Journal of Machine Learning Research 26 (2025) 1-52 Submitted 6/24; Revised 10/25; Published 12/25

Continuum Attention for Neural Operators

Edoardo Calvello E.CALVELLO@CALTECH.EDU
California Institute of Technology

Nikola B. Kovachki NKOVACHKI@NVIDIA.COM
NVIDIA

Matthew E. Levine MATT@BASIS.AI
The Broad Institute of MIT and Harvard
Basis Research Institute

Andrew M. Stuart ASTUART@CALTECH.EDU
California Institute of Technology

Editor: Lorenzo Rosasco

Abstract

Transformers, and the attention mechanism in particular, have become ubiquitous in machine learning. Their success in modeling nonlocal, long-range correlations has led to their widespread adoption in natural language processing, computer vision, and time series problems. Neural operators, which map spaces of functions into spaces of functions, are necessarily both nonlinear and nonlocal if they are universal; it is thus natural to ask whether the attention mechanism can be used in the design of neural operators. Motivated by this, we study transformers in the function space setting. We formulate attention as a map between infinite dimensional function spaces and prove that the attention mechanism as implemented in practice is a Monte Carlo or finite difference approximation of this operator. The function space formulation allows for the design of transformer neural operators, a class of architectures designed to learn mappings between function spaces. In this paper, we state and prove the first universal approximation result for transformer neural operators, using only a slight modification of the architecture implemented in practice. The prohibitive cost of applying the attention operator to functions defined on multi-dimensional domains leads to the need for more efficient attention-based architectures. For this reason we also introduce a function space generalization of the patching strategy from computer vision, and introduce a class of associated neural operators. Numerical results, on an array of operator learning problems, demonstrate the promise of our approaches to function space formulations of attention and their use in neural operators.

Keywords: Operator Learning, Attention, Transformers, Discretization Invariance, Partial Differential Equations.

1. Introduction

Attention was introduced in the context of neural machine translation in Bahdanau et al. (2014) to effectively model correlations in sequential data. Many attention mechanisms used in conjunction with recurrent networks were subsequently developed, for example long short-term memory (Hochreiter and Schmidhuber, 1997) and gated recurrent units (Chung et al., 2014), which were then employed for a wide range of applications, see for example the review Chung et al. (2014). The seminal work Vaswani et al. (2017) introduced the now ubiquitous transformer architecture, which relies solely on the attention mechanism and not on the recurrence of the network in order to effectively model nonlocal, long-range correlations. In its full generality, the attention function described in Vaswani et al. (2017) defines an operation between sequences $\{u_i\}_{i=1}^N \subset \mathbb{R}^{d_u}, \{v_i\}_{i=1}^M \subset \mathbb{R}^{d_v}$ of lengths $N, M \in \mathbb{N}$ respectively, represented as tensors $u \in \mathbb{R}^{N\times d_u}, v \in \mathbb{R}^{M\times d_v}$, where learnable weights $Q \in \mathbb{R}^{d_K \times d_u}, K \in \mathbb{R}^{d_K \times d_v}, V \in \mathbb{R}^{d_V \times d_v}$ parametrize

$$\texttt{attention}(u, v) := \texttt{softmax}(uQ^T K v^T) v V^T. \tag{1}$$

In this context, we recall that the function $\texttt{softmax} : \mathbb{R}^{N \times M} \to \mathbb{R}^{N \times M}$ is defined component-wise via its action on the input as

$$\big(\texttt{softmax}(w)\big)_{ij} = \frac{\exp(w_{ij})}{\sum_{j=1}^{M} \exp(w_{ij})}, \tag{2}$$

for any $w \in \mathbb{R}^{N \times M}$. Therefore, in this form the attention function defines a mapping $\texttt{attention} : \mathbb{R}^{N \times d_u} \times \mathbb{R}^{M \times d_v} \to \mathbb{R}^{N \times d_V}$.[1] Alternatively, we can view the output of the attention function as a sequence $\{\texttt{attention}(u, v)_i\}_{i=1}^N \subset \mathbb{R}^{d_V}$.

Viewed more generally, attention defines a mapping from two sequences $\{u_i\}_{i=1}^N$ and $\{v_i\}_{i=1}^M$ into a third sequence $\{\texttt{attention}(u, v)_i\}_{i=1}^N$; all three sequences take values in $\mathbb{R}^r$ for various values of r. In this paper, we generalize this in two ways: (a) we replace index i by index $x \in D$, where D is a bounded open subset in $\mathbb{R}^d$; (b) we retain a discrete index i but allow the sequences to take values in Banach spaces of $\mathbb{R}^r$-valued functions defined over D, where D is a bounded open subset in $\mathbb{R}^d$. We find the first generalization (a) most useful in settings where $d = 1$, that is for time series. However, when discretizing the index set $D \subset \mathbb{R}^d$ with $N := n^d$ points, the quadratic scaling of the attention mechanism in N can be prohibitive. This motivates generalization (b) where we retain the discrete index but allow sequences to be function valued, corresponding to patching of functions defined over $D \subset \mathbb{R}^d$, with $d \geq 2$.

Our motivation for these generalizations is operator learning and, in particular, design of discretization invariant neural operator architectures. Such architectures disentangle model parameters from any particular discretization of the input functional data; learning is focused on the intrinsic continuum problem and not any particular discretization of it. This allows models to obtain consistent errors when trained with different resolution data, as well as enabling training, testing, and inference at different resolutions. We consider the general setting where $D \subset \mathbb{R}^d$ is a bounded open set and $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^{d_u})$ and $\mathcal{Z} = \mathcal{Z}(D; \mathbb{R}^{d_z})$ are separable Banach spaces of functions. To train neural operators we consider the following standard data scenario. The set $\{u^{(j)}, z^{(j)}\}_{j=1}^J$ of input-output observation pairs is such that $\big(u^{(j)}, z^{(j)}\big) \sim \nu$ is an i.i.d. sequence distributed according to the measure ν, which is supported on $\mathcal{U} \times \mathcal{Z}$. We let μ and π denote the marginals of ν on $\mathcal{U}$ and $\mathcal{Z}$ respectively, so that $u^{(j)} \sim \mu$ and $z^{(j)} \sim \pi$. In what follows, we consider $\mathsf{G}^\dagger : \mathcal{U} \to \mathcal{Z}$ to be an arbitrary map so that

$$z^{(j)} = \mathsf{G}^\dagger\big(u^{(j)}\big), \tag{3}$$

noting that such maps will typically be both nonlocal (with respect to domain D) and nonlinear. We assume for simplicity that the data generation mechanism is compatible with (3) so that $\pi = \mathsf{G}^\dagger_\sharp \mu$, i.e., the data generation mechanism is not corrupted by noise. Neural operators construct an approximation of $\mathsf{G}^\dagger$ picked from the parametric class

$$\mathsf{G} : \mathcal{U} \times \Theta \to \mathcal{Z};$$

equivalently we may write $\mathsf{G}_\theta : \mathcal{U} \to \mathcal{Z}$, for $\theta \in \Theta$. The set Θ is assumed to be a finite dimensional parameter space from which a $\theta^* \in \Theta$ is selected so that $\mathsf{G}(\cdot, \theta^*) = \mathsf{G}_{\theta^*} \approx \mathsf{G}^\dagger$. We then define a cost functional $c : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ and define θ^* by

$$\theta^* = \arg\min_{\theta \in \Theta} \mathbb{E}_{u \sim \mu}\Big[c\big(\mathsf{G}(u, \theta), \mathsf{G}^\dagger(u)\big)\Big]. \tag{4}$$

1. We note that in most implementations, the input of the softmax function is rescaled by a factor of $1/\sqrt{d_K}$. In our context, continuum attention, a different scaling will be used.

In practice, we will only have access to the values of each $u^{(j)}$ and $z^{(j)}$ at a set of discretization points of the domain D, which we denote as $\{x_i\}_{i=1}^N$. We seek discretization invariant approaches to operator learning which allow for sharing of learned parameters θ^* between different discretizations.

The choice of an architecture defining the approximation family G_θ is sometimes known as surrogate modeling. We use continuum attention concepts to make this choice. The resulting transformer based neural operator methodology that we develop here can be used to learn solution operators to parametric PDEs, to solve PDE inverse problems, and may also be applied to solve a class of data assimilation problems. In the context of parametric PDEs we will explore how this algorithmic framework may be employed in operator learning for Darcy flow and 2d Navier-Stokes problems. As a particular case of operator learning, we will also explore a class of data assimilation problems that involve recovering unobserved trajectories of possibly chaotic dynamical systems. Namely, we illustrate examples involving the Lorenz-63 dynamical system and a controlled ODE.

1.1 Literature Review

The attention mechanism, as introduced in Vaswani et al. (2017) and defined in (1), has shown widespread success for a variety of tasks. In particular, given the suitability of attention and transformers for applications involving sequential data, there has been a large body of work extending the methodology from natural language processing, the setting in which it was originally introduced, to the more general context of time series. Namely, as surveyed in Wen et al. (2023), transformers and variants thereof have been applied to time series forecasting, classification and anomaly detection tasks. This generalization has involved modifications and innovations relating to the positional encoding and the design of the attention function itself. We refer the reader to Wen et al. (2023) for a detailed review of the large body of work and architectural variants of transformers that have been developed for time series applications.

Attention in Vision The quadratic scaling in the input of the attention mechanism has made transformers a computationally prohibitive architecture for applications involving long sequences. Nonetheless, through techniques such as patching, the attention mechanism and transformers have also enjoyed success in computer vision, where high-resolution image data presents an increased computational cost. In fact, in vision transformers (ViT), introduced in Dosovitskiy et al. (2021), the input image is subdivided into patches; attention is then applied across the shorter sequence of patches, thus decreasing the overall cost.

Neural Operators Given the success of the methodology in computer vision, there has been growing interest in extending the transformer methodology to the operator learning context. This operator setting concerns learning mappings between infinite dimensional spaces of functions. In this case, the domains on which the functions are defined (spatial and temporal) may inherently be a continuum. Neural operators (Kovachki et al., 2023), generalizations of neural networks that learn operators mapping between infinite-dimensional spaces, have been developed for this task. As they parametrize operators acting on spaces of functions, the learned parameters of neural operators may be applied to any discretization of the input function; this property of neural operators is referred to as “discretization invariance”. Discretization invariance is desirable both at a theoretical and practical level. Indeed, for a neural network to approximate an operator acting on functions defined on a continuum, its parameters cannot depend on the discretizations of the input functions that are used for learning the parameters themselves. On the other hand, discretization invariance allows to scalably train models on coarser discretizations of functions and to deploy or finetune trained models at different resolutions for downstream tasks. A recent example of the success of this approach is the GenCast medium-range weather forecasting model (Price et al., 2025), which because of discretization invariance properties is pretrained on low resolutions and finetuned at the desired predictive resolution.

Transformers for Operator Learning In the following paragraphs, we discuss several approaches in the literature employing transformers for operator learning. We first discuss some approaches leading to neural operators, followed by a discussion of architectures that are not discretization invariant.

Neural Operators. An integral kernel interpretation of the attention mechanism appearing for example in Tsai et al. (2019); Martins et al. (2020); Moreno et al. (2022); Wright and Gonzalez (2021); Cao (2021); Kovachki et al. (2023) inspired a range of attention-based architectures for operator learning. For example, the work of Cao (2021) leverages insight from the integral kernel formulation to design novel attention mechanisms that do not employ the softmax function. The author designs attention mechanisms using integral kernels which may be approximated by a Fourier-type projection, leading to quadratic complexity, and by a Galerkin projection, leading to linear complexity. While the resulting attention maps are discretization invariant, the full architecture proposed uses convolutional neural networks to downsample the input to a feature map of sufficiently small resolution for the method to be scalable. Hence the full architecture is not a neural operator. Furthermore, the lack of the softmax nonlinearity leads to a mechanism lacking the probabilistic description outlined in this paper, thus differing from the original definition of attention in Vaswani et al. (2017). In Li et al. (2024) the authors take the Fourier attention approach as in Cao (2021) to design a neural operator for large eddy simulation, while in Luo et al. (2024) the authors take the Galerkin attention approach from Cao (2021) to design a novel neural operator architecture. In Li et al. (2023) the authors present "OFormer" a neural operator with an encoder-decoder structure that employs cross-attention to obtain the latent embeddings and then stacked self-attention blocks; the model uses the attention from Cao (2021) and random Fourier features for query coordinate encodings while using a recurrent multi-layer perceptron for unrolling in time-dependent problems as opposed to masked attention in Vaswani et al. (2017). In the recent work Rahman et al. (2024), the authors propose a neural operator architecture that employs an attention mechanism that acts across channels of the function in the latent embedding, and do so in the context of pre-training a foundation model for PDE solution operators. We demonstrate how such an architecture can be derived from our formulation in Remark 16. As in our paper, various attention-related concepts are defined in a function-space setting, but our formulations are more general; this issue is discussed in detail at the relevant point in the paper. We note that the architectures introduced in this paper rely solely on the attention mechanism defined in the continuum, hence they are neural operators; furthermore, because of the simple design, with a slight modification to our architectures we may prove the first universal approximation theorem for transformer operators.

Non discretization invariant architectures. In Guibas et al. (2022) the authors propose a modification of the Fourier neural operator that involves mixing of tokens in Fourier space. This is achieved by first applying the FFT to the sequence of tokens and then applying a MLP to each token, before inverting back to the original space. This approach differs fundamentally from the original definition of attention in Vaswani et al. (2017) and lacks the probabilistic description outlined in this paper. Furthermore, because of learnable positional embeddings and patching via convolution, the AFNO is not discretization invariant. Various other approaches have been developed. The work Hao et al. (2023) introduces "GNOT", an architecture employing a normalized attention layer that handles multiple inputs. This design, along with a learned gating mechanism for input coordinates, allows the model to be applied to irregular meshes and multi-scale problems; however, this architecture employs an attention mechanism that does not involve rescaling by the grid, hence not directly allowing for discretization invariance. In Ovadia et al. (2024) the authors propose to use an architecture composed of a ViT in the latent space of a U-Net (Ronneberger et al., 2015). The proposed model takes as input the input field evaluated on a grid with associated coordinate values and outputs an approximation of the solution to PDE inverse problems. While it is shown that the architecture achieves state-of-the-art accuracy, the parameters in the scheme are not invariant to the input function resolution. In Wang et al. (2025) the authors design "Continuous ViT", an architecture consisting of a ViT encoder and a cross-attention decoder. The method handles input functions defined on space and time and the method is shown to present evidence of discretization invariance, with test discretizations close to training discretizations. However, while the architecture is discretization invariant in time, it is not by design in space. This is because the tokenization procedure employed by standard ViTs is not discretization invariant; we elaborate on this point in Remark 15.

Transformers for Continuum Data The use of attention-based transformer models in the context of dynamical systems and data assimilation problems, where spatial and temporal domains of the systems are continuous, is at its infancy. For a comprehensive review on the application of machine learning to data assimilation we refer the reader to Cheng et al. (2023). In this context, the efficacy of transformers has been explored for forecasting in numerical weather prediction. In Chattopadhyay et al. (2022), a transformer module is integrated in the latent space of a U-Net (Ronneberger et al., 2015) architecture in order to leverage the permutation equivariance property of the transformer to improve physical consistency and forecast accuracy. The work in Bi et al. (2023) achieves state-of-the-art performance among end-to-end learning methods for numerical weather prediction by applying a ViT-like architecture and employing the shifted window methodology from Liu et al. (2021). These approaches, however, are not discretization invariant by design.

Mathematics of Transformers Mathematical foundations for the methodology are only now starting to emerge, however most analysis has been preformed in the finite-dimensional setting. In the recent work Geshkovski et al. (2023), the authors study the attention dynamics by viewing tokens as particles in an interacting particle system. The formulation of the continuous-time limit of the dynamics, where the limit is taken in "layer time" allows to investigate the emergence of clusters among tokens. In Yun et al. (2020), the authors prove that transformers are universal approximators of continuous permutation equivariant sequence-to-sequence functions with compact support. More broadly, developing a firm theoretical framework for the subject is necessary to understand the empirical performance of the methodology and to inform the design of novel attention-based architectures.

This Work This paper bridges the gap between the recent theoretical developments for the attention mechanism and practical application in operator learning. In fact, starting from a description of the attention mechanism as a map on spaces of functions, we build discretization invariant transformer blocks that may be directly deployed for operator learning tasks. A numerical investigation shows that the ensuing transformer neural operators are competitive with state-of-the-art architectures for a range of operator learning tasks. The framework developed may be used more generally to design larger, more complex architectures for which the universal approximation theorem proved in this paper may potentially be generalized.

1.2 Contributions and Outline

Motivated by the success of transformers for sequential data, we set out to formulate the attention mechanism and transformer architectures as mappings between infinite dimensional spaces of functions. In this work we introduce a continuum perspective on the transformer methodology. The resulting theoretical framework enables the design of attention-based neural operator architectures, which find a natural application to operator learning problems. The strength of the approach of devising implementable discrete architectures from the continuum perspective stems from the discretization invariance property that the schemes inherit. Indeed, the resulting neural operator architectures are designed as mappings between infinite dimensional functions spaces that are invariant to the particular discretization, or resolution, at which the input functions are defined. Notably, this allows for zero-shot generalization to different function discretizations. With these insights in view, our main contributions can be summarized as follows:

1. Formulation of attention as a mapping between function spaces.

2. Approximation theorem that quantifies the error between application of the attention operator to a continuous function and the result of applying its finite dimensional analogue to a discretization of the same function. The relevant results may be found in Theorems 6 and 12.

3. A formulation of the transformer architecture as a neural operator, hence as a mapping between function spaces; the resulting scheme is resolution invariant and allows for zero-shot generalization to irregular, non-uniform meshes.

4. A first universal approximation theorem for transformer neural operators. The relevant results may be found in Theorems 22 and 23.

5. Formulation of patch-attention as a mapping between spaces of functions acting on patch indices. This continuum perspective leads to the design of efficient attention-based neural operator architectures.

6. Numerical evidence of the competitiveness of transformer-based neural operators with state-of-the-art operator learning architectures.

After introducing, in Subsection 1.3, the notation that will be used throughout, Section 2 is focused on contributions 1 and 2, formulating attention as acting on functions defined on a continuous domain. In Section 3 we formulate the attention mechanism as acting on a sequence of “patches” of a function, addressing contribution 5. In Section 4 we make use of the operator frameworks developed in Sections 2 and 3 to formulate transformer neural operator architectures acting on function space. Thus we address 3, whilst the last two subsections concern the algorithmic aspect of extending to patching and address contribution 5. In Section 5 we state and prove a first universal approximation theorem for a transformer-based neural operator, hence addressing contribution 4. Finally, in Section 6 we explore applications of the transformer neural operator architectures to a variety of operator learning problems, contribution 6.

1.3 Notation

Throughout we denote the positive integers and non-negative integers respectively by $\mathbb{N} = \{1, 2, \cdots\}$ and $\mathbb{Z}^+ = \{0, 1, 2, \cdots\}$, and the notation $\mathbb{R} = (-\infty, \infty)$ and $\mathbb{R}^+ = [0, \infty)$ for the reals and the non-negative reals. We let $[\![1, n]\!] \subset \mathbb{N}$ denote the set $\{1, \ldots, n\}$. For a set D, we denote by $\bar{D}$ the closure of the set, i.e., the union of the set itself and the set of all its limit points. We let $\langle \cdot, \cdot \rangle, |\cdot|$ denote the Euclidean inner-product and norm, noting that $|v|^2 = \langle v, v\rangle$. We may also use $|\cdot|$ to denote the cardinality of a set, but the distinction will be clear from context. We write $\langle \cdot, \cdot \rangle_{\mathbb{R}^d}, |\cdot|_{\mathbb{R}^d}$ if we want to make explicit the dimension of the space on which the Euclidean inner product is computed. We will also denote by $|\cdot|_1$ the norm on the vector space ℓ^1.

