{j=1}^t (C_0)^{sj} C_0^{tsj}(t)^{sj} (\sigma^* + ||v_0||_{\infty})^{sj}\right) + C_0^{ts} \left(C_0^{t^2s}(t)^{st} (\sigma^* + ||v_0||_{\infty})^{st}\right) |v_0|_s$$

$$\lesssim \left(\sum_{j=1}^t C_0^{2tsj}(t)^{sj} (\sigma^* + ||v_0||_{\infty})^{sj}\right) + C_0^{2t^2s}(t)^{st} (\sigma^* + ||v_0||_{\infty})^{st} |v_0|_s$$

$$\lesssim (C_0^{2t^2s} t^{st+1} + C_0^{2t^2s} t^{st} |v_0|_s) (\sigma^* + ||v_0||_{\infty})^{st}$$

and the following bound on  $||v_t||_{H^s}$ 

$$||v_t||_{H^s} \lesssim (C_0^{2t^2s} t^{st+1} |v_0|_s) (\sigma^* + ||v_0||_{\infty})^{st} + \sigma^* t C_0^t + C_0^t ||v_0||_{\infty}.$$
 (E.1)

Recall that  $v_0 = \mathcal{P}(a)$ , and  $\mathcal{P}$  is a shallow neural network, which is a special case of a Fourier layer where the coefficients  $P_t^{(k)}$  are set to 0. Assumptions 3.1 include boundedness of the coefficients of  $\mathcal{P}$  by M. Thus we may increment t by 1 in the bound and write

$$\sup_{a \in A} \|v_t(a)\|_{H^s} \tag{E.2a}$$

$$\lesssim \sup_{a \in \mathcal{A}_c} (C_0^{2(t+1)^2 s} (t+1)^{s(t+1)+1} |a|_s) (\sigma^* + ||a||_{\infty})^{s(t+1)} + \sigma^* (t+1) C_0^{t+1} + C_0^{t+1} ||a||_{\infty}.$$
(E.2b)

Since  $\mathcal{A}$  is a compact set in  $H^s$ , and  $s>\frac{d}{2}$ , both  $\|a\|_{\infty}$  and  $|a|_s$  are bounded uniformly over  $\mathcal{A}$  by a constant depending on  $\mathcal{A}$  since  $H^s$  is continuously embedded in  $L^{\infty}$ . Thus, we may denote this upper bound by  $C_1$ , which does not depend on N. Let  $\mathcal{E}^{(0)}_{t+1}(a)=v^N_t(a)-v_t(a)$ . Then from Lemma C.2, we have

$$\sup_{a \in \mathcal{A}_c} \frac{1}{N^{d/2}} \|\mathcal{E}_{t+1}^{(0)}(a)\|_{\ell^2(n \in [N]^d)} \lesssim BM \left( \frac{2}{N^{d/2}} \sup_{a \in \mathcal{A}_c} \|\mathcal{E}_t^{(0)}(a)\|_{\ell^2(n \in [N]^d)} + \alpha_{d,s} N^{-s} C_1 \right).$$

By the discrete Gronwall lemma,

$$\begin{split} \sup_{a \in \mathcal{A}_c} \frac{1}{N^{d/2}} \|\mathcal{E}_t^{(0)}(a)\|_{\ell^2(n \in [N]^d)} \\ \lesssim \frac{BM\alpha_{d,s} N^{-s} C_1}{1 - 2BM} (1 - (2BM)^t) + \frac{1}{N^{d/2}} \sup_{a \in \mathcal{A}_c} \|\mathcal{E}_0^{(0)}(a)\|_{\ell^2(n \in [N]^d)} (2BM)^t. \end{split}$$

Since we assume we begin with no error,  $\|\mathcal{E}_0^{(0)}(a)\|_{\ell^2(n\in[N]^d)}=0$ , this simplifies to

$$\sup_{a \in \mathcal{A}_c} \frac{1}{N^{d/2}} \|\mathcal{E}_t^{(0)}(a)\|_{\ell^2(n \in [N]^d)} \lesssim \frac{BM\alpha_{d,s}C_1}{1 - 2BM} (1 - (2BM)^t) N^{-s}.$$

Denoting the factor in front of  $N^{-s}$  by C and absorbing the effects of  $\lesssim$  into C, we have the result that

$$\sup_{a \in A_s} \frac{1}{N^{d/2}} \|v_t(a) - v_t^N(a)\|_{\ell^2(n \in [N]^d)} \le CN^{-s}.$$

**Remark E.1.** A trivial consequence of the above theorem is that under Assumptions 3.1,

$$\lim_{N \to \infty} \sup_{a \in \mathcal{A}_c} \frac{1}{N^{d/2}} \|v_t^N(a) - v_t(a)\|_{\ell^2(n \in [N]^d)} = 0.$$

Indeed, a stronger result holds that the discrete  $\ell^{\infty}$  norm converges at a rate  $N^{-s+d/2}$  by a straightforward inverse inequality.