We denote by $C(D; \mathbb{R}^d)$ the infinite dimensional Banach space of continuous functions mapping the set D to the d-dimensional vector space $\mathbb{R}^d$. The space is endowed with the supremum norm. Similarly, we denote by $L^2(D; \mathbb{R}^d)$ the infinite dimensional space of square integrable functions mapping D to $\mathbb{R}^d$. The space is endowed with the L^2 inner product, which induces the norm on the space. We will sometimes use the shorthand notation $C(D)$ and $L^2(D)$ to denote continuous functions and square integrable functions defined on the domain D, respectively, when the image space is irrelevant. For integers $s \geq 0$ we denote by C^s the space of continuously differentiable functions up to order s; furthermore, for $p \in [1, \infty)$, we denote by $W^{s,p}$ the Sobolev space of functions possessing weak derivatives up to order s of finite L^p norm. We use the notation H^s to denote $W^{s,2}$. Throughout, general vector spaces will be written using a calligraphic font $\mathcal{U}$. We denote by $\mathcal{L}(\mathcal{U}, \mathcal{V})$ the infinite-dimensional space of linear operators mapping the vector space $\mathcal{U}$ to the vector space $\mathcal{V}$. Throughout, we use different fonts to highlight the difference between operators acting on finite dimensional spaces, such as Q, and operators acting on infinite dimensional spaces, such as Q. We use $\mathcal{F}$ to denote the Fourier transform, while $\mathcal{F}^{-1}$ will denote the inverse Fourier transform.

We use $\mathbb{E}$ to denote expectation under the prevailing probability measure; if we wish to make clear that measure π is the prevailing probability measure then we write $\mathbb{E}_\pi$. We use $\mathcal{N}(m, C)$ to denote a Gaussian with mean m and covariance C; we also use the same notation in the infinite dimensional context do denote a Gaussian measure with covariance operator C; whether we are in the finite dimensional setting or the infinite dimensional one will be clear from context. We also use the notation $\mathsf{unif}(D)$ to denote a uniform distribution defined over the domain D.

2. Continuum Attention

In this section we formulate a continuum analogue of the attention mechanism described in Vaswani et al. (2017). We note that in the transformer architecture of Vaswani et al. (2017), a distinction is made between the self-attention and cross-attention functions. The former, a particular instance of (1) where $u = v$, can be used to produce a representation of the input that captures intra-sequence correlations. On the other hand, cross-attention is used to output a representation of the input u that models cross-correlations between both inputs u and v. Indeed, in Vaswani et al. (2017) cross-attention is employed in the decoder component of the transformer architecture so that the input to the decoder attends to the output of the encoder. For natural language applications in particular, the form of the cross-attention function has allowed for sequences of arbitrary length N to be outputted from a transformer architecture while attending to sequences of different lengths, e.g. as outputs of length M from a self-attention encoder block.

For pedagogical purposes we subdivide the presentation between the self-attention operator and the cross-attention operator, which are treated in Subsection 2.1 and Subsection 2.2 respectively. In Subsection 2.1.1 we define self-attention for discrete sequences indexed on finite bounded sets. Here, we outline the interpretation of self-attention as an expectation under a family of carefully defined discrete probability measures, each acting on the space of sequence indices. We show that under this formulation we recover the standard definition of self-attention from Vaswani et al. (2017). We proceed in Subsection 2.1.2 by formulating self-attention as an operator mapping between spaces of functions defined on bounded uncountable sets. This interpretation is natural, as functions may be thought of as "sequences" indexed over an uncountable domain. As in the discrete case, we formulate the self-attention operator as an expectation under a carefully defined probability measure acting on the domain of the input. We obtain an approximation result given by Theorem 6 that shows that for continuous functions the self-attention mechanism as implemented in practice may be viewed as a Monte Carlo approximation of the self-attention operator described. We mimic these constructions in Subsections 2.2.1 and 2.2.2 for the discrete and continuous formulations of cross-attention, respectively, and obtain an analogous approximation result given by Theorem 12. As in (1), throughout this section $Q \in \mathbb{R}^{d_K \times d_u}, K \in \mathbb{R}^{d_K \times d_v}, V \in \mathbb{R}^{d_V \times d_v}$ are learnable weights that parametrize the attention operators. When we consider self-attention we will have $d_v = d_u$.

2.1 Self-Attention

In the following discussion we define the self-attention operator in the discrete setting and in the continuum setting, in Subsections 2.1.1 and 2.1.2, respectively.

2.1.1 Sequences over $D^N \subset \mathbb{Z}$

We begin by considering the finite, bounded set $D^N = \{1, \cdots, N\}$. We let $u : D^N \to \mathbb{R}^{d_u}$ be a sequence and $j \in D^N$ an index so that $u(j) \in \mathbb{R}^{d_u}$. We will now focus on formulating the attention mechanism in a manner that allows generalization to more general domains D^N; at the end of this subsection we connect our formulation with the more standard one used in the literature, which is specific to the current choice of domain $D^N = \{1, \cdots, N\}$. We define the following discrete probability measure.

Definition 1 *We define a probability measure on D^N, parameterized by (u,j), via the probability mass function $p(\cdot;u,j): D^N \to [0,1]$ defined by*

$$p(k;u,j) = \frac{\exp\Big(\big\langle Qu(j), Ku(k)\big\rangle_{\mathbb{R}^{d_K}}\Big)}{\sum_{\ell \in D^N} \exp\Big(\big\langle Qu(j), Ku(\ell)\big\rangle_{\mathbb{R}^{d_K}}\Big)},$$

for any $k \in D^N$.

We may now define the self-attention operator as an expectation over this measure p of the linear transformation $V \in \mathbb{R}^{d_V \times d_u}$ applied to u.

Definition 2 *We define the self-attention operator A^N as a mapping from a $\mathbb{R}^{d_u}$-valued sequence over D^N, u, into a $\mathbb{R}^{d_V}$-valued sequence over D^N, $\mathsf{A}^N(u)$, that takes the form*

$$\mathsf{A}^N(u)(j) := \mathbb{E}_{k \sim p(k;u,j)}[Vu(k)],$$

for any $j \in D^N$.

To connect these definitions with the standard formulation of transformers, note that sequence u can be reformulated as the matrix $u \in \mathbb{R}^{N \times d_u}$. Then

$$\{uQ^T K u^T\}_{jk} = \big\langle Qu_j, Ku_k\big\rangle_{\mathbb{R}^{d_K}};$$

that is, the $\{j,k\}-$entry of $uQ^TKu^T \in \mathbb{R}^{N\times N}$ is the right-hand side of the preceding identity, where u_ℓ denotes the ℓ'th row of $u \in \mathbb{R}^{N \times d_u}$. It is then apparent that applying the `softmax` function along rows of uQ^TKu^T delivers the vector of probabilities defined in Definition 1, indexed by k; note that j is a parameter in this vector of probabilities, indicating the row of the matrix uQ^TKu^T along which the `softmax` operation is applied. Finally the attention operation from Definition 2 can be rewritten to act between matrices as

$$\mathsf{A}^N(u) = \texttt{attention}(u,u) := \texttt{softmax}(uQ^TKu^T)uV^T,$$

so that $\texttt{attention}(u,u) \in \mathbb{R}^{N \times d_V}$.

Remark 3 (Sequences over $D^N \subset \mathbb{Z}^d$) *Once the attention mechanism is formulated as in Definitions 1, 2, it is straightforward to extend to (generalized) sequences indexed over bounded subsets of $\mathbb{Z}^d$; note, for example, that pixellated images are naturally indexed over bounded subsets of $\mathbb{Z}^2$. Let $D^N \subset \mathbb{Z}^d$ possess cardinality $|D^N| = N$. Let $u: D^N \to \mathbb{R}^{d_u}$ be a sequence and $j \in D^N$ an index. Then the definition of probability on D^N, indexed by (u,j), and the resulting definition of self-attention operator A, is exactly as in Definitions 1, 2. This shows the power of non-standard notation we have employed here.*

2.1.2 Sequences over $D \subset \mathbb{R}^d$

We note that the preceding two formulations of the attention mechanism view it as a transformation defined as an expectation applied to the input sequence. The expectation is with respect to a probability measure supported on an index of the input sequence. Furthermore, the expectation itself depends on the input sequence to which it is applied, and is parameterized by the input sequence index, resulting in a new output sequence of the same length as the input sequence. The formulation using an expectation results from the softmax part of the definition of the attention mechanism; the query and key linear operations define the expectation, and the value linear operation lifts the output sequence to take values in a (possibly) different Euclidean space than the input sequence. These components may be readily extended to work with (generalized) sequences defined over open subsets of $\mathbb{R}^d$. (These would often be referred to as functions, but we call them generalized sequences to emphasize similarity with the discrete case.)

Let $D \subseteq \mathbb{R}^d$ be an open set. Let $u: D \to \mathbb{R}^{d_u}$ be a sequence (function) and $x \in D$ an index.

Definition 4 *We define a probability measure on D, parameterized by (u, x), via the probability density function $p(\cdot; u, x) : D \to \mathbb{R}^+$ defined by*

$$p(y; u, x) = \frac{\exp\Big(\big\langle Qu(x), Ku(y)\big\rangle_{\mathbb{R}^{d_K}}\Big)}{\int_D \exp\Big(\big\langle Qu(x), Ku(s)\big\rangle_{\mathbb{R}^{d_K}}\Big)\,\mathrm{d}s},$$

for any $y \in D$.

Definition 5 *We define the self-attention operator A as a mapping from a $\mathbb{R}^{d_u}$-valued sequence (function) over D, u, into a $\mathbb{R}^{d_V}$-valued sequence (function) over D, $\mathsf{A}(u)$, as follows:*

$$\mathsf{A}(u)(x) = \mathbb{E}_{y\sim p(y;u,x)}[Vu(y)],$$

for any $x \in D$.

A connection between the discrete and continuum formulations of self-attention is captured in the following theorem.

Theorem 6 *The self-attention operator A may be viewed as a mapping $\mathsf{A} : L^\infty(D; \mathbb{R}^{d_u}) \to L^\infty(D; \mathbb{R}^{d_V})$ and thus as a mapping $\mathsf{A} : C(\bar{D}; \mathbb{R}^{d_u}) \to C(\bar{D}; \mathbb{R}^{d_V})$. Furthermore, for any compact set $B \subset C(\bar{D}; \mathbb{R}^{d_u})$,*

$$\lim_{N\to\infty} \sup_{u\in B} \mathbb{E} \left\| \mathsf{A}(u) - \frac{\sum_{j=1}^N \exp\big(\langle Qu(\cdot), Ku(y_j)\rangle\big) Vu(y_j)}{\sum_{\ell=1}^N \exp\big(\langle Qu(\cdot), Ku(y_\ell)\rangle\big)} \right\|_{C(\bar{D};\mathbb{R}^{d_V})} = 0,$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^N \sim \mathsf{unif}(\bar{D})$.

Proof The proof of this result is developed in Appendix A.1. ■

Remark 7 *This result connects our continuum formulation of self-attention with the standard definition, using Monte Carlo approximation of the relevant integrals; a similar result could be proved using finite difference approximation on, for example, a uniform grid. We note that it is not possible to obtain a uniform convergence rate for all $u \in B$ in the norm of $C(\bar{D}; \mathbb{R}^{d_V})$ as the Riemann integral may exhibit arbitrarily slow convergence depending on the regularity of the integrand.*

2.2 Cross-Attention

In the following discussion we define the cross-attention operator in the discrete setting and in the continuum setting, in Subsections 2.2.1 and 2.2.2, respectively.

2.2.1 Sequences over $D^N \subset \mathbb{Z}^d, E^M \subset \mathbb{Z}^e$

We begin by letting $D^N \subseteq \mathbb{Z}^d$ and $E^M \subseteq \mathbb{Z}^e$ be finite sets of points with cardinalities N and M respectively. Let $u : D^N \to \mathbb{R}^{d_u}$ be a sequence (function), $v : E^M \to \mathbb{R}^{d_v}$ another sequence (function), and $j \in D^N$, $k \in \mathbb{E}^M$ indices so that $u(j) \in \mathbb{R}^{d_u}$ and $v(k) \in \mathbb{R}^{d_v}$. We define the following discrete probability measure.

Definition 8 *We define a probability measure on D^N, parameterized by (u, v, j), via the probability mass function $q(\cdot; u, v, j) : D^N \to [0, 1]$ defined by*

$$q(k; u, v, j) = \frac{\exp\Big(\big\langle Qu(j), Kv(k)\big\rangle_{\mathbb{R}^{d_K}}\Big)}{\sum_{\ell\in E^M} \exp\Big(\big\langle Qu(j), Kv(\ell)\big\rangle_{\mathbb{R}^{d_K}}\Big)},$$

for any $k \in E^M$.

We may now define the cross-attention operator as an expectation over this measure q of the linear transformation $V \in \mathbb{R}^{d_V \times d_v}$ applied to v.

Definition 9 *We define the cross-attention operator C^N as a mapping from a $\mathbb{R}^{d_u}$-valued sequence (function) over D^N, u, and a $\mathbb{R}^{d_v}$-valued sequence (function) over E^M, v, into a $\mathbb{R}^{d_V}$-valued sequence (function) over D^N, $\mathsf{C}^N(u,v)$, as follows:*

$$\mathsf{C}^N(u,v)(j) = \mathbb{E}_{k \sim q(k;u,v,j)}[Vv(k)],$$

for any $j \in D^N$.

2.2.2 Sequences over $D \subset \mathbb{R}^d, E \subset \mathbb{R}^e$

The framework described in Subsection 2.2.1 may be readily extended to work with generalized sequences (functions) defined over open subsets of $\mathbb{R}^d$. Indeed, let $D \subseteq \mathbb{R}^d$ and $E \subseteq \mathbb{R}^e$ be open sets. Let $u : D \to \mathbb{R}^{d_u}$ be a sequence (function), $v : E \to \mathbb{R}^{d_v}$ another sequence (function), and $x \in D$, $y \in E$ indices. We define the following probability measure over D before providing the definition of the cross-attention operator.

Definition 10 *We define a probability measure on D, parameterized by (u,v,x), via the probability density function $q(\cdot;u,v,x) : D \to \mathbb{R}^+$ defined by*

$$q(y;u,v,x) = \frac{\exp\Big(\langle Qu(x), Kv(y)\rangle_{\mathbb{R}^{d_K}}\Big)}{\int_E \exp\Big(\langle Qu(x), Kv(s)\rangle_{\mathbb{R}^{d_K}}\Big)\,\mathrm{d}s},$$

for any $y \in E$.

Definition 11 *We define the cross-attention operator C as a mapping from a $\mathbb{R}^{d_u}$-valued sequence (function) over D, u, and a $\mathbb{R}^{d_v}$-valued sequence (function) over E, v, into a $\mathbb{R}^{d_V}$-valued sequence (function) over D, $\mathsf{C}(u,v)$, as follows:*

$$\mathsf{C}(u,v)(x) = \mathbb{E}_{y \sim q(y;u,v,x)}[Vv(y)],$$

for any $x \in D$.

A connection between the discrete and continuum formulations of cross-attention is captured in the following theorem.

Theorem 12 *The cross-attention operator C may be viewed as a mapping $\mathsf{C} : L^\infty(D;\mathbb{R}^{d_u}) \times L^\infty(E;\mathbb{R}^{d_v}) \to L^\infty(D;\mathbb{R}^{d_V})$, and so as a mapping $\mathsf{C} : C(\bar{D};\mathbb{R}^{d_u}) \times C(\bar{E};\mathbb{R}^{d_v}) \to C(\bar{D};\mathbb{R}^{d_V})$. Furthermore, for any compact set $B \subset C(\bar{D};\mathbb{R}^{d_u}) \times C(\bar{E};\mathbb{R}^{d_v})$*

$$\lim_{N\to\infty} \sup_{(u,v)\in B} \mathbb{E} \left\| \mathsf{C}(u,v) - \frac{\sum_{j=1}^N \exp\big(\langle Qu(\cdot), Kv(y_j)\rangle\big) Vv(y_j)}{\sum_{\ell=1}^N \exp\big(\langle Qu(\cdot), Kv(y_\ell)\rangle\big)} \right\|_{C(\bar{D};\mathbb{R}^{d_V})} = 0,$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^N \sim \mathsf{unif}(\bar{E})$.

Proof The proof of this result is developed in Appendix A.2, which contains straightforward generalizations of the techniques in Appendix A.1. ■

3. Continuum Patched Attention

Patches of a function are defined as the restriction of the function itself to elements of a partition of the domain. We may then define patch-attention as a mapping between spaces of functions defined on patch indices. Interest in applying the attention mechanism for computer vision led to the development of methodologies to overcome its prohibitive computational cost. Patching, as employed in vision transformers (ViT), involves subdividing the data space domain into $P \in \mathbb{N}$ "patches" and applying attention to a sequence of flattened patches. This step drastically reduces the complexity of the attention mechanism, which is quadratic in the input sequence length. An analogue of the patching strategy in vision transformers can be considered in the function space setting. Indeed, in image space, attention can be applied across subsets of pixels (patches); in function space, attention can be applied across the function defined on elements of a partition of the domain. Such a generalization to functions defined on the continuum allows for the development of architectures that are mesh-invariant. We describe the patched-attention methodology in the continuum framework developed thus far.

Throughout this section, we let $D \subset \mathbb{R}^d$ be a bounded open set and let $D := D_1 \cup \cdots \cup D_P$ be a uniform partition of the space D so that $D_j \cong D'$ are congruent for all $j \in [\![1, P]\!]$ for some $D' \subset D$, where $P \in \mathbb{N}$ represents the number of patches. We note that we require a uniform partition, namely a partition of subsets of equivalent finite volume, in order to define operators that can be applied to functions defined on each of these subsets; however, for practical application we may have a different number of discretization points within each subset. We consider a function defined on the full domain D, denoted as $u \in \mathcal{U}\big(D; \mathbb{R}^{d_u}\big)$. We define a mapping

$$\widetilde{u} : [\![1, P]\!] \to \mathcal{U}(D', \mathbb{R}^{d_u}), \tag{5}$$

that defines the patched version of the function $u \in \mathcal{U}\big(D; \mathbb{R}^{d_u}\big)$, so that

$$\widetilde{u}(p) = u\big|_{D_p}, \tag{6}$$

for any $p \in [\![1, P]\!]$. We proceed to establish the definitions of the patched self-attention operator A_P in Subsection 3.1 and for completeness the patched cross-attention operator C_P in Subsection 3.2.

3.1 Patched Self-Attention

We begin with the definition of the patched self-attention operator A_P. Indeed, we let the operators $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ be defined such that $\mathsf{Q} : \mathcal{U}(D', \mathbb{R}^{d_u}) \to L^2(D', \mathbb{R}^{d_K}), \mathsf{K} : \mathcal{U}(D', \mathbb{R}^{d_u}) \to L^2(D', \mathbb{R}^{d_K}), \mathsf{V} : \mathcal{U}(D', \mathbb{R}^{d_u}) \to \mathcal{U}(D', \mathbb{R}^{d_V})$ and let $j \in [\![1, P]\!]$ be an index.

Definition 13 *Define a probability measure on $[\![1, P]\!]$, parameterized by $(\widetilde{u}, j)$, via the probability density function $p(\cdot; \widetilde{u}, j) : [\![1, P]\!] \to [0, 1]$ defined by*

$$p(k; \widetilde{u}, j) = \frac{\exp\Big(\big\langle \mathsf{Q}\widetilde{u}(j), \mathsf{K}\widetilde{u}(k)\big\rangle_{L^2(D', \mathbb{R}^{d_K})}\Big)}{\sum_{\ell=1}^{P} \exp\Big(\big\langle \mathsf{Q}\widetilde{u}(j), \mathsf{K}\widetilde{u}(\ell)\big\rangle_{L^2(D', \mathbb{R}^{d_K})}\Big)},$$

for any $k \in [\![1, P]\!]$.[2]

Definition 14 *The self-attention operator A_P maps the function taking a patch index to a $\mathbb{R}^{d_u}$-valued function over D', $\widetilde{u}$, into a function taking a patch index to a $\mathbb{R}^{d_V}$-valued function over D', $\mathsf{A}_\mathsf{P}(\widetilde{u})$, as follows:*

$$\mathsf{A}_\mathsf{P}(\widetilde{u})(j) = \mathbb{E}_{k \sim p(k; \widetilde{u}, j)}[\mathsf{V}\widetilde{u}(k)],$$

for any $j \in [\![1, P]\!]$.

2. We note that as Q is linear, $\mathsf{Q}u(j)$ is shorthand notation for $\mathsf{Q}\big(u(j)\big)$, and similarly for K and V.

Remark 15 (Patched Attention in ViT) *An image may be viewed as a mapping $u : D \to \mathbb{R}^{d_u}$, where $D \subset \mathbb{Z}^2$, and where $d_u = 1$ for grey-scale and $d_u = 3$ for RGB-valued images. In the context of vision transformers (Dosovitskiy et al., 2021), the input to the attention mechanism is a sequence of P "flattened" patches of the input image; we next outline the details to this procedure.*

Letting N be the total number of discretization points in D and P the number of patches, in the discrete setting each function patch $\widetilde{u}(p) : D' \to \mathbb{R}^{d_u}$ may be represented as an $\mathbb{R}^{N/P \times d_u}$ matrix. The flattening procedure of each patch in ViT involves reshaping this matrix into an $\mathbb{R}^{N/P \cdot d_u}$ vector. The full image is thus represented as a sequence of P vectors of dimension $N/P \cdot d_u$. Each patch vector is then linearly lifted to an embedding space of dimension d_{model}. The embedded image, to which attention is applied, may hence be viewed as a matrix $\widetilde{u} \in \mathbb{R}^{P \times d_{model}}$. Therefore, ViT may be cast in the setting of the discrete interpretation of the self-attention operator outlined in Subsection 2.1.1, where the domain on which attention is applied is D^P. A key observation to the ensuing discussion is that the application of a linear transformation to a "flattened" patch in ViT breaks the mesh-invariance of the architecture; however, this operation may be viewed as a nonlocal transformation on each patch. We leverage this insight in Section 4 to design discretization invariant transformer neural operators that employ patching. We further note that in practice, patching in ViT is often also achieved via strided convolutions, with stride equivalent to the kernel size; however, because the kernel size determines the number of parameters, such a procedure also breaks discretization invariance as it leads to architectures with parameters dependent on resolution of the input.

Remark 16 (Codomain Attention) *The work of Rahman et al. (2024) employs an attention mechanism that acts across the channels of the function $u \in \mathcal{U}\big(D; \mathbb{R}^{d_{model}}\big)$ in the latent space of a neural operator architecture. Indeed the "codomain" attention mechanism defined may be viewed as a particular variant of Definition 14, where $D' \cong D$, where $\mathsf{A}_\mathsf{P}(\widetilde{u})(j)$ is defined for $j \in [\![1, d_{model}]\!]$ and where $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ are implemented as integral operators. In particular the indexing is by channels and not by patch. For problems in computer vision, similar ideas have been explored with discrete versions of the $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ operators (Chen et al., 2017).*

3.2 Patched Cross-Attention

For completeness, we outline the definition of the patched cross-attention operator. We consider $E \subset \mathbb{R}^e$ to be a bounded open set and $E := E_1 \cup \ldots \cup E_O$ a uniform partition of E so that $E_j \cong E' \cong D'$ for all $j \in [\![1, O]\!]$ for some $E' \subset E$ and $O \in \mathbb{N}$. We note that for the following construction we require $E' \cong D'$, but it may hold in general that $P \neq O$. As done for $\widetilde{u}$ we define an operator

$$\widetilde{v} : [\![1, O]\!] \to \mathcal{U}(E', \mathbb{R}^{d_v}), \tag{7}$$

that defines the patched version of the function $v \in \mathcal{U}\big(E; \mathbb{R}^{d_v}\big)$, so that

$$\widetilde{v}(o) = v\big|_{E_o}, \tag{8}$$

for any $o \in [\![1, O]\!]$. Furthermore, we let the operators $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ be defined such that $\mathsf{Q} : \mathcal{U}(D', \mathbb{R}^{d_u}) \to L^2(D', \mathbb{R}^{d_K}), \mathsf{K} : \mathcal{U}(E', \mathbb{R}^{d_v}) \to L^2(E', \mathbb{R}^{d_K}), \mathsf{V} : \mathcal{U}(E', \mathbb{R}^{d_v}) \to \mathcal{U}(E', \mathbb{R}^{d_V})$ and let $j \in [\![1, P]\!]$ be an index.

Definition 17 *We define a probability measure on D, parameterized by $(\tilde{u}, \tilde{v}, j)$, via the probability density function $q(\cdot; \widetilde{u}, \widetilde{v}, j) : [\![1, P]\!] \to [0, 1]$ defined by*

$$q(k; \widetilde{u}, \widetilde{v}, j) = \frac{\exp\Big(\big\langle \mathsf{Q}\widetilde{u}(j), \mathsf{K}\widetilde{v}(k)\big\rangle_{L^2(E', \mathbb{R}^{d_K})}\Big)}{\sum_{\ell=1}^{O} \exp\Big(\big\langle \mathsf{Q}\widetilde{u}(j), \mathsf{K}\widetilde{v}(\ell)\big\rangle_{L^2(E', \mathbb{R}^{d_K})}\Big)},$$

for any $k \in [\![1, O]\!]$.

Definition 18 *The cross-attention operator* C *maps the operator taking a patch index to a* $\mathbb{R}^{d_u}$*-valued function over* D'*,* $\widetilde{u}$*, and the operator taking a patch index to a* $\mathbb{R}^{d_v}$*-valued function over* E'*,* $\widetilde{v}$*, into an operator taking a patch index to a* $\mathbb{R}^{d_\mathsf{V}}$*-valued function over* D'*,* $\mathsf{C}_\mathsf{P}(\widetilde{u},\widetilde{v})$*, as follows:*

$$\mathsf{C}_\mathsf{P}(\widetilde{u},\widetilde{v})(j) = \mathbb{E}_{k\sim q(k;\widetilde{u},\widetilde{v},j)}[\mathsf{V}\widetilde{v}(k)],$$

for any $j \in [\![1, P]\!]$*.*

4. Transformer Neural Operators

In this section we turn our focus to describing how to devise transformer-based neural operators using the continuum attention operator A as defined in the function space setting in Section 2, and the continuum patched-attention operator A_P acting on function space from Section 3. We make use of the operator frameworks developed in Sections 2 and 3 to formulate transformer neural operator architectures acting on function space. We start in Subsection 4.1, setting-up the framework. In Subsection 4.2 we formulate an analogue of the transformer from Vaswani et al. (2017) as an operator mapping an input function space $\mathcal{U}\big(D;\mathbb{R}^{d_u}\big)$ to the solution function space $\mathcal{Z}\big(D;\mathbb{R}^{d_z}\big)$. The resulting scheme, which we define as transformer neural operator (TNO) is invariant to the discretization of the input function and allows for zero-shot generalization to irregular, non-uniform meshes. However, this architecture exhibits quadratic complexity with respect to the number of discretization points of the input functions. Hence, motivated by the need to develop new efficient architectures to employ attention on longer sequences, or in our case higher resolution discretizations of functions, in Subsection 4.3 we describe a generalization of the ViT methodology to the function space setting, inspired by the work in Dosovitskiy et al. (2021). In particular we generalize the procedure of lifting flattened patches to the embedding dimension, as done in ViT, to the function space setting. This is achieved by lifting patches of functions to an embedding dimension using a nonlocal linear operator, namely, an integral operator. Unlike ViT, the resulting neural operator architecture, which we call ViT neural operator (ViTNO), is discretization invariant and allows for zero-shot generalization to different resolutions. In Subsection 4.4 we describe a different patching-based transformer neural operator acting on function space, the Fourier attention neural operator (FANO). The architecture is discretization invariant and is based on $\mathsf{Q},\mathsf{K},\mathsf{V}$, learnable parameters in the attention mechanism, being defined as integral operators acting themselves on function space. We demonstrate in Section 6 that the approach involving parametrizing $\mathsf{Q},\mathsf{K},\mathsf{V}$ as integral operators yields an architecture with greater parameter count for a similar computational complexity. We show that this architecture produces improved results in the context of a problem with smooth inputs.

4.1 Set-Up

In the subsections that follow we describe transformer neural operators of the form $\mathsf{G} : \mathcal{U}(D;\mathbb{R}^{d_u}) \times \Theta \to \mathcal{Z}(D;\mathbb{R}^{d_z})$, which serve as approximations to an operator $\mathsf{G}^\dagger : \mathcal{U}(D;\mathbb{R}^{d_u}) \to \mathcal{Z}(D;\mathbb{R}^{d_z})$. The neural operators we devise have the general form

$$\mathsf{G}(u,\theta) := \Big(\mathsf{T}_{\text{out}} \circ \mathsf{E}_L \circ \mathsf{T}_{\text{in}}\Big)(u,\theta), \tag{9}$$

for any $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$ and $\theta \in \Theta$. In (9), the operators $\mathsf{T}_{\text{in}} : \mathcal{U}(D;\mathbb{R}^{d_u}) \times \Theta \to \mathcal{V}$ and $\mathsf{T}_{\text{out}} : \mathcal{V} \times \Theta \to \mathcal{Z}(D;\mathbb{R}^{d_z})$ are defined for some appropriate embedding function space $\mathcal{V}$. We will make explicit the definitions of the general operators T_{in}, T_{out} and the embedding function space $\mathcal{V}$ in the context of each architecture. Throughout this section, we drop explicit dependencies on θ for brevity unless we wish to stress a parametric dependence; for example, we may abuse notation and write $\mathsf{T}_{\text{in}}(u) := \mathsf{T}_{\text{in}}(u;\ \theta)$. The operator $\mathsf{E}_L : \mathcal{V} \times \Theta \to \mathcal{V}$ defines the neural operator analogue of the transformer encoder block from Vaswani et al. (2017). Its action on the input $v^{(0)} := \mathsf{T}_{\text{in}}(u) \in \mathcal{V}$ is summarized by the iteration

$$v^{(l-1)} \hookleftarrow \mathsf{W}_1 v^{(l-1)} + \mathsf{A}_{\text{MultiHead}}\big(v^{(l-1)}\big), \tag{10a}$$

$$v^{(l-1)} \hookleftarrow \mathsf{F}_{\text{LayerNorm}}\big(v^{(l-1)}\big), \tag{10b}$$

$$v^{(l-1)} \hookleftarrow \mathsf{W}_2 v^{(l-1)} + \mathsf{F}_{\text{NN}}\big(v^{(l-1)}\big), \tag{10c}$$

$$v^{(l)} \hookleftarrow \mathsf{F}_{\text{LayerNorm}}\big(v^{(l-1)}\big), \tag{10d}$$

for $l = 1, \dots, L$ layers, so that $\mathsf{E}_L\big(v^{(0)}\big) = v^{(L)} \in \mathcal{V}$ [3]. We note that for notational convenience, we have suppressed dependence of the operators on the layer ℓ; indeed, each of the operators $\mathsf{A}_{\text{MultiHead}}$, $\mathsf{W}_1, \mathsf{W}_2$, F_{NN} and $\mathsf{F}_{\text{LayerNorm}}$, will have different learnable parametrizations for each encoder layer ℓ. Before we delve into the specific neural operator architectures, we summarize the definitions of the elements of the transformer encoder described in (10), which are common to all the subsequent architectures. The multi-head self-attention operator $\mathsf{A}_{\text{MultiHead}} : \mathcal{V} \to \mathcal{V}$ is defined as the composition of a linear transformation $W_{\text{MultiHead}}$ applied pointwise to its input and a concatenation of the outputs of $H \in \mathbb{N}$ self-attention operators so that

$$\Big(\mathsf{A}_{\text{MultiHead}}(v)\Big)(x) = W_{\text{MultiHead}}\Big(\mathsf{A}_{\bullet}(v;\theta_1)(x), \dots, \mathsf{A}_{\bullet}(v;\theta_H)(x)\Big), \tag{11}$$

for any $v \in \mathcal{V}$. Here the notation $\mathsf{A}_{\bullet}$ indicates the ambiguity in the definition of self-attention. We note that we will make explicit which definition of self-attention operator from the previous sections we employ for each architecture; we will also make clear the definitions of the parameters $\theta_h \in \Theta$ defining each attention operator. The operators $\mathsf{W}_1, \mathsf{W}_2 : \mathcal{V} \to \mathcal{V}$ are pointwise linear operators.[4] Next, the operator $\mathsf{F}_{\text{LayerNorm}} : \mathcal{V} \to \mathcal{V}$ defines the layer normalization and the operator $\mathsf{F}_{\text{NN}} : \mathcal{V} \to \mathcal{V}$ defines a feed-forward neural network layer. Both operators are applied pointwise to the input; further details are provided in the context of each architecture.

A key aspect of the ensuing discussion will be the distinction between local and nonlocal linear operators. We note that for an arbitrary function space $\mathcal{U}\big(D;\mathbb{R}^r\big)$ where $D \subset \mathbb{R}^d$, the space of linear operators $\mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$ includes both local and nonlocal linear operators. In the following, we will consider pointwise linear operators in $\mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$ that admit a finite-dimensional representation $W \in \mathbb{R}^{r' \times r}$. On the other hand, we will also consider linear integral operators in $\mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$, a class of nonlocal operators. Namely, we consider operators W that are integral operators learned from data of the form $\mathsf{W} : \Theta_{\mathsf{W}} \to \mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$.

Definition 19 (Integral Operator W) *The integral operator* $\mathsf{W} : \Theta_{\mathsf{W}} \to \mathcal{L}\Big(\mathcal{U}\big(D;\mathbb{R}^r\big);\mathcal{U}\big(D;\mathbb{R}^{r'}\big)\Big)$ *is defined as the mapping given by*

$$\big(\mathsf{W}(\theta)u\big)(x) := \int_D \kappa_\theta(x,y)u(y)\mathrm{d}y, \qquad \forall u \in \mathcal{U}\big(D;\mathbb{R}^r\big),\ \forall x \in D, \tag{12}$$

where $\kappa_\theta : D \to \mathbb{R}^{r' \times r}$ *is a neural network parametrized by* $\theta \in \Theta_{\mathsf{W}}$.

Assuming $\kappa_\theta(x,y) = \kappa_\theta(x-y)$, hence the kernel to be periodic, then (12) may be viewed as a convolution operator, which makes it possible to use the Fast Fourier Transform (FFT) to compute (12) and to parametrize W in Fourier space. Following Li et al. (2021), this insight leads to the following formulation of the operator given in Definition 19.

3. We note that we use v to denote an arbitrary function in the space $\mathcal{V}$. This is not to be confused with v used for cross-attention in Section 2. We also denote the output of the ℓ'th encoder layer by $v^{(\ell)}$, not to be confused with the notation used for the data pairs.
4. We note that for all experiments in Section 6 we set $\mathsf{W}_1, \mathsf{W}_2$ to be the identity operators.

Definition 20 (Fourier Integral Operator W) *We define the Fourier integral operator as*

$$\big(\mathsf{W}(\theta)u\big)(x) = \mathcal{F}^{-1}\Big(R_{\mathsf{W}}(\theta)\cdot(\mathcal{F}u)\Big)(x) \qquad \forall u \in \mathcal{U}\big(D;\mathbb{R}^{r}\big),\, \forall x \in D. \tag{13}$$

To link to the definition (12) *with translation-invariant kernel we take $R_{\mathsf{W}}(\theta)$ to be the Fourier transform of the periodic function $\kappa_\theta : D \to \mathbb{R}^{r'\times r}$ parametrized by $\theta \in \Theta_{\mathsf{W}}$.*

For uniform discretizations $n_1 \times \cdots \times n_d = N$ of the space D, the Fourier transform can be replaced with the Fast Fourier Transform (FFT). Indeed, for the Fourier transform we have $\mathcal{F}u : \mathbb{R}^d \to \mathbb{C}^r$ and $R_{\mathsf{W}}(\theta) : \mathbb{R}^d \to \mathbb{C}^{r'\times r}$. On the other hand, for the FFT we have $\mathcal{F}u : \mathbb{Z}^d \to \mathbb{C}^r$ and $R_{\mathsf{W}}(\theta) : \mathbb{Z}^d \to \mathbb{C}^{r'\times r}$. In practice, it is possible to select a finite-dimensional parametrization by choosing a finite set of wave numbers

$$\big|Z_{k_{\max}}\big| := \Big|\big\{k \in \mathbb{Z}^d : |k_j| \le k_{\max,j},\ \text{for}\ j = 1,\dots,d\big\}\Big|.$$

We can therefore parametrize R_{W} by a complex $\big((2k_{\max,1}+1)\times\ldots\times(2k_{\max,d}+1)\times r'\times r\big)$-tensor. We will use the notation $k_{\max} = \big|Z_{k_{\max}}\big|$ when discussing parameter scalings in the subsequent sections.

These integral operator definitions will be used to define architectures for the Vision Transformer Neural Operator in Subsection 4.3 and the Fourier Attention Neural Operator in Subsection 4.4, but are not needed to define the basic Transformer Neural Operator in Subsection 4.2. In the context of each architecture, we make explicit the parametrizations of the integral operators employed.

4.2 Transformer Neural Operator

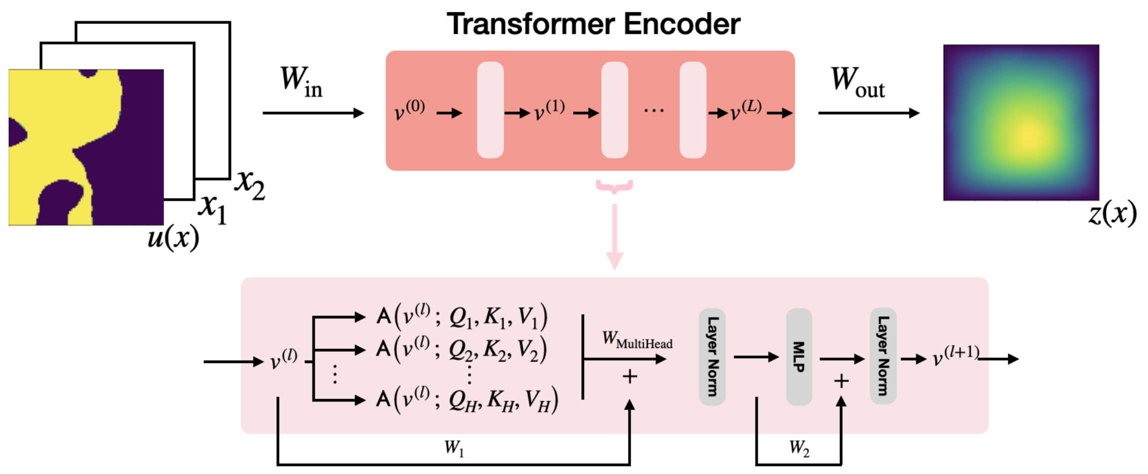

Figure 1: Transformer Neural Operator.