#### F Proof of Theorem 3.3

**Theorem 3.3.** Let  $p_t^N(a)(x) = \sum_{k \in [[N]]^d} \mathsf{DFT}(v_t^N(a))(k) e^{2\pi i \langle k, x \rangle}$  denote the interpolating trigonometric polynomial of  $\{v_t^N(a)(x_n)\}_{n \in [N]^d}$ . Let the assumptions of Theorem 3.2 hold. Then,

$$\sup_{a \in \mathcal{A}_c} \|v_t(a) - p_t^N(a)\|_{L^2(\mathbb{T}^d)} \le C' N^{-s}. \tag{3.3}$$

Here, C' depends on B, M, d, s, t, and A.

*Proof.* We temporarily drop the dependence of  $p_t^N$  and  $v_t^N$  on a. Let  $p_t^N(x)$  be the interpolating trigonometric polynomial associated with the data  $\{v_t^N(x_n)\}_{n\in[N]^d}$ . By Proposition A.6, we have

$$||v_t - p_t^N||_{L^2(\mathbb{T}^d)} \le \frac{1}{N^{d/2}} ||v_t - v_t^N||_{\ell^2(n \in [N]^d)} + c_{d,s} ||v||_{H^s(\mathbb{T}^d)} N^{-s}.$$

22

By (E.2), we have  $\sup_{a \in \mathcal{A}_c} \|v_t(a)\|_{H^s(\mathbb{T}^d)} \leq C_1$ . Furthermore, it follows from Theorem 3.2, that

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

We conclude that

$$\sup_{a \in \mathcal{A}_c} \|v_t(a) - p_t^N(a)\|_{L^2(\mathbb{T}^d)} \le (C + c_{d,s}C_1)N^{-s}$$

Thus, the claimed bound holds with  $C' = C + c_{d,s}C_1$ .

#### G Additional numerical results

Figure 8 addresses the question of error decreasing or increasing with layer count. The figure shows that when the FNO weights are randomly initialized with the default initialization and then multiplied by 10, the error increases with the number of layers instead of decreases. Additionally, in this model, the large weights mean that the GeLU activation acts like a ReLU activation for smaller discretizations. This phenomenon is apparent for inputs with regularity s=2, where the first layer has the appropriate slope, but the other layers only begin to approach that rate at higher discretizations. Earlier layers achieve this rate first because of the smaller magnitude state norm in earlier layers for this model.

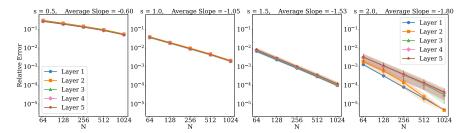


Figure 8: Relative error versus N and s for an FNO with default  $\times 10$  initial weights.

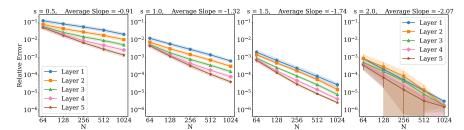


Figure 9: Relative error versus N and s for an FNO with all weights equals to 1.

As an alternative setting of the weights, Figure 9 shows the discretization error when all the weights are set to 1. In this case, the error is more erratic. The error decreases faster than expected and with less consistency than the Gaussian weight models, and the decay rate increases with each layer. In this sense, the all-ones model has a smoothing effect on the state at each layer. We note that this generally occurs with any initialization that sets the spectral weights on the same order of magnitude as the affine weights; for instance, the same super-convergence effect occurs when all weights are initialized U(0,1). We hypothesize that this is because when the spectral weights are of equal magnitude to the affine weights, the function is progressively smoothed as it passes through the model.

We can also observe the state norm as the state passes through the layers for various settings of the weights. Indeed, for all choices of initialization that we explored except the default setting, the state norm increases exponentially through the layers, while for the default initialization the magnitude stays roughly constant. This phenomenon is illustrated in Figure 10.

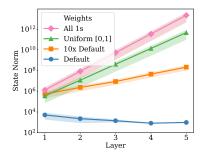


Figure 10: State norm versus layer for various untrained model initializations.

# H Additional implementation details for error analysis experiments

All the trained models were trained on an Nvidia P100 GPU for approximately 6 hours. The evaluation scripts were run on a Mac laptop with an M2 processor.

# I Implementation details for adaptive subsampling

Our model has 4 hidden layers, channel width 64 and Fourier cut-off 12. Our results are based on 9000 training samples and 500 test samples. For training with a subsampling scheduler, we include an additional 500 samples for validation. Models are trained for 300 epochs on an Nvidia P100 GPU.