The transformer neural operator architecture follows the general structure of Equations (9) and (10); here, we make explicit the choices for $\mathsf{T}_{\text{in}}, \mathsf{T}_{\text{out}}, \mathsf{E}_L$ that instantiate it. In Figure 1 we provide a schematic representation of the full transformer neural operator architecture. We consider model inputs $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$, model embedding space $\mathcal{V} := \mathcal{U}(D;\mathbb{R}^{d_{\text{model}}})$, and model output space $\mathcal{Z}(D;\mathbb{R}^{d_z})$, with $D \subset \mathbb{R}^d$. The function $u \in \mathcal{U}(D;\mathbb{R}^{d_u})$ is first concatenated with the identity defined on the domain D, so that the resulting function $u_{\text{in}} \in \mathcal{U}(D;\mathbb{R}^{d_u+d})$ is defined by its action on $x \in D$ as

$$u_{\text{in}}(x) := \big(u(x), x\big). \tag{14}$$

In practice, this step defines a positional encoding that is appended to the input function. A linear transformation $W_{\text{in}} \in \mathbb{R}^{d_{\text{model}} \times (d_u + d)}$ is then applied pointwise to lift the function u_{in} into the embedding space $\mathcal{V}$. The input operator $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^{d_u}) \to \mathcal{V}$ is hence defined as

$$\Big(\mathsf{T}_{\text{in}}(u)\Big)(x) := W_{\text{in}} u_{\text{in}}(x) \tag{15}$$

for $u_{\text{in}} \in \mathcal{U}\big(D; \mathbb{R}^{d_u + d}\big)$ defined as in (14), where $x \in D$, $u(x) \in \mathbb{R}^{d_u}$, and $W_{\text{in}} \in \mathbb{R}^{d_{\text{model}} \times (d_u + d)}$. Observe that T_{in} acts as a local, pointwise linear operator on u_{in}.

Next, we define the details of $\mathsf{E}_L : \mathcal{V} \to \mathcal{V}$ in (10) by specifying a definition of $\mathsf{A}_{\text{MultiHead}}, \mathsf{W}_1, \mathsf{W}_2$, $\mathsf{F}_{\text{LayerNorm}}$, and F_{NN} for the transformer neural operator setting. The operator $\mathsf{A}_{\text{MultiHead}} : \mathcal{V} \to \mathcal{V}$ is the multi-head attention operator from (11) based on the continuum self-attention from Definition 5. Letting $H \in \mathbb{N}$ denote the number of attention heads, the multi-head attention operator is parametrized by the learnable linear transformations $Q_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, K_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, V_h \in \mathbb{R}^{d_K \times d_{\text{model}}}$, for $h = 1, \dots, H$, where for implementation purposes d_K is chosen as $d_K := d_{\text{model}}/H$.[5] The multi-head attention operator is thus defined by the application of a linear transformation $W_{\text{MultiHead}} \in \mathbb{R}^{d_{\text{model}} \times H \cdot d_K}$ to the concatenation of the outputs of $H \in \mathbb{N}$ self-attention operators. The operators $\mathsf{W}_1, \mathsf{W}_2 : \mathcal{V} \to \mathcal{V}$ are pointwise linear operators and hence admit finite-dimensional representations $W_1, W_2 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, so that they define a map

$$v(x) \mapsto W_1 v(x),$$

for each $x \in D$, and similarly for W_2. The operator $\mathsf{F}_{\text{LayerNorm}} : \mathcal{V} \to \mathcal{V}$ is defined such that

$$\Big(\mathsf{F}_{\text{LayerNorm}}(v; \gamma, \beta)(x)\Big)_k = \gamma_k \cdot \frac{\big(v(x)\big)_k - m\big(v(x)\big)}{\sqrt{\sigma^2\big(v(x)\big) + \epsilon}} + \beta_k, \tag{16}$$

for $k = 1, \dots, d_{\text{model}}$, any $v \in \mathcal{V}$, and any $x \in D \subset \mathbb{R}^d$, where the notation $(\,\cdot\,)_k$ is used to denote the k'th entry of the vector. In equation (16), $\epsilon \in \mathbb{R}^+$ is a fixed parameter, $\gamma_k, \beta_k \in \mathbb{R}$ are learnable parameters and m, σ are defined as

$$\begin{aligned} m(y) &:= \frac{1}{d_{\text{model}}} \sum_{k=1}^{d_{\text{model}}} y_k, \\ \sigma^2(y) &:= \frac{1}{d_{\text{model}}} \sum_{k=1}^{d_{\text{model}}} \big(y_k - m(y)\big)^2, \end{aligned} \tag{17}$$

for any $y \in \mathbb{R}^{d_{\text{model}}}$. The operator $\mathsf{F}_{\text{NN}} : \mathcal{V} \to \mathcal{V}$ is defined such that

$$\mathsf{F}_{\text{NN}}(v; W_3, W_4, b_1, b_2)(x) = W_3 f\big(W_4 v(x) + b_1\big) + b_2, \tag{18}$$

for any $v \in \mathcal{V}$ and $x \in D$, where $W_3, W_4 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ and $b_1, b_2 \in \mathbb{R}^{d_{\text{model}}}$ are learnable parameters and where f is a nonlinear activation function.

Finally, we define the output operator $\mathsf{T}_{\text{out}} : \mathcal{V} \to \mathcal{Z}(D; \mathbb{R}^{d_z})$ as a local linear operator applied pointwise, so that

$$\Big(\mathsf{T}_{\text{out}}(v)\Big)(x) = W_{\text{out}} v(x), \tag{19}$$

for $x \in D$, $v \in \mathcal{V}$ and $W_{\text{out}} \in \mathbb{R}^{d_z \times d_{\text{model}}}$.

The composition of the operators $\mathsf{T}_{\text{in}}, \mathsf{E}_L, \mathsf{T}_{\text{out}}$ completes the definition of the transformer neural operator. In Appendix B.1 we outline how this neural operator architecture is implemented in the finite-dimensional setting, where the input function $u \in \mathcal{U}\big(D; \mathbb{R}^{d_u}\big)$ is defined on a finite set of discretization points $\{x_i\}_{i=1}^N \subset D$.

5. We note that d_{model} and H should be chosen so that d_{model} is a multiple of H.

4.3 Vision Transformer Neural Operator

We introduce an analogue of the vision transformer architecture as a neural operator using the patching in the continuum framework developed in Section 3. We note that in the context of the ViT from Dosovitskiy et al. (2021), performing patch "flattening" before a local lifting to the embedding dimension introduces a nonlocal transformation on the patches, as described in Remark 15. This procedure as implemented in Dosovitskiy et al. (2021) violates the desirable (for neural operators) discretization invariance property. Hence, for a neural operator analogue, we substitute the lifting to embedding dimension with a nonlocal integral operator. The ViT neural operator (ViT NO) introduced is thus discretization invariant. We describe this methodology within the framework provided by making explicit choices for $\mathsf{T}_{\text{in}}, \mathsf{T}_{\text{out}}, \mathsf{E}_L$ that instantiate Equations (9) and (10) appropriately. In Figure 2 we provide a schematic for the architecture, with a graphical representation of the encoder layer in Figure 3.

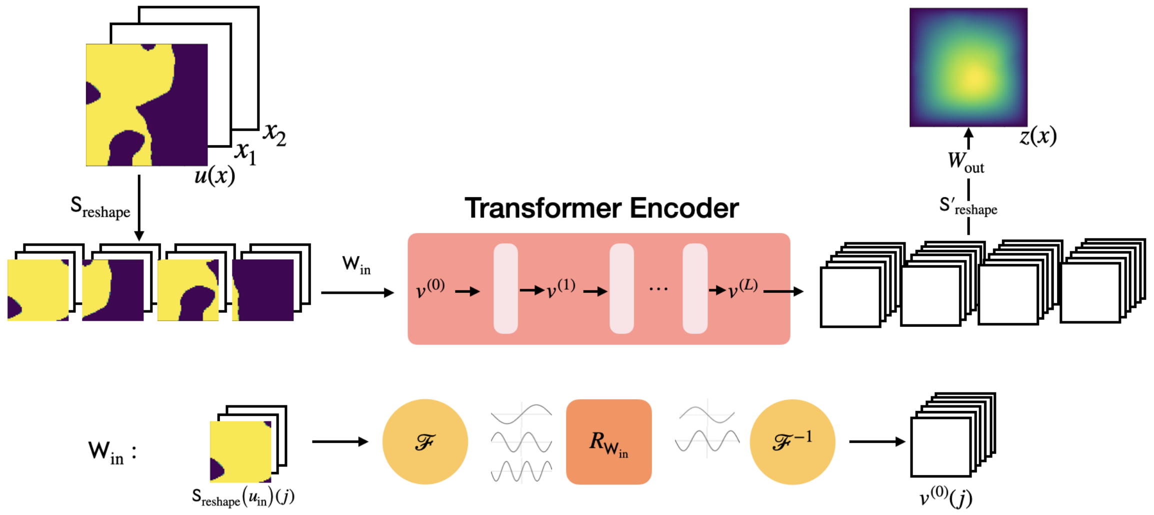
x_2
x_1
$u(x)$
$\mathsf{S}_{\text{reshape}}$
Transformer Encoder
W_{in}
$v^{(0)}$
$v^{(1)}$
$\cdots$
$v^{(L)}$
$z(x)$
W_{out}
$\mathsf{S}'_{\text{reshape}}$
W_{in} :
$\mathsf{S}_{\text{reshape}}(u_{\text{in}})(j)$
$\mathcal{F}$
$R_{\mathsf{W}_{\text{in}}}$
$\mathcal{F}^{-1}$
$v^{(0)}(j)$

Figure 2: Vision Transformer Neural Operator.

Let $D \subset \mathbb{R}^d$ be a bounded open set. Consider $u \in \mathcal{U}(D; \mathbb{R}^{d_u})$ and let $D := D_1 \cup \cdots \cup D_P$ be a uniform partition of the space D so that $D_j \cong D'$ have equivalent finite volume for all $j \in [\![1, P]\!]$ for some $D' \subset D$, where $P \in \mathbb{N}$ represents the number of patches. We consider model inputs $u \in \mathcal{U}(D; \mathbb{R}^{d_u})$, model embedding space $\mathcal{V} := \{v \mid v : [\![1, P]\!] \to \mathcal{U}(D'; \mathbb{R}^{d_{\text{model}}})\}$, which is a space of functions acting on patch indices, and model output space $\mathcal{Z}(D; \mathbb{R}^{d_z})$, with domain $D \subset \mathbb{R}^d$.

We begin by defining the input function $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^{d_u}) \to \mathcal{V}$. The function $u \in \mathcal{U}(D; \mathbb{R}^{d_u})$ is first concatenated with the identity defined on the domain D, so that the resulting function $u_{\text{in}} \in \mathcal{U}(D; \mathbb{R}^{d_u+d})$ is defined as in (14); recall that this defines a positional encoding. An operator $\mathsf{S}_{\text{reshape}}$ is applied to the function $u_{\text{in}} \in \mathcal{U}(D; \mathbb{R}^{d_u+d})$ that acts on the input so that the resulting function is of the form

$$\mathsf{S}_{\text{reshape}}(u_{\text{in}}) : [\![1, P]\!] \to \mathcal{U}(D', \mathbb{R}^{d_u+d}). \tag{20}$$

We apply a nonlocal linear operator $\mathsf{W}_{\text{in}} \in \mathcal{L}\Big(\mathcal{U}(D'; \mathbb{R}^{d_u+d}); \mathcal{U}\big(D'; \mathbb{R}^{d_{\text{model}}}\big)\Big)$ that acts on any $u : [\![1, P]\!] \to \mathcal{U}(D'; \mathbb{R}^{d_u+d})$ so that for any $j \in [\![1, P]\!]$

$$u(j) \mapsto \mathsf{W}_{\text{in}} u(j). \tag{21}$$

The above nonlocal linear operator is chosen as an integral operator as in Definition 20 and the subsequent discussion. The composition of the operators defined in Equations (14), (20) and (21) defines our choice of T_{in}. Namely, we define $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^{d_u}) \to \mathcal{V}$ via

$$\Big(\mathsf{T}_{\text{in}}(u)\Big)(j) \coloneqq \mathsf{W}_{\text{in}}\Big(\mathsf{S}_{\text{reshape}}\big(u_{\text{in}}\big)(j)\Big), \tag{22}$$

where $u_{\text{in}} \in \mathcal{U}(D; \mathbb{R}^{d_u+d})$ is defined as in (14), $j \in [\![1, P]\!]$ and W_{in} is defined as in (21).

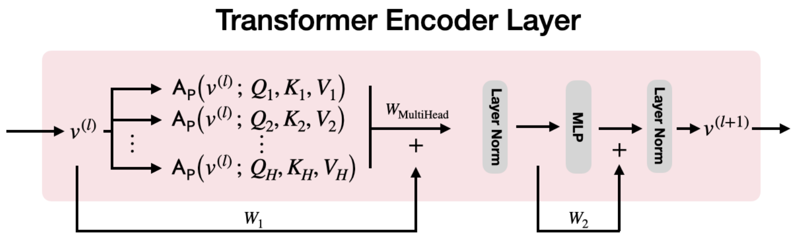
Transformer Encoder Layer
$v^{(l)}$
$\mathsf{A}_\mathsf{P}(v^{(l)}; Q_1, K_1, V_1)$
$\mathsf{A}_\mathsf{P}(v^{(l)}; Q_2, K_2, V_2)$
$\mathsf{A}_\mathsf{P}(v^{(l)}; Q_H, K_H, V_H)$
$W_{\text{MultiHead}}$
+
Layer Norm
MLP
+
Layer Norm
$v^{(l+1)}$
W_1
W_2

Figure 3: Vision Transformer Neural Operator Encoder Layer.

Next, we define the details of $\mathsf{E}_L : \mathcal{V} \to \mathcal{V}$ in Equation (10) by specifying a definition of $\mathsf{A}_{\text{MultiHead}}$, $\mathsf{F}_{\text{LayerNorm}}$, $\mathsf{W}_1, \mathsf{W}_2$, and F_{NN} in this patched setting. In each layer the operator $\mathsf{A}_{\text{MultiHead}} : \mathcal{V} \to \mathcal{V}$ is the multi-head attention operator from (11) based on the self-attention from Definition 14. Letting $H \in \mathbb{N}$ denote the number of attention heads, the multi-head attention operator is parametrized by learnable local (pointwise) linear operators $\mathsf{Q}_h, \mathsf{K}_h, \mathsf{V}_h$ for $h = 1, \dots, H$. In this setting, these are chosen to be linear operators applied pointwise, they can be represented by finite dimensional linear transformations $Q_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, K_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, V_h \in \mathbb{R}^{d_K \times d_{\text{model}}}$, for $h = 1, \dots, H$. [6] For implementation purposes d_K is chosen as $d_K \coloneqq d_{\text{model}}/H$. [7] The multi-head attention operator is thus defined by the application of a local linear transformation $W_{\text{MultiHead}} \in \mathbb{R}^{d_{\text{model}} \times H \cdot d_K}$ to the concatenation of the outputs of $H \in \mathbb{N}$ self-attention operators. The operators $\mathsf{W}_1, \mathsf{W}_2 : \mathcal{V} \to \mathcal{V}$ are pointwise linear operators and hence admit finite-dimensional representations $W_1, W_2 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, so that they define a map

$$v(x; p) \mapsto W_1 v(x; p), \tag{23}$$

for each $x \in D'$ and $p \in [\![1, P]\!]$, and similarly for W_2. The operator $\mathsf{F}_{\text{LayerNorm}} : \mathcal{V} \to \mathcal{V}$ is defined such that

$$\Big(\mathsf{F}_{\text{LayerNorm}}(v; \gamma, \beta)(x; p)\Big)_k = \gamma_k \cdot \frac{\big(v(x;p)\big)_k - m\big(v(x;p)\big)}{\sqrt{\sigma^2\big(v(x;p)\big) + \epsilon}} + \beta_k, \tag{24}$$

for $k = 1, \dots, d_{\text{model}}$, any $p \in [\![1, P]\!]$ and any $x \in \mathbb{R}^d \subset D$, where we use the notation $(\,\cdot\,)_k$ to denote the k'th entry of the vector. In equation (24), $\epsilon \in \mathbb{R}^+$ is a fixed parameter, $\gamma_k, \beta_k \in \mathbb{R}$ are learnable parameters and m, σ are defined as in (17). The operator $\mathsf{F}_{\text{NN}} : \mathcal{V} \to \mathcal{V}$ is defined by

$$\mathsf{F}_{\text{NN}}(v; W_3, W_4, b_1, b_2)(x; p) = W_3 f\big(W_4 v(x; p) + b_1\big) + b_2, \tag{25}$$

for any $x \in D$ and any $p \in [\![1, P]\!]$, where $W_3, W_4 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ and $b_1, b_2 \in \mathbb{R}^{d_{\text{model}}}$ are learnable parameters and f is a nonlinear activation function.

Finally, we define the action of $\mathsf{T}_{\text{out}} : \mathcal{V} \to \mathcal{Z}(D; \mathbb{R}^{d_z})$. This is given by first applying a reshaping operator $\mathsf{S}'_{\text{reshape}}$ to the output of the encoder, where this map is defined as

$$v \mapsto \mathsf{S}'_{\text{reshape}}\big(v\big) \in \mathcal{U}(D; \mathbb{R}^{d_{\text{model}}}). \tag{26}$$

6. We note that in the context of the Fourier attention neural operator, presented in the next Subsection 4.4, we generalize this definition to $\mathsf{Q}_h, \mathsf{K}_h, \mathsf{V}_h$ for $h = 1, \dots, H$ being nonlocal linear integral operators.
7. We note that d_{model} and H should be chosen so that d_{model} is a multiple of H.

We note that the operator $\mathsf{S}'_{\text{reshape}}$ may be viewed as an inverse of $\mathsf{S}_{\text{reshape}}$. A pointwise linear transformation $W_{\text{out}} \in \mathbb{R}^{d_z \times d_{\text{model}}}$ defined by

$$v(x) \mapsto W_{\text{out}} v(x) \in \mathcal{Z}(D; \mathbb{R}^{d_z}), \tag{27}$$

for any $x \in D$, is then applied. This procedure completes the definition of the operator $\mathsf{T}_{\text{out}} : \mathcal{V} \to \mathcal{Z}(D; \mathbb{R}^{d_z})$, which is given by

$$\Big(\mathsf{T}_{\text{out}}(v)\Big)(x) := W_{\text{out}}\Big(\mathsf{S}'_{\text{reshape}}(v)(x)\Big), \tag{28}$$

for any $v \in \mathcal{V}$ and $x \in D$.

Remark 21 *Discontinuities may arise when the patching architecture is used and may be visible in error plots and can also interfere with self-composition of maps learned as solution operators of time-dependent PDEs. Ameliorating such discontinuities can be achieved using a smoothing operator as a final layer of the network. To this end, assuming $\mathcal{Z}\big(D; \mathbb{R}^{d_z}\big) \subset H^s\big(D; \mathbb{R}^{d_z}\big)$ for some $s \geq 0$, we define for $\epsilon > 0$ and $\alpha > 1$ the operator $(I - \epsilon\Delta)^{-\alpha} : \mathcal{Z}\big(D; \mathbb{R}^{d_z}\big) \to H^{s+2\alpha}\big(D; \mathbb{R}^{d_z}\big)$, and define the last layer of the architecture as*

$$z \mapsto (I - \epsilon\Delta)^{-\alpha}(z), \tag{29}$$

for $z := \Big(\mathsf{T}_{\text{out}} \circ \mathsf{E}_L \circ \mathsf{T}_{\text{in}}\Big)(u)$.

The composition of the operators $\mathsf{T}_{\text{in}}, \mathsf{E}_L, \mathsf{T}_{\text{out}}$ completes the definition of the ViT neural operator. In Appendix B.2 we outline how this neural operator architecture is implemented in the finite-dimensional setting, where the input function $u \in \mathcal{U}\big(D; \mathbb{R}^{d_u}\big)$ is defined on a finite set of discretization points $\{x_i\}_{i=1}^N \subset D$.

4.4 Fourier Attention Neural Operator

In this subsection, we introduce a different transformer neural operator based on patching, the Fourier attention neural operator (FANO). In this setting, to gain additional expressivity, we replace the local linear operators $\mathsf{Q}, \mathsf{K}, \mathsf{V}$ in the attention mechanism with nonlocal integral operators. In Figure 4 we provide a schematic for the Fourier attention neural operator architecture, with a graphical representation of the encoder layer in Figure 5. We outline the details of the architecture in the following discussion.

We again let $D \subseteq \mathbb{R}^d$ be a bounded open set. Consider $u \in \mathcal{U}(D; \mathbb{R}^{d_u})$ and let $D := D_1 \cup \ldots \cup D_P$ be a uniform partition of the space D so that $D_j \cong D'$ for all $j \in [\![1, P]\!]$ for some $D' \subset D$, where $P \in \mathbb{N}$ represents the number of patches. We consider model inputs $u \in \mathcal{U}(D; \mathbb{R}^{d_u})$, model embedding space $\mathcal{V} := \big\{v \mid v : [\![1, P]\!] \to \mathcal{U}\big(D'; \mathbb{R}^{d_{\text{model}}}\big)\big\}$ which is a space of functions acting on patch indices , and model output space $\mathcal{Z}(D; \mathbb{R}^{d_z})$, with domain $D \subset \mathbb{R}^d$.

We define T_{in} similarly to (22) in the previous section, via a composition of Equations (14), (20) and (21); however here the linear lifting operator is chosen to be a pointwise linear transformation $W_{\text{in}} \in \mathbb{R}^{d_{\text{model}} \times (d_u + d)}$. Namely, we define $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^{d_u}) \to \mathcal{V}$ via

$$\Big(\mathsf{T}_{\text{in}}(u)\Big)(j)(x) := W_{\text{in}}\Big(\mathsf{S}_{\text{reshape}}\big(u_{\text{in}}\big)(j)\Big)(x), \tag{30}$$

for any $j \in [\![1, P]\!]$ and any $x \in D$, where $u_{\text{in}} \in \mathcal{U}(D; \mathbb{R}^{d_u + d})$ is defined as in (14), $\mathsf{S}_{\text{reshape}}$ as in (20) and the nonlocal operation (21) is replaced by a local pointwise one.

Next, we define the particulars of $\mathsf{E}_L : \mathcal{V} \to \mathcal{V}$ in Equation (10). The key distinction between the Fourier attention neural operator that we define here and the ViT neural operator from the previous subsection 4.3 lies in the nonlocality of operations performed within $\mathsf{A}_{\text{MultiHead}}$. Here, in each layer the operator $\mathsf{A}_{\text{MultiHead}} : \mathcal{V} \to \mathcal{V}$ is the multi-head attention operator from (11) based on

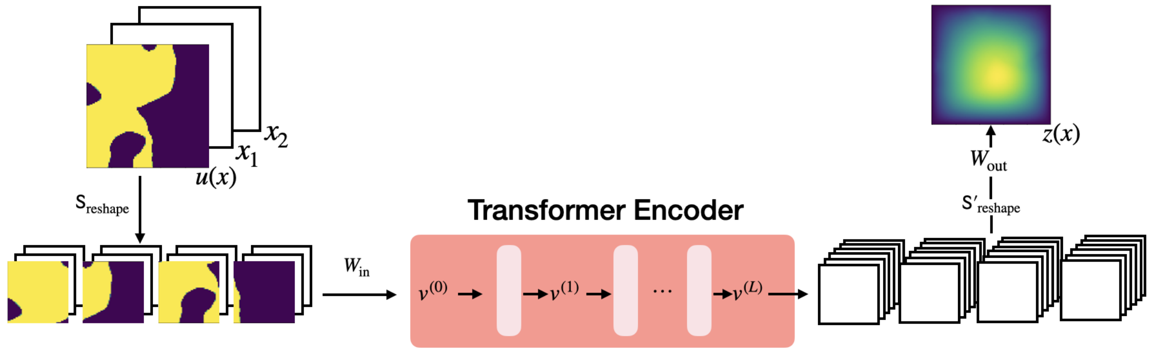

Figure 4: Fourier Attention Neural Operator.

the self-attention operator from Definition 14. Letting $H \in \mathbb{N}$ denote the number of attention heads, the multi-head attention operator is parametrized by learnable nonlocal linear integral operators $\mathsf{Q}_h, \mathsf{K}_h \in \mathcal{L}\Big(\mathcal{U}(D', \mathbb{R}^{d_{\text{model}}}); L^2(D', \mathbb{R}^{d_K})\Big)$ and $\mathsf{V}_h \in \mathcal{L}\Big(\mathcal{U}(D', \mathbb{R}^{d_{\text{model}}}); \mathcal{U}(D', \mathbb{R}^{d_K})\Big)$, for $h = 1, \dots, H$, where for implementation purposes d_K is chosen as $d_K := d_{\text{model}}/H$. The specific formulation of these linear integral operators may be found in Definition 20 and the subsequent discussion. The multi-head attention operator is thus defined by the application of a pointwise linear transformation $W_{\text{MultiHead}} \in \mathbb{R}^{d_{\text{model}} \times H \cdot d_K}$ to a concatenation of the outputs of $H \in \mathbb{N}$ self-attention operators. The operators $\mathsf{W}_1, \mathsf{W}_2$ are defined as in (23). The layer normalization operator $\mathsf{F}_{\text{LayerNorm}} : \mathcal{V} \to \mathcal{V}$ is defined as in (24) so that it is applied to every point in every patch. Furthermore, the operator $\mathsf{F}_{\text{NN}} : \mathcal{V} \to \mathcal{V}$ is defined as in (25).

We define T_{out} identically to Equation (28). We refer to Remark 21 for an additional operation that can be applied to ameliorate the effect of patch discontinuities. Finally, the composition of the operators $\mathsf{T}_{\text{in}}, \mathsf{E}_L, \mathsf{T}_{\text{out}}$ completes the definition of the Fourier attention neural operator. In Appendix B.3 we outline how this neural operator architecture is implemented in the finite-dimensional setting, where the input function $u \in \mathcal{U}\big(D; \mathbb{R}^{d_u}\big)$ is defined on a finite set of discretization points $\{x_i\}_{i=1}^N \subset D$.

5. Universal Approximation by Transformer Neural Operators

Universal approximation is a minimal requirement for neural operators. When the operator to be approximated maps between spaces of functions defined over Euclidean domains, universal approximation necessarily requires both nonlocality and nonlinearity. In the recent paper Lanthaler et al. (2025), a minimal architecture possessing these two properties is exhibited. In this section, rather than trying to prove universal approximation for the various discretization-invariant neural operators introduced in the paper, we construct a simple canonical setting that exhibits the universality of the attention mechanism on function space, allowing direct exploitation of the ideas in Lanthaler et al. (2025). We build on the notation used in Subsection 4.1. In particular A denotes the self-attention operator from Definition 5 and f an activation function. Throughout this section, we consider activation functions $f \in C^\infty(\mathbb{R})$ which are non-polynomial and Lipschitz continuous.

To be specific, we consider neural operators $\mathsf{G} : \mathcal{U}\big(D; \mathbb{R}^{d_u}\big) \times \Theta \to \mathcal{Z}\big(D; \mathbb{R}^{d_z}\big)$ of the form

$$\mathsf{G}(u; \theta) := \Big(\mathsf{T}_{\text{out}} \circ \mathsf{E} \circ \mathsf{T}_{\text{in}}\Big)(u; \theta), \tag{31}$$

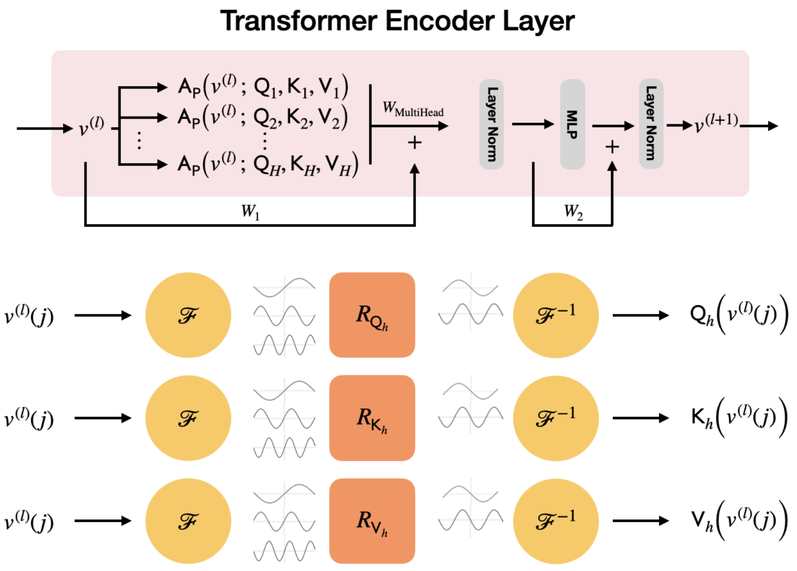

Figure 5: Encoder Layer of the Fourier Attention Neural Operator.

where $\mathsf{T}_{\text{in}} : \mathcal{U}(D; \mathbb{R}^{d_u}) \to \mathcal{V}(D; \mathbb{R}^{d_{\text{model}}})$ and $\mathsf{T}_{\text{out}} : \mathcal{V}(D; \mathbb{R}^{d_{\text{model}}}) \to \mathcal{Z}(D; \mathbb{R}^{d_z})$ are defined by neural networks of the form

$$\Big(\mathsf{T}_{\text{in}}(u)\Big)(x) = R_2 f\Big(R_1\big(u(x), x\big) + b_R\Big) + b'_R, \tag{32}$$

$$\Big(\mathsf{T}_{\text{out}}(v)\Big)(x) = P_2 f\Big(P_1\big(v(x), x\big) + b_P\Big) + b'_P, \tag{33}$$

where $\big(u(x), x\big) \in \mathbb{R}^{d_u + d}$ and $\big(v(x), x\big) \in \mathbb{R}^{d_{\text{model}} + d}$, where R_1, R_2, P_1, P_2 are learned linear transformations of appropriate dimensions and b_R, b'_R, b_P, b'_P are learned vectors. We define the operator $\mathsf{E} : \mathcal{V}(D; \mathbb{R}^{d_{\text{model}}}) \to \mathcal{V}(D; \mathbb{R}^{d_{\text{model}}})$ as a variant of the layer defined by the iteration step in Equation (10), given by the two-step map acting on its inputs $v \in \mathcal{V}$ as

$$v(x) \leftarrow W_1 v(x) + \mathsf{A}\big(v; Q, K, V\big)(x), \tag{34a}$$

$$v(x) \leftarrow W_2 v(x) + W_3 f\big(W_4 v(x) + b_1\big) + b_2, \tag{34b}$$

for any $x \in D$. Note that the neural operator thus defined does not include layer normalizations; it is thus a variant of the transformer neural operator from Subsection 4.2. We may now apply the result of Lanthaler et al. (2025) to show two universal approximation theorems for the resulting transformer neural operator.

Theorem 22 *Let $D \subset \mathbb{R}^d$ be a bounded domain with Lipschitz boundary, and fix integers $s, s' \geq 0$. If $\mathsf{G}^\dagger : C^s\big(\bar{D}; \mathbb{R}^r\big) \to C^{s'}\big(\bar{D}; \mathbb{R}^{r'}\big)$ is a continuous operator and $K \subset C^s\big(\bar{D}; \mathbb{R}^r\big)$ a compact set, then for any $\epsilon > 0$, there exists a transformer neural operator $\mathsf{G}(\cdot; \theta) : K \subset C^s\big(\bar{D}; \mathbb{R}^r\big) \to C^{s'}\big(\bar{D}; \mathbb{R}^{r'}\big)$ so that*

$$\sup_{u \in K} \left\| \mathsf{G}^\dagger(u) - \mathsf{G}(u; \theta) \right\|_{C^{s'}} \leq \epsilon. \tag{35}$$

Proof We begin by noting that for $Q, K = 0$ and $V = I$, the self-attention mapping reduces to

$$\mathsf{A}\big(v; Q, K, V\big)(\cdot) = \mathbb{E}_{y \sim p(y:v,x)}[V v(\cdot)] = \frac{1}{|D|} \int v(x) \, \mathrm{d}x. \tag{36}$$

For weights $W_2 = 0$, $W_3 = W_4 = I$ and $b_2 = 0$, the transformer encoder neural operator layer (34) reduces to the mapping

$$v(\cdot) \mapsto f\left(W_1 v(\cdot) + b_1 + \frac{1}{|D|}\int v(x)\, \mathrm{d}x\right). \tag{37}$$

The existence of $W_1, R_1, R_2, P_1, P_2, b_1, b_R, b_R', b_P, b_P'$ so that $\mathsf{G}(\cdot\,;\theta)$ satisfies (35) then follows from Lanthaler et al. (2025, Theorem 2.1), which also involves the application of the universality result for two-layer neural networks of Pinkus (1999, Theorem 4.1). ■

The analysis in Lanthaler et al. (2025) allows the derivation of an analogous universal approximation theorem for functions belonging to Sobolev spaces. This generalization is the content of the next theorem. The proof follows easily by applying the same argument as in the proof of Theorem 22 and the result of Lanthaler et al. (2025, Theorem 2.2).

Theorem 23 *Let $D \subset \mathbb{R}^d$ be a bounded domain with Lipschitz boundary and fix integers $s, s' \geq 0$, $p, p' \in [1,\infty)$. If $\mathsf{G}^\dagger : W^{s,p}(D;\mathbb{R}^r) \to W^{s',p'}(D;\mathbb{R}^{r'})$ is a continuous operator and $K \subset W^{s,p}(D;\mathbb{R}^r)$ a compact set of bounded functions so that $\sup_{u\in K}\|u\|_{L^\infty} < \infty$, then for any $\epsilon > 0$, there exists a transformer neural operator $\mathsf{G}(\cdot;\theta) : K \subset W^{s,p}(D;\mathbb{R}^r) \to W^{s',p'}(D;\mathbb{R}^{r'})$ so that*

$$\sup_{u\in K}\left\|\mathsf{G}^\dagger(u) - \mathsf{G}(u;\theta)\right\|_{W^{s',p'}} \leq \epsilon. \tag{38}$$

6. Numerical Experiments

In this section we illustrate, through numerical experiments, the capability of the transformer neural operator architectures described in Section 4. Throughout, we consider the supervised learning problem described by the data model in (3) and take the viewpoint of surrogate modeling to construct approximations of the operators $\mathsf{G}^\dagger$. In Subsection 6.1 we consider problems given by dynamical systems and ordinary differential equations, namely operators acting on functions defined on a one-dimensional time domain. In this context, we explore the use of the transformer neural operator for operator learning problems given by the Lorenz ‘63 dynamical system and a controlled ODE. On the other hand, in Subsection 6.2 we consider problems given by partial differential equations equations, namely operators acting on functions defined on a two-dimensional spatial domain. In this context we explore the use of the transformer neural operator and of the more efficient ViT neural operator and Fourier attention neural operator architectures for operator learning problems given by the Darcy flow and Navier-Stokes equations. The code repository for the experiments in this paper may be found at https://github.com/EdoardoCalvello/TransformerNeuralOperators.

Remark 24 (Relative Loss) *For each neural operator architecture we optimize for the parameters $\theta \in \Theta$ according to the relative loss, i.e.*

$$\inf_{\theta\in\Theta} \frac{1}{J}\sum_{j=1}^{J}\left(\frac{\|\mathsf{G}(u^{(j)};\theta) - \mathsf{G}^\dagger(u^{(j)})\|_{\mathcal{Z}}}{\|\mathsf{G}^\dagger(u^{(j)})\|_{\mathcal{Z}}}\right), \tag{39}$$

for data points $\{u^{(j)}\}_{j=1}^J$.

6.1 1D Time Domain

In this subsection we consider problems where attention is applied to functions in the one-dimensional time domain. The first setting we consider is that of the Lorenz ‘63 dynamical system (Lorenz, 1963), a simplified model for atmospheric convection. The dynamics are given by

$$\begin{aligned} dx &= \sigma(y-x)dt, \\ dy &= \big(x(\rho-z)-y\big)dt, \\ dz &= \big(xy-\beta z\big)dt, \end{aligned} \tag{Lorenz ‘63}$$

with parameters set to $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. We consider learning the operator $\mathsf{G}^\dagger : C\big([0,T];\mathbb{R}^{d_x}\big) \times \mathbb{R}^{d_y} \times \mathbb{R}^{d_z} \to C\big([0,T];\mathbb{R}^{d_y}\big) \times C\big([0,T];\mathbb{R}^{d_z}\big)$ given by

$$\mathsf{G}^\dagger : \{x(t), y(0), z(0)\}_{t\in[0,T]} \mapsto \{y(t), z(t)\}_{t\in[0,T]}. \tag{40}$$

In this case, $\mathsf{G}^\dagger$ is known to exist and given by the solution operator for the system of linear ODEs arising from the second and third components of (Lorenz ‘63), given known first component. We note that including the initial condition of the unobserved trajectories in the model introduces a discrepancy in dimension within the input function to the model. Namely, denoting by u the input to the model, $u(0) \in \mathbb{R}^3$ while $u(t) \in \mathbb{R}$ for $t \neq 0$. We overcome this issue by learning separate embedding linear transformations W_{in_0} and W_{in}, for the initial condition and the rest of the trajectory, respectively. In Figure 6 we display the results obtained by applying a transformer neural operator model on a test data set of 1000 samples with the same discretization ($\Delta t = 0.01$ and $T = 2$) as the training set of 8000 samples on the operator learning problem described by (40). The operator is successfully learned, and displays median relative L^2 error of less than 1%, with worst case rising to just below 10%.

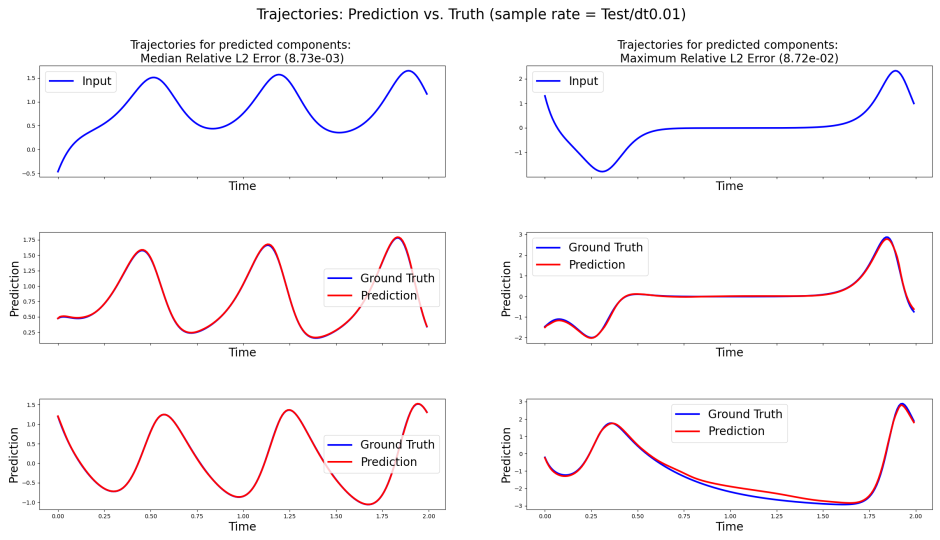

Figure 6: The panel displays the performance of the transformer neural operator when applied to the Lorenz ‘63 operator learning problem of recovering both unobserved y and z trajectories. In each column, the top plot shows the input x trajectory data while the second and third row display the predicted against true y and z trajectories, respectively; the left column concerns the testing example achieving a median relative L^2 error, and the right columns shows the test example achieving the worst error.

We also experiment with a setting for which $\mathsf{G}^\dagger$ is not well-defined: we study the recovery of unobserved trajectory $\{y(t)\}_{t\in[0,T]}$ from the observed trajectory $\{x(t)\}_{t\in[0,T]}$ so that $\mathsf{G}^\dagger : C\big([0,T];\mathbb{R}^{d_x}\big) \to$

$C\big([0,T];\mathbb{R}^{d_y}\big)$ and

$$\mathsf{G}^\dagger : \{x(t)\}_{t\in[0,T]} \mapsto \{y(t)\}_{t\in[0,T]}.$$

This operator would only be well-defined if $(y(0), z(0))$ were included as inputs. We nonetheless expect accurate recovery of the hidden trajectory by the property of synchronization (Pecora and Carroll, 1990; Hayden et al., 2011).

We investigate an additional operator learning problem given by the setting of the controlled ODE

$$\begin{aligned} dz &= \sin(z)du, \qquad\qquad z(0) = z_0,\\ u(t) &= \sum_{j=1}^{J} \xi_j \sin(\pi\eta_j t), \quad \xi_j \sim \mathcal{N}(0,1)\,\text{i.i.d.}\ , \eta_j \sim \mathsf{unif}([0,J])\,\text{i.i.d.}\ . \end{aligned} \tag{CDE}$$

In this case we aim to approximate the operator $\mathsf{G}^\dagger : C\big([0,T];\mathbb{R}^{d_u}\big) \to C\big([0,T];\mathbb{R}^{d_z}\big)$ defined by

$$\mathsf{G} : \{u(t);\xi,\eta\}_{t\in[0,T]} \mapsto \{z(t);\xi,\eta\}_{t\in[0,T]}.$$

For this problem we make the choice $J = 10$.

The experimental settings we consider for the continuous time operator learning problem may be viewed through the lens of data assimilation as smoothing problems (Sanz-Alonso et al., 2023). Indeed, the mappings to be learned represent the recovery of an unobserved trajectory $\{z(t)\}_{t\in[0,T]}$ conditioned on the observation of a whole trajectory $\{u(t)\}_{t\in[0,T]}$.

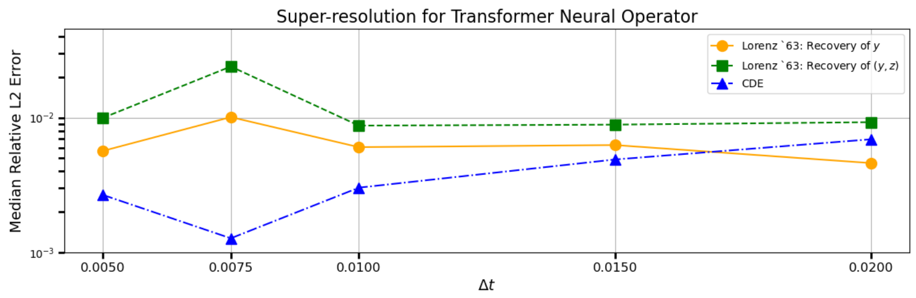

Figure 7: The panel displays the performance in relative L^2 errors (in log-scale) of the transformer neural operator when applied to the Lorenz '63 and controlled ODE operator learning test sets of different resolutions at inference time (without retraining). All models were parametrized by $d_{\text{model}} = 128$ and 6 encoder layers, and were trained with 8000 trajectories with $T = 2$, sampled according to $\Delta t = 0.01$.

In Figure 7 we demonstrate the mesh-invariance property of the transformer neural operator in the context of the operator learning problems arising from equations (Lorenz '63) and (CDE). A key motivation for considering the continuum formulation of attention and the related transformer neural operator is the property of zero-shot generalization to different non-uniform meshes. To demonstrate this, in Figure 8 we consider the operator learning problem of mapping the observed x trajectory of (Lorenz '63) to the unobserved y trajectory. We display the median relative L^2 error and worst error samples from a test set consisting of trajectories $\{x(t)\}_{t\in\mathfrak{T}_{\text{test}}}$, where the test time index set $\mathfrak{T}_{\text{test}}$ is defined as

$$\mathfrak{T}_{\text{test}} := \{n\Delta t\}_{n=0}^{N/2} \cup \{2n\Delta t\}_{n=N/2+1}^{3N/4}, \tag{41}$$

for $\Delta t := T/N$, where the model deployed is trained on trajectories indexed at a set $\mathfrak{T}_{\text{train}}$ defined by

$$\mathfrak{T}_{\text{train}} := \{n\Delta t\}_{n=0}^{N}. \tag{42}$$

In other words, the second half of the testing domain is up-sampled by a factor of 2 compared to the training discretization. As seen qualitatively in Figure 8 testing on the non-uniform discretization based on a model trained on the uniform grid is successful. The use of different grids in test and train data does lead to a larger error than the one incurred when they match (Figure 7), but the errors remain small. It is our continuum formulation of attention from Appendix B.1 that enables this direct generalization; indeed, as displayed in Figure 9, the transformer from Vaswani et al. (2017) when applied to the same setting yields larger errors than the transformer neural operator, due to the TNO's reweighting based on the discretization of the grid.

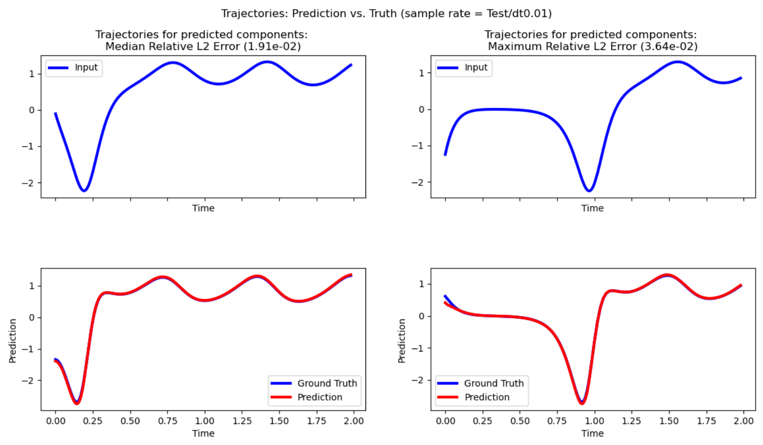

Figure 8: The panel displays the performance in relative L^2 errors of the transformer neural operator when applied to the Lorenz '63 operator learning problem of recovering the unobserved y trajectory. The results displayed concern a model trained on trajectories indexed on the index set (42) deployed on a test of trajectories indexed on (41). In each column, the top plot shows the input data and the bottom compares the prediction and ground truth; the left column shows the testing example achieving a median error, and the right columns shows the test example achieving the worst error.

6.2 2D Spatial Domain

In this subsection we study two different 2d operator learning problems – flow in a porous medium, governed by the Darcy model, and incompressible viscous fluid mechanics governed by the Navier–Stokes equation, in particular, a Kolmogorov flow. We apply the three architectures developed in Sections 2 and 3, concentrating on the patched architectures. In the first Subsection 6.2.1 we report the results obtained by applying the neural operators described to an array of operator learning problems, where the domain of the functions is two-dimensional i.e. $D \subset \mathbb{R}^2$. In Subsection 6.2.2 we describe the settings of the two Darcy flow operator learning problems considered, along with the parametrization and training details for the various architectures. Similarly in Subsection 6.2.3, we describe the Kolmogorov flow operator learning problem and relevant implementation details. Subsections 6.2.2 and 6.2.3 also contain further numerical results illustrating the behavior of the proposed transformer neural operators, and discretization invariance in particular.

6.2.1 Results

The results presented concern the two variants of Darcy flow, with lognormal and piecewise constant inputs, and the Kolmogorov flow. The operator learning architectures we deploy include the three transformer-based methods introduced in this paper, and the implementations of the Fourier neural operator from the "NeuralOperator" library (Li et al., 2021; Kovachki et al., 2023), the Galerkin

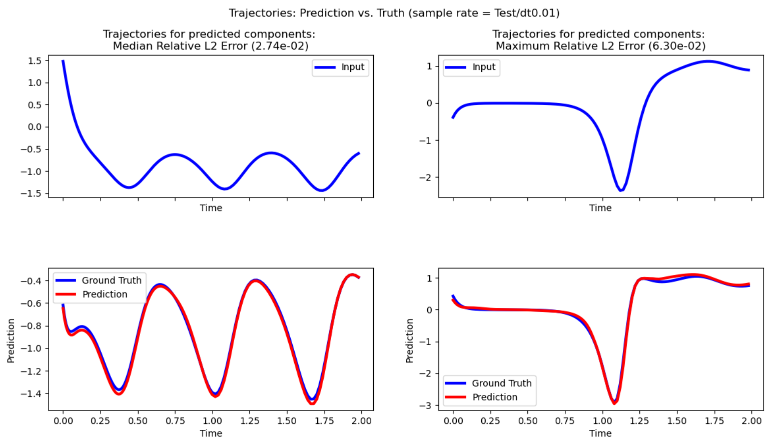

Figure 9: The panel displays the performance in relative L^2 errors of the transformer from Vaswani et al. (2017) when applied to the Lorenz '63 operator learning problem of recovering the unobserved y trajectory. The results displayed concern a model trained on trajectories indexed on the index set (42) deployed on a test of trajectories indexed on (41). In each column, the top plot shows the input data and the bottom compares the prediction and ground truth; the left column shows the testing example achieving a median error, and the right columns shows the test example achieving the worst error.

transformer from the "galerkin-transformer" library (Cao, 2021), and the AFNO (Guibas et al., 2022) from the "physics-nemo" library (PhysicsNeMo Contributors, 2023). We note that in order to obtain a numerical comparison of the attention mechanisms themselves, in the context of the Galerkin transformer architecture we only use the Galerkin transformer encoder, with lifting and projections achieved by linear layers as in TNO. This allows for a more direct comparison between the Fourier attention from Cao (2021) and our continuum attention, without potential expressivity gain from additional architectural components. In the following discussion, we do not include experiments performed using the Codomain-attention neural operator (Rahman et al., 2024) in the high-resolution settings: the memory usage with the implementation from the "NeuralOperator" library did not allow for comparably sized models, in terms of parameters; the performance of the models we trained was hence incomparably inferior to the others.

Our first results compares the transformer neural operator from Subsection 4.2 with FNO, AFNO and the Galerkin transformer (employing Fourier attention) on the lognormal Darcy Flow problem in a low 64×64 resolution setting. Table 1 provides a brief summary of the test relative L^2 errors obtained. It is notable that the transformer architecture obtains the best performance with an order of magnitude fewer parameters. However, since higher resolution renders the architecture from Subsection 4.2 impractical in dimensions $d \geq 2$ the remainder of our experiments use the patched ViT and Fourier attention neural operators from Subsections 4.3 and 4.4 respectively.

We now consider training two patched architectures on the Darcy Flow problem with piecewise constant inputs and the Kolmogorov Flow problem; in both cases we use a 416×416 resolution setting. We compare cost versus accuracy for the different neural operators implemented, defining cost in terms of both number of parameters and number of FLOPS. In Table 2 we present the FLOPS for each method; details of the derivation are provided in Appendix C. We note that the parameter scaling for the transformer neural operator is given by $\mathcal{O}\big(d_{\text{model}}^2\big)$, while the scaling for the FNO and the Fourier attention neural operator is $\mathcal{O}\big(d_{\text{model}}^2 \cdot k_{\max}\big)$, where $k_{\max}$ is the total number of Fourier modes used. The parameter scaling for the ViT neural operator is given by $\mathcal{O}\big(d_{\text{model}} \cdot k_{\max} + d_{\text{model}}^2\big)$. We note that the parameter scaling for the AFNO architecture is given by $\mathcal{O}\big(N d_{\text{model}} + d_{\text{model}}^2\big)$,

Architecture	**Number of Parameters**	**Darcy Flow Lognormal Input**
Transformer NO	$5.98 \cdot 10^5$	$1.19 \cdot 10^{-2}$
FNO	$5.70 \cdot 10^6$	$1.96 \cdot 10^{-2}$
AFNO	$1.10 \cdot 10^6$	$2.57 \cdot 10^{-2}$
Galerkin Transformer	$6.04 \cdot 10^5$	$7.30 \cdot 10^{-2}$

Table 1: Median relative L^2 error for each architecture applied to a low resolution experimental setting.

where the Nd_{model} appears because of the learnable positional encoding. Further details are provided in Appendix C.

Architecture	**Scaling**
FNO	$\mathcal{O}\big(d_{\text{model}} N \log(\sqrt{N}) + N d_{\text{model}}^2\big)$
AFNO	$\mathcal{O}\big(d_{\text{model}} N \log(\sqrt{N}) + N d_{\text{model}}^2 / b\big)$
Transformer NO	$\mathcal{O}\big(d_{\text{model}} N^2 + N d_{\text{model}}^2\big)$
ViT NO	$\mathcal{O}\Big(d_{\text{model}} N \big(P + \log(\sqrt{N/P})\big) + N \log(\sqrt{N}) + N d_{\text{model}}^2\Big)$
FANO	$\mathcal{O}\Big(d_{\text{model}} N \big(P + \log(\sqrt{N/P})\big) + N \log(N) + N d_{\text{model}}^2\Big)$

Table 2: Evaluation cost (measured in FLOPS) for the three proposed transformer neural operators compared to the Fourier neural operator (Li et al., 2021) and adaptive Fourier neural operator (Guibas et al., 2022). We include the scaling of the AFNO as implemented for our use case, i.e. not employing patching and not employing mode truncation; we further note that in the context of this architecture b is a chosen integer hyperparameter. We give further details on the derivation of these scalings in Appendix C.

Figures 10 and 11 display the cost-accuracy trade-off. Table 3 provides the test relative L^2 errors values, depicted in the cost-accuracy analysis figures, obtained with the models with the highest parameter count. Figure 10 demonstrates the parameter-efficiency of the ViT neural operator in comparison with the FNO, AFNO and Fourier attention neural operator. Once cost is defined in terms of FLOPS it is apparent that both patched transformer architectures outperform the FNO; furthermore in the Kolmogorov flow problem, the FNO exhibits accuracy-saturation, caused by the finite data set, whereas the patched architectures are less affected, in the ranges we test here. This suggests that the patched transformer architectures use the information content in the data more efficiently. The AFNO is comparable in terms of parameter count to FANO but more costly than ViTNO. On the other hand, even with no patching, the AFNO is cheaper in terms of evaluation complexity. Both ViTNO and FANO outperform the AFNO in terms of parameter-accuracy tradeoff in the context of the Darcy flow problem. For the same problem ViTNO and FANO bring an improvement in accuracy, albeit at a higher computational complexity. In the context of the Kolmogorov flow problem, ViTNO and FANO are competitive with AFNO in parameter-accuracy tradeoff. However for this problem, the AFNO can achieve higher accuracy with a lower complexity. It is important to note that because of the learnable positional embedding, and potential strided convolutions (if using patching), the AFNO is not a neural operator. This means that applying this architecture to different resolutions would require retraining, thus significantly increasing computational complexity. This is a fundamental advantage of the other architectures as their parameters may be deployed across different resolutions for training and inference. We recall the example of GenCast (Price et al., 2025), where a medium-range weather forecasting model is trained using a neural operator at low resolutions, and then finetuned at higher resolutions, thus drastically reducing the cost for such a large scale application.

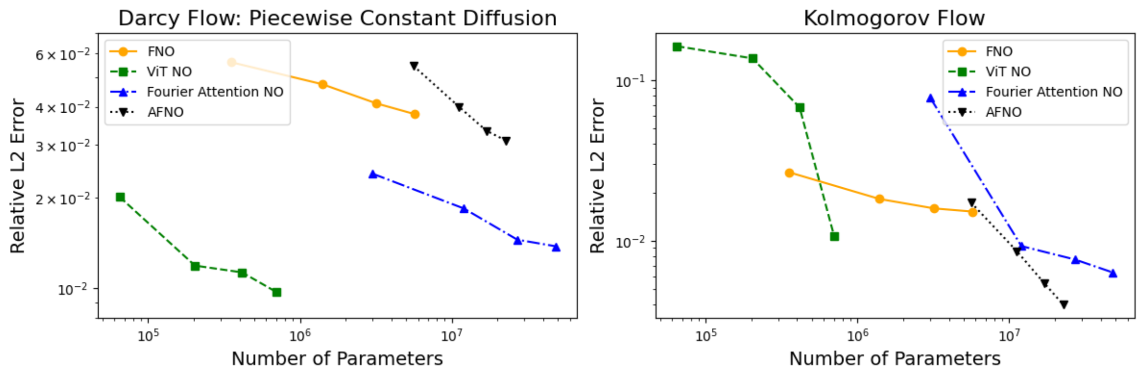

Figure 10: The two panels display the test errors (in log-scale) against number of parameters for the architectures with the channel widths of 32, 64, 96 and 128.

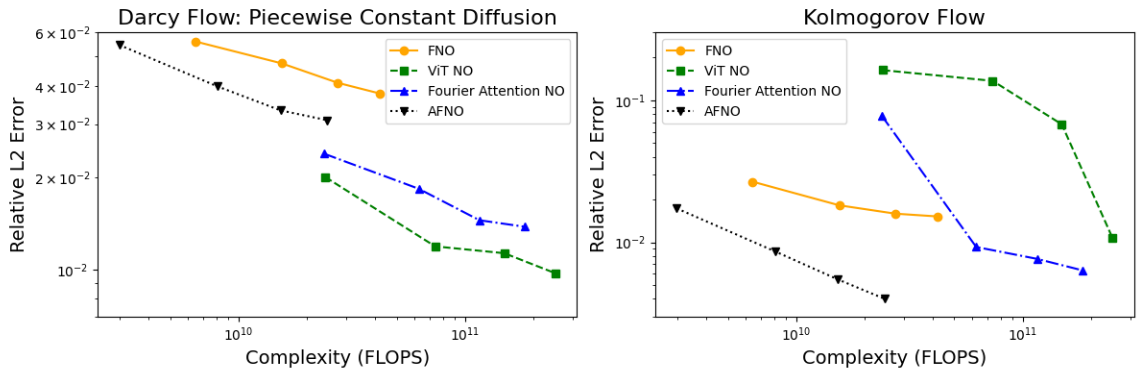

Figure 11: The two panels display the test errors (in log-scale) against evaluation cost in FLOPS for the architectures with the channel widths of 32, 64, 96 and 128.

Architecture	**Darcy Flow Lognormal Input**	**Darcy Flow Piecewise Constant Input**	**Kolmogorov Flow**
ViT NO	$7.68 \cdot 10^{-3}$	$9.72 \cdot 10^{-3}$	$1.07 \cdot 10^{-2}$
FANO	$8.39 \cdot 10^{-3}$	$1.38 \cdot 10^{-2}$	$6.34 \cdot 10^{-3}$
FNO	$2.38 \cdot 10^{-2}$	$3.78 \cdot 10^{-2}$	$1.52 \cdot 10^{-2}$
AFNO	$2.66 \cdot 10^{-2}$	$3.09 \cdot 10^{-2}$	$3.99 \cdot 10^{-3}$

Table 3: Median relative L^2 error for each architecture applied to the different high resolution experimental settings.

6.2.2 Darcy Flow

We consider the linear, second-order elliptic PDE defined on the unit square

$$\begin{aligned} -\nabla \cdot \big(a(x)\nabla u(x)\big) &= f(x), \quad && x \in (0,1)^2, \\ u(x) &= 0, && x \in \partial(0,1)^2, \end{aligned} \tag{Darcy Flow}$$

which is the steady-state of the 2d Darcy flow equation. The equation is equipped with Dirichlet boundary conditions and is well-posed when the diffusion coefficient $a \in L^\infty\big((0,1)^2;\mathbb{R}_+\big)$ and the forcing function is $f \in L^2\big((0,1)^2;\mathbb{R}\big)$. In this context, we will consider learning the nonlinear operator $\mathsf{G}^\dagger : L^\infty\big((0,1)^2;\mathbb{R}_+\big) \to H_0^1\big((0,1)^2;\mathbb{R}\big)$ defined as

$$\mathsf{G}^\dagger : a \mapsto u.$$

We consider two different inputs. In both experimental settings the forcing function is chosen as $f \equiv 1$.

In the first the data points $\{a^{(j)}\}_{j=1}^J$ are sampled from the probability measure

$$\mu_1 := (T_1)_\sharp \mathcal{N}\big(0, C\big),$$

where $\mathcal{N}\big(0, C\big)$ is a Gaussian measure with covariance operator C defined as

$$C = 12^2\big(-\Delta + 6^2 I\big)^{-2},$$

where the Laplacian is equipped with zero Neumann boundary conditions and viewed as acting between spaces of spatially mean-zero functions, and $T_1(\cdot) = \exp(\cdot)$ is the exponential function. We hence refer to the data inputs $a^{(j)}$ as being *lognormal*. In the second setting $\{a^{(j)}\}_{j=1}^J$ are sampled from the probability measure

$$\mu_2 := (T_2)_\sharp \mathcal{N}\big(0, C\big),$$

where the covariance operator is defined as before and

$$T_2(x) = \begin{cases} 12, & x \geq 0, \\ 3, & x < 0. \end{cases}$$

We hence refer to the data inputs $a^{(j)}$ as being *piecewise constant.*

In the context of the lognormal experimental setting, we investigate a scenario with low resolution training samples, for which pointwise attention and hence the transformer neural operator from Subsection 4.2 is suitable, and a high resolution scenario where patching becomes necessary for application of the patched-based architectures of Subsections 4.3 and 4.4.

For the low resolution setting, we train the FNO, AFNO, Galerkin transformer and TNO architectures using 3600 independent samples of resolution 64×64, we use 200 samples for validation and 200 samples for testing. The Fourier neural operator is parametrized by 12 Fourier modes in each dimension and channel width of 128; the model is trained using a batch size of 8 and learning rate of 10^{-4}. On the other hand, the transformer neural operator is parametrized by 128 channels in the encoder and is trained using a batch size of 2 and learning rate of 10^{-3}. The AFNO is instantiated with 4 layers of 128 hidden channels, and does not employ patching; it is trained using a learning rate of 10^{-3} and batch size of 1. The Galerkin transformer uses the Fourier type attention from Cao (2021), with quadratic complexity, making it prohibitive for higher resolutions. It uses 4 layers of 128 hidden channels and is trained using a learning rate of 10^{-3} and batch size of 2. On this problem we train all neural operators using the relative H^1 loss.

For the high resolution setting, we train the AFNO, FNO, ViT neural operator and Fourier attention neural operator architectures using 3600 independent samples of resolution 416×416; we

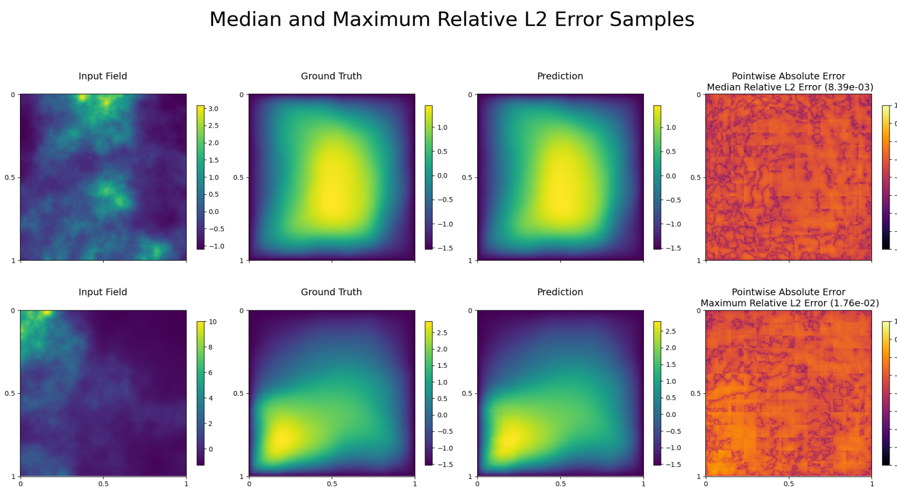

Figure 12: The panel displays the result of the application of the Fourier attention neural operator on the Darcy flow experiment with lognormal diffusion for the median and maximum relative L^2 error samples. The first row shows the sample from the test set of the same resolution as the training set yielding the median relative L^2 error. On the other hand, the second row displays the sample yielding the maximum relative L^2 error. The first column of the panel displays the input diffusion coefficient $a(x)$ of the relevant sample. The second column shows the true solution $u(x; a)$, while the third column displays the the predicted solution. The last column displays the pointwise absolute error (in log-scale) between the prediction and truth.

use 200 samples for validation and 200 samples for testing. The Fourier neural operator is again parametrized by 12 Fourier modes in each dimension and channel width of 128; the model is trained using a batch size of 8 and learning rate of 10^{-4}. The AFNO is instantiated with 4 layers of 128 hidden channels, and does not employ patching; it is trained using a learning rate of 10^{-3} and batch size of 1. Both the ViT neural operator and Fourier attention neural operator are parametrized by 128 channels in the transformer encoder. The integral operator in the ViT NO is parametrized by 12 Fourier modes in each dimension while the integral operators appearing in the Fourier attention neural operator are parametrized by 9 Fourier modes in each dimension. Both architectures employ a final spectral convolution layer (see Remark 21) that is parametrized by 64 Fourier modes in each dimension. Both of these neural operators are trained using a batch size of 1 and learning rate of 10^{-3}. On this problem we train all the neural operators with the relative H^1 loss. Figure 12 displays the high resolution test results obtained by applying the Fourier attention neural operator.

In the context of the piecewise constant experimental setting, we train the AFNO, FNO, ViT neural operator and Fourier attention neural operator architectures using 3600 independent samples of resolution 416×416, we use 200 samples for validation and 200 samples for testing. The Fourier neural operator is again parametrized by 12 Fourier modes in each dimension and channel width of 128; the model is trained using a batch size of 8 and learning rate of 10^{-4}. The AFNO is instantiated with 4 layers of 128 hidden channels, and does not employ patching; it is trained using a learning rate of 10^{-3} and batch size of 1. Both the ViT neural operator and Fourier attention neural operator are parametrized by 128 channels in the transformer encoder. The integral operator in the ViT NO is parametrized by 12 Fourier modes in each dimension while the integral operators appearing in the Fourier attention neural operator are parametrized by 9 Fourier modes in each dimension. Both

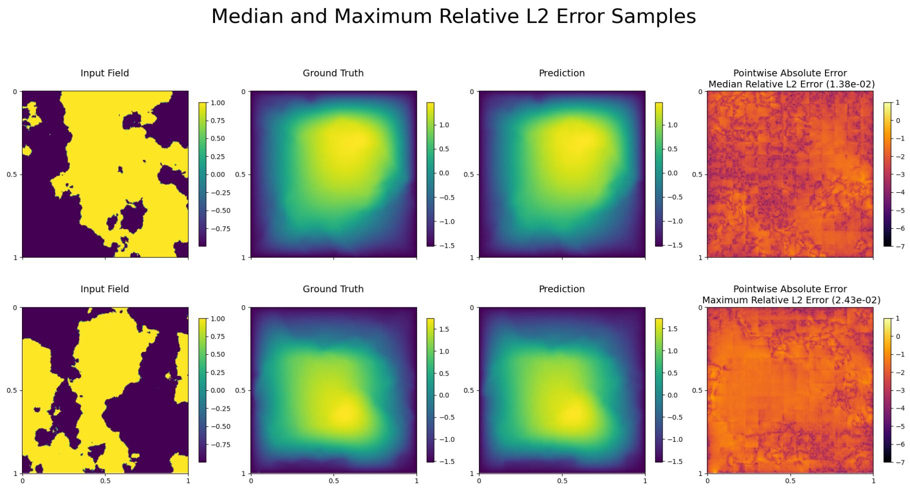

Figure 13: The panel displays the result of the application of the Fourier attention neural operator on the Darcy flow experiment with piecewise constant diffusion for the median and maximum relative L^2 error samples. The first row displays the sample from the test set of the same resolution as the training set yielding the median relative L^2 error. On the other hand, the second row displays the sample yielding the maximum relative L^2 error. The first column of the panel displays the input diffusion coefficient $a(x)$ of the relevant sample. The second column shows the true solution $u(x; a)$, while the third column displays the the predicted solution. The last column displays the pointwise absolute error (in log-scale) between the prediction and truth.

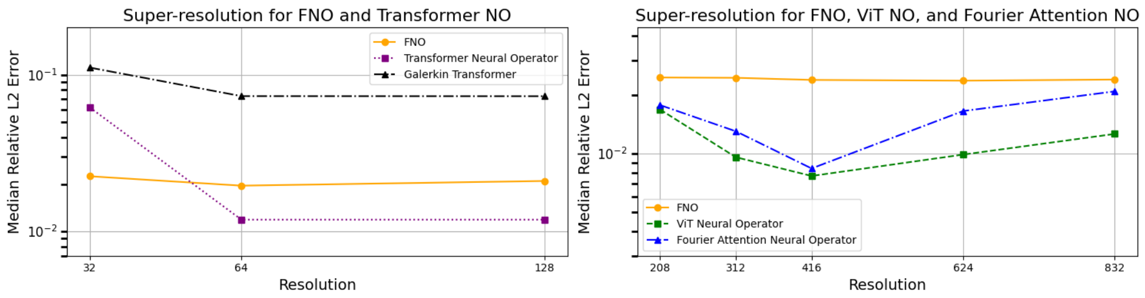

Figure 14: The panel displays the performance in relative L^2 errors (in log-scale) of the various architectures when applied to Darcy flow with lognormal diffusion test sets of different resolutions at inference time (without retraining). The left panel concerns FNO, the transformer neural operator and the Galerkin transformer when trained on data of resolution 64×64. On the other hand, the right panel concerns the FNO, ViT neural operator and Fourier attention neural operator architectures trained on data of resolution 416×416.

architectures employ a final spectral convolution layer (see Remark 21) that is parametrized by 64 Fourier modes in each dimension. Both of these neural operators are trained using a batch size of 1 and learning rate of 10^{-3}. On this problem we train all the neural operators with the relative H^1 loss. Figure 13 displays the high resolution test results obtained by applying the Fourier attention neural operator.

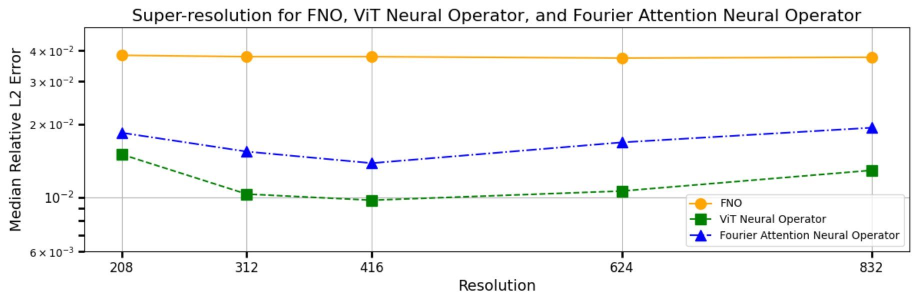

Figure 15: The panel displays the performance in relative L^2 errors (in log-scale) of the FNO, ViT neural operator and Fourier attention neural operator when applied to Darcy flow with piecewise constant diffusion test sets of different resolutions at inference time (without retraining). All architectures are trained on data that is of resolution 416×416.

In Figures 14 and 15 we display the results of applying the neural operator architectures for the lognormal and piecewise input settings, respectively, to test samples of different resolutions than the one used for training. The results demonstrate the zero-shot generalization to different resolutions capability of the transformer neural operator architectures, which does not require retraining of the models. In the lognormal setting the FNO exhibits invariance to discretization that is more stable to changing resolution than are the ViT neural operator and the Fourier attention neural operator.

6.2.3 Kolmogorov Flow

We consider the two-dimensional Navier-Stokes equation for a viscous, incompressible fluid,

$$\begin{aligned} \frac{\partial u}{\partial t} + u\cdot\nabla u + \nabla p = \nu\Delta u + \sin(ny)\hat{x}, &\quad (x,t)\in[0,2\pi]^2\times(0,\infty), \\ \nabla\cdot u = 0, &\quad (x,t)\in[0,2\pi]^2\times[0,\infty), \\ u(\cdot,0) = u_0, &\quad x\in[0,2\pi]^2, \end{aligned} \tag{KF}$$

where u denotes the velocity, p the pressure and ν the kinematic viscosity. The particular choice of forcing function $\sin(ny)$ leads to a particular example of a Kolmogorov flow. We equip the domain $[0,2\pi]$ with periodic boundary conditions. We assume

$$u_0 \in \mathcal{U} := \Big\{ u \in \dot{L}^2_{\text{per}}\big([0,2\pi]^2;\mathbb{R}^2\big) : \nabla\cdot u = 0 \Big\}.$$

The dot on L^2 denotes the space of spatially mean-zero functions. The vorticity is defined as $w = (\nabla\times u)\hat{z}$ and the stream function f as the solution to the Poisson equation $-\Delta f = w$. Existence of the semigroup $S_t : \mathcal{U}\to\mathcal{U}$ is shown in Temam (2012, Theorem 2.1). We generate the data by solving (KF) in vorticity-streamfunction form by applying the pseudo-spectral split step method from Chandler and Kerswell (2013). In our experimental set-up $n = 4$ and $\nu = 1/70$ are chosen. Random initial conditions are sampled from the Gaussian measure $\mathcal{N}(0,C)$ where the covariance operator is given by

$$C = 7^3\big(-\Delta + 49I\big)^{-5},$$

where the Laplacian is equipped with periodic boundary conditions on $[0,2\pi]^2$, and viewed as acting between spaces of spatially mean-zero functions. The input output-data pairs for the Kolmogorov flow experiment are given by

$$\Big\{ \omega\big(x,T;\omega_0^{(j)}\big); \omega\big(x,T+\Delta t;\omega_0^{(j)}\big) \Big\}_{j=1}^{J},$$

for $\omega_0^{(j)} \sim \mathcal{N}(0, C)$ being i.i.d. samples. Indeed, given any $r > 0$ we aim to approximate the nonlinear operator $\mathsf{G}^\dagger : H^r\big([0, 2\pi]^2; \mathbb{R}\big) \to H^r\big([0, 2\pi]^2; \mathbb{R}\big)$ defined by

$$\mathsf{G}^\dagger : \omega\big(x, T; \omega_0\big) \mapsto \omega\big(x, T + \Delta t; \omega_0\big).$$

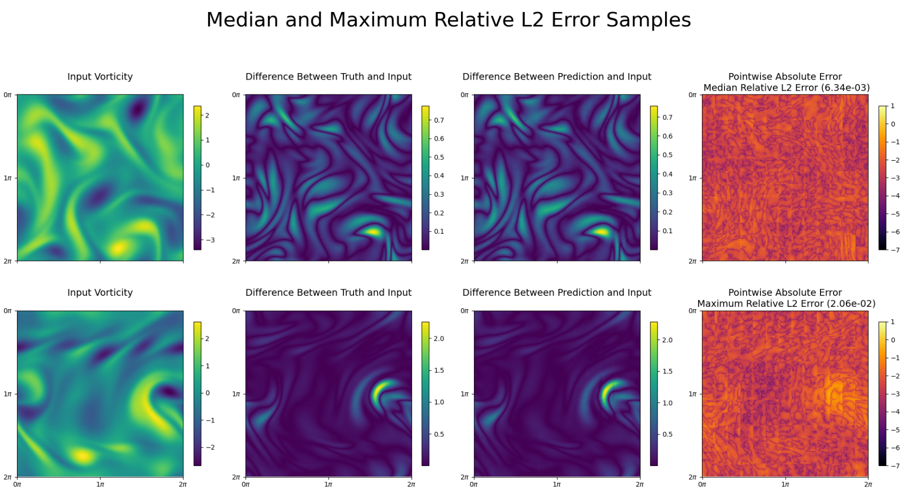

Figure 16: The panel displays the result of the application of the Fourier attention neural operator on the Kolmogorov flow experiment for the median and maximum relative L^2 error samples. The first row displays the sample from the test set of the same resolution as the training set yielding the median relative L^2 error. The second row on the other hand displays the sample yielding the maximum relative L^2 error. The first column of the panel displays the input vorticity $w(x, T)$ of the relevant sample. The second column shows the absolute difference between the true vorticity $w(x, T + \Delta t)$ and the input vorticity $w(x, T)$, while the third column displays the absolute difference between the predicted vorticity at time $T + \Delta t$ and the input $w(x, T)$. The last column shows the pointwise absolute error (in log-scale) between the prediction and truth.

In our experimental setting, we set $T = 11$ and $\Delta t = 0.1$. We train the AFNO, FNO, ViT neural operator and Fourier attention neural operator architectures on this problem using 9000 independent samples of resolution 416×416, 500 samples for validation and 500 samples for testing. The Fourier neural operator is parametrized by 12 Fourier modes in each dimension and channel width of 128; the model is trained using a batch size of 4 and learning rate of 10^{-3}. The AFNO is instantiated with 4 layers of 128 hidden channels, and does not employ patching; it is trained using a learning rate of 10^{-3} and batch size of 1. Both the ViT neural operator and Fourier attention neural operator are parametrized by 128 channels in the transformer encoder. The integral operator in the ViT NO is parametrized by 12 Fourier modes in each dimension while the integral operators appearing in the Fourier attention neural operator are parametrized by 9 Fourier modes in each dimension. Both architectures employ a final spectral convolution layer (Remark 21) that is parametrized by 64 Fourier modes in each dimension. Both of these neural operators are trained using a batch size of 2 and learning rate of 10^{-3}. On this problem we train all the neural operators with the relative H^1 loss. Figure 16 displays the test results using the Fourier attention neural operators.

In Figure 17 we display the results of applying the three architectures to test samples of resolutions different to the 416×416 training resolution. The results demonstrate the zero-shot generalization capability of the three neural operator architectures to different discretizations, which does not

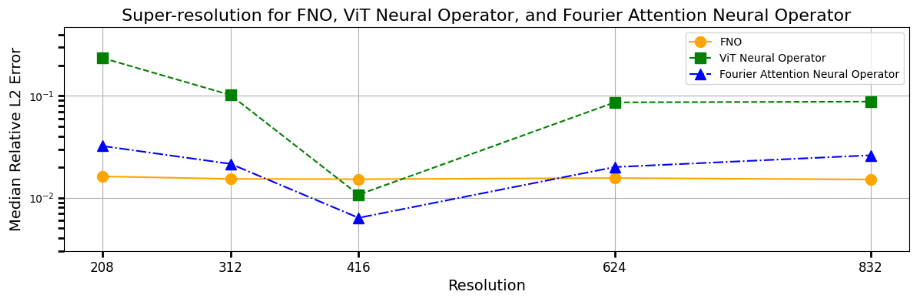

Figure 17: The panel displays the performance in relative L^2 errors (in log-scale) of the FNO, ViT neural operator and Fourier attention neural operator when applied to Kolomgorov flow test sets of different resolutions at inference time (without retraining). All architectures are trained on data that is of resolution 416×416.

require retraining of the models. Again we note that FNO exhibits invariance to discretization that is more stable to changing resolution than the methods introduced here; but for all the neural operators it is nonetheless clear that intrinsic properties of the continuum limit are learned and may be transferred between discretizations.

A few additional considerations are in order. The mode truncation employed in the Fourier neural operator make the architecture computationally efficient but yields an over-smoothing effect that is not suitable for problems at high resolutions, where high frequency detail is prominent. Attention-based neural operators offer a possible solution to this issue, as demonstrated by the performance of the transformer neural operators proposed when applied to the PDE operator learning problems considered, which involve rough diffusion coefficients and rough initial conditions. On the other hand, patching leads to more efficient attention-based neural operators, but also introduces possible discontinuities at patch-intersections. This issue has been observed to be resolved in the large data regime. Furthermore, it is possible to employ problem specific smoothing operators as the one introduced in Remark 21. In Figure 18 we display the result of applying a smoothing operator as the last layer in the training of the FANO architecture in the context of the Kolmogorov flow problem. The smoothing via the inverse of the negative Laplacian is applied using the Fourier basis, given the periodic boundary conditions. Here, we make the specific choice $\epsilon = 10^{-3}$ and $\alpha = 1.001$, in the context of Remark 21. Such an approach is effective at reducing the discontinuities introduced by patching and reducing the evaluation error, but is problem dependent and requires knowledge of boundary conditions. Such discontinuity issues constitute an inherent limitation of patching and highlights the need for more suitable efficient attention mechanisms.

Furthermore, it is observed that in some experimental settings the patching-based neural operators exhibit worse stability to changing resolution than FNO. This is likely due to the effect of patching which may then be further amplified for FANO due to the application of a higher number of FFT(s) to non-periodic domains. In the context of the FANO architecture, such issues can be ameliorated using periodic extensions for Fourier-based parametrizations. In Figure 19 we demonstrate the discretization invariance of the standard FANO compared to a FANO architecture that involves building a periodic extension of each patch. The Fourier integral operator is applied to each extended patch and the extension is then removed from the output. Such a technique leads to an increase in complexity and runtime but yields more stable discretization invariance. Other remedies could potentially include other kinds of parametrizations for the nonlocal operations, or fine-tuning at different resolutions.

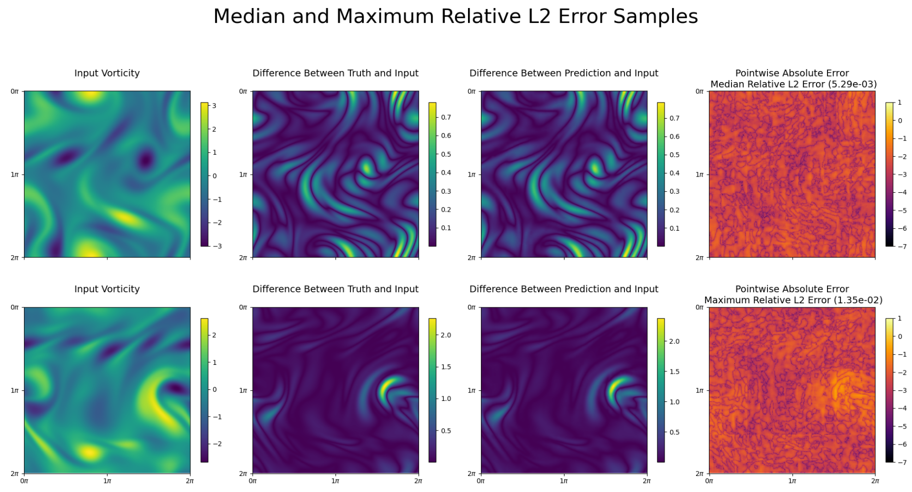

Figure 18: The panel displays the result of the application of the Fourier attention neural operator employing a final smoothing layer on the Kolmogorov flow experiment for the median and maximum relative L^2 error samples. The first row displays the sample from the test set of the same resolution as the training set yielding the median relative L^2 error. The second row on the other hand displays the sample yielding the maximum relative L^2 error. The first column of the panel displays the input vorticity $w(x, T)$ of the relevant sample. The second column shows the absolute difference between the true vorticity $w(x, T + \Delta t)$ and the input vorticity $w(x, T)$, while the third column displays the absolute difference between the predicted vorticity at time $T + \Delta t$ and the input $w(x, T)$. The last column shows the pointwise absolute error (in log-scale) between the prediction and truth.

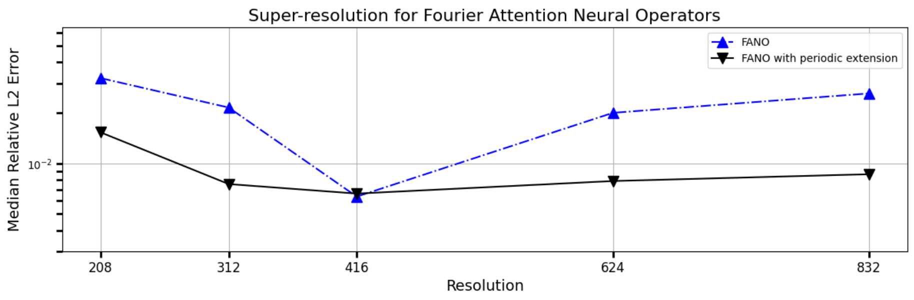

Figure 19: The panel displays the performance in relative L^2 errors (in log-scale) of the Fourier attention neural operator and Fourier attention neural operator with periodic extensions of patches when applied to Kolomgorov flow test sets of different resolutions at inference time (without retraining). All architectures are trained on data that is of resolution 416×416.

7. Conclusions

In this work we have introduced a continuum formulation of the attention mechanism from the seminal work Vaswani et al. (2017). Our continuum formulation can be used to design transformer neural operators. Indeed, this continuum formulation leads to discretization invariant implementations of attention and schemes that exhibit zero-shot generalization to different resolutions. In the first result of its kind for transformers, with a slight modification to the architecture implemented in practice, the resulting neural operator architecture mapping between infinite-dimensional spaces of functions is shown to be a universal approximator of continuous functions and functions of Sobolev regularity defined over a compact domain. We extend the continuum formulation to patched attention, which makes it possible to design efficient transformer neural operators. To this end, we introduce a neural operator analogue of the vision transformer Dosovitskiy et al. (2021) and a more expressive architecture, the Fourier attention neural operator. Through a cost-accuracy analysis, we demonstrate the power of the methodology in the context of a range of operator learning problems. In the following we highlight potential avenues for further work.

1. It is of interest to extend the analysis of Theorems 6 and 12 to quantify the norm-dependent rates of convergence of discretized attention to its continuum limit. Furthermore, obtaining the results over i.i.d. samples from arbitrary strictly positive probability distributions would be of interest.

2. The Fourier attention neural operator architecture uses operator parametrizations for $\mathsf{Q}, \mathsf{K}, \mathsf{V}$. Other design choices for these operators not employing the Fourier basis are possible, for example using the wavelet basis as in Tripura and Chakraborty (2023) or using a multigrid approach as in He et al. (2024). An investigation of other attention neural operator architectures arising from this consideration would be of interest.

3. It is of interest to investigate the discontinuity issues arising from patching. In particular, it is desirable to design schemes that impose continuity amongst these throughout the architecture.

4. Deriving universal approximation theorems for the patch-based transformer neural operators would be of great theoretical interest.

Acknowledgments

AMS is supported by a Department of Defense Vannevar Bush Faculty Fellowship; this funding also supports EC and partly supported MEL. In addition, AMS and EC are supported by the SciAI Center, funded by the Office of Naval Research (ONR), under Grant Number N00014-23-1-2729. NBK is grateful to the Nvidia Corporation for support through full-time employment. MEL is supported by funding from the Eric and Wendy Schmidt Center at the Broad Institute of MIT and Harvard.

We thank Théo Bourdais, Matthieu Darcy, Miguel Liu-Schiaffini, Georgia Gkioxari, Sabera Talukder, Zihui Wu, and Zongyi Li for helpful discussions and feedback.

Appendix A. Proofs of Approximation Theorems

A.1 Proof of Self-Attention Approximation Theorem

We introduce the following auxiliary result that we will apply to prove Theorem 6.

Lemma 25 *Let $D \subset \mathbb{R}^d$ be a bounded open set and let $\{y_j\}_{j=1}^N \sim U(\bar{D})$ be an i.i.d. sequence. If $f \in C(\bar{D} \times \bar{D})$ then, with the expectation taken over the data $\{y_j\}_{j=1}^N$,*

$$\mathbb{E}\left\| \int_D f(\cdot, y)\, \mathrm{d}y - \frac{|D|}{N} \sum_{j=1}^N f(\cdot, y_j) \right\|_{L^2} \le |D|^{3/2} \|f\|_{C(\bar{D}\times\bar{D})} N^{-1/2}.$$

Proof Notice that, for any $x \in \bar{D}$,

$$\int_D f(x,y)\,\mathrm{d}y = |D|\mathbb{E}_{y\sim U(\bar{D})}\big[f(x,y)\big].$$

Hence, by expanding the square, we can observe that

$$\mathbb{E}\left(\int_D f(x,y)\,\mathrm{d}y - \frac{|D|}{N}\sum_{j=1}^{N} f(x,y_j)\right)^2 \le \frac{|D|^2}{N}\mathbb{E}_{y\sim U(\bar{D})}\big[f(x,y)^2\big].$$

Bounding f in L^∞, and noting that it is a continuous function, gives

$$\begin{aligned}\mathbb{E}\left\|\int_D f(\cdot,y)\,\mathrm{d}y - \frac{|D|}{N}\sum_{j=1}^{N} f(\cdot,y_j)\right\|_{L^2}^2 &\le \frac{|D|^2}{N}\int_D \mathbb{E}_{y\sim U(\bar{D})}\big[f(x,y)^2\big]\,\mathrm{d}x\\ &\le |D|^3\|f\|^2_{C(\bar{D}\times\bar{D})}N^{-1}.\end{aligned}$$

The result follows by Jensen's inequality. ■

As part of the proof of Theorem 6, we first show that the the self-attention operator is a mapping of the form $\mathsf{A} : L^\infty(D;\mathbb{R}^{d_u}) \to L^\infty(D;\mathbb{R}^{d_V})$ and hence $\mathsf{A} : C(\bar{D};\mathbb{R}^{d_u}) \to C(\bar{D};\mathbb{R}^{d_V})$.

Lemma 26 *Let $u \in L^\infty(D;\mathbb{R}^{d_u})$, then it holds that $\mathsf{A}(u) \in L^\infty(D;\mathbb{R}^{d_V})$. Furthermore, for $u \in C(\bar{D};\mathbb{R}^{d_u})$ it holds that $\mathsf{A}(u) \in C(\bar{D};\mathbb{R}^{d_V})$.*

Proof The result follows readily from the fact that p is a probability density function. For completeness, we show the boundedness of the normalization constant of the function p. Indeed we note that,

$$\begin{aligned}\int_D \exp\big(\langle Qu(x), Ku(s)\rangle\big)\,\mathrm{d}s &\le \int_D \exp\big(|Q||K||u(x)||u(y)|\big)\mathrm{d}s\\ &\le \int_D \exp\left(\frac{1}{2}|Q|^2|K|^2|u(x)|^2\right)\exp\left(\frac{1}{2}|u(y)|^2\right)\mathrm{d}s\\ &\le |D|\exp\big(R\|u\|^2_{L^\infty}\big)\end{aligned}$$

for some constant $R > 0$. From definition, it is clear that

$$\int_D p(y;u,x)\,\mathrm{d}y = 1,$$

hence p is indeed a valid probability density function. It hence follows that

$$\|A(u)\|_{L^\infty} \le |V|\|u\|_{L^\infty}$$

hence $A : L^\infty(D;\mathbb{R}^{d_u}) \to L^\infty(D;\mathbb{R}^{d_V})$. Similarly, it follows that $A : C(\bar{D};\mathbb{R}^{d_u}) \to C(\bar{D};\mathbb{R}^{d_V})$. We note that the continuity of u is preserved by A under the continuity of the inner product in its first argument and the continuity of the exponential. ■

We are now ready to establish the full result of Theorem 6 which states that

$$\lim_{N\to\infty}\sup_{u\in B}\mathbb{E}\left\|\mathsf{A}(u) - \frac{\sum_{j=1}^{N}\exp\big(\langle Qu(\cdot), Ku(y_j)\rangle\big)Vu(y_j)}{\sum_{\ell=1}^{N}\exp\big(\langle Qu(\cdot), Ku(y_\ell)\rangle\big)}\right\|_{C(\bar{D};\mathbb{R}^{d_V})} = 0,$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^N \sim \mathsf{unif}(\bar{D})$.

Proof [Proof of Theorem 6] Before commencing the proof, we first establish useful shorthand notation. Letting $u \in B$ we define,

$$f(x,y;u) \coloneqq \exp\bigl(\langle Qu(x), Ku(y)\rangle\bigr), \quad g_l(x,y;u) \coloneqq \exp\bigl(\langle Qu(x), Ku(y)\rangle\bigr)\bigl(Vu(y)\bigr)_l$$

for all $x, y \in \bar{D}$ and $l \in [\![1, d_V]\!]$. For $u \in B$ we also define

$$a(x;u) \coloneqq \int_D f(x,y;u)\,\mathrm{d}y, \qquad a^{(N)}(x;u) \coloneqq \frac{|D|}{N}\sum_{j=1}^{N} f(x,y_j;u),$$

and

$$b_l(x;u) \coloneqq \int_D g_l(x,y;u)\,\mathrm{d}y, \qquad b_l^{(N)}(x;u) \coloneqq \frac{|D|}{N}\sum_{j=1}^{N} g_l(x,y_j;u)$$

for any $x \in \bar{D}$ and any $l \in [\![1, d_V]\!]$. We set $\alpha_l = b_l/a$ and $\alpha_l^{(N)} = b_l^{(N)}/a^{(N)}$. Given this notation, to prove the result we must show

$$\lim_{N\to\infty}\sup_{u\in B}\mathbb{E}\left\|\alpha_l(\cdot\,;u) - \alpha_l^{(N)}(\cdot\,;u)\right\|_{C(\bar{D};\mathbb{R})} = 0, \tag{43}$$

for any $l \in [\![1, d_V]\!]$. We divide the proof in the following key steps. We first show that for fixed $u \in B$, it holds that

$$\lim_{N\to\infty}\mathbb{E}\|\alpha_l(\cdot\,;u) - \alpha_l^{(N)}(\cdot\,;u)\|_{L^2} = 0, \tag{44}$$

for any $l \in [\![1, d_V]\!]$. By using compactness and applying the Arzelà-Ascoli theorem, we then show that

$$\lim_{N\to\infty}\mathbb{E}\|\alpha_l(\cdot\,;u) - \alpha_l^{(N)}(\cdot\,;u)\|_{C(\bar{D})} = 0, \tag{45}$$

for any $l \in [\![1, d_V]\!]$. We then use the compactness of B and continuity of the mappings $u \mapsto \alpha_\ell(\cdot\,;u)$ and $u \mapsto \alpha_\ell^{(N)}(\cdot\,;u)$ to deduce the result given by (43). We now focus on establishing (44). Since B is bounded, there exists a constant $M > 0$ such that,

$$\sup_{u\in B}\|u\|_{L^\infty} \le M.$$

Therefore, we have

$$\sup_{u\in B}\max\{\|f\|, \|g_1\|, \dots, \|g_m\|\} \le \max\left\{\exp\bigl(RM^2\bigr), M|V|\exp\bigl(RM^2\bigr)\right\} \coloneqq J,$$

where $\|\cdot\| = \|\cdot\|_{C(\bar{D}\times\bar{D})}$. Clearly,

$$\sup_{u\in B}\max\{\|a\|_{L^2}, \|a^{(N)}\|_{L^2}\} \le |D|^{3/2}J$$

Notice that, since $|\langle Qu(x), Ku(s)\rangle| \le RM^2$, we have $f(x,y) \ge \exp\bigl(-RM^2\bigr)$. It follows that

$$\inf_{u\in B}\min\{\|a\|_{L^2}, \|a^{(N)}\|_{L^2}\} \ge |D|^{3/2}\exp\bigl(-RM^2\bigr) \coloneqq |D|^{3/2}I > 0.$$

Similarly,

$$\sup_{u\in B}\max_{l\in[\![1,d_V]\!]}\{\|b_l\|_{L^2}, \|b_l^{(N)}\|_{L^2}\} \le |D|^{3/2}J.$$

Applying Lemma 25, for fixed $u \in B$ we find that

$$\mathbb{E}\left\|\frac{b_l}{a} - \frac{b_l^{(N)}}{a^{(N)}}\right\|_{L^2} \leq \mathbb{E}\frac{\|a^{(N)}\|_{L^2}\|b_l - b_l^{(N)}\|_{L^2} + \|b_l^{(N)}\|_{L^2}\|a - a^{(N)}\|_{L^2}}{\|a^{(N)}\|_{L^2}\|a\|_{L^2}} \tag{46a}$$

$$\leq \frac{2J^2}{I^2}N^{-1/2}. \tag{46b}$$

Setting $\alpha_l = b_l/a$ and $\alpha_l^{(N)} = b_l^{(N)}/a^{(N)}$, from (46a) we deduce that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,; u) - \alpha_l^{(N)}(\cdot\,; u)\|_{L^2} = 0, \tag{47}$$

for any $l \in [\![1, d_V]\!]$. We now proceed to the second step of the proof, i.e. showing that (45) holds. Using similar reasoning as before, we notice that

$$\inf_{u\in B} \min\{\|a\|_{C(\bar{D})}, \|a^{(N)}\|_{C(\bar{D})}\} \geq |D|I, \tag{48a}$$

$$\sup_{u\in B} \max_{l\in[\![1,d_V]\!]}\{\|b_l\|_{C(\bar{D})}, \|b_l^{(N)}\|_{C(\bar{D})}\} \leq |D|J, \tag{48b}$$

hence the sequence $\{\alpha_l^{(N)}\}$ is uniformly bounded in N. Now, for fixed $u \in B$ the sequence $\{\alpha_l^{(N)}(\cdot\,; u)\}$ is also uniformly equicontinuous with probability 1 over the choice $\{y_j\}_{j=1}^N \sim U(\bar{D})$. Indeed, we note that

$$\left|\frac{b_l^{(N)}(r)}{a^{(N)}(r)} - \frac{b_l^{(N)}(t)}{a^{(N)}(t)}\right| = \left|\frac{\sum_{j=1}^N g_l(r, y_j)}{\sum_{k=1}^N f(r, y_k)} - \frac{\sum_{j=1}^N g_l(t, y_j)}{\sum_{k=1}^N f(t, y_k)}\right| \tag{49a}$$

$$\leq \frac{1}{N^2 I}\left|\sum_{j,n=1}^N g_l(r, y_j)f(t, y_n) - g_l(t, y_j)f(r, y_n)\right| \tag{49b}$$

$$\leq \frac{J}{N^2 I}\sum_{j,n=1}^N \big(\left|g_l(r, y_j) - g_l(t, y_j)\right| + \left|f(t, y_n) - f(r, y_n)\right|\big). \tag{49c}$$

The result then follows from the uniform continuity of f and g_l in their first arguments, due to the compactness of $\bar{D}$. Therefore, since $\bar{D} \subset \mathbb{R}^d$ is compact, by the Arzelà–Ascoli theorem there exists a subsequence of indices $(N_k)_{k\in\mathbb{N}}$ such that $\alpha_l^{(N_k)}$ converges in $C(\bar{D})$ to some $\alpha_l^{(\infty)}$. Since,

$$\mathbb{E}\|\alpha_l^{(\infty)} - \alpha_l^{(N_k)}\|_{L^2} \leq |D|^{1/2}\mathbb{E}\|\alpha_l^{(\infty)} - \alpha_l^{(N_k)}\|_{C(\bar{D})},$$

by uniqueness of limits, it is readily observed that $\alpha_l^{(\infty)} = \alpha_l$. Therefore, since the limit is independent of the subsequence, it holds that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,; u) - \alpha_l^{(N)}(\cdot\,; u)\|_{C(\bar{D})} = 0,$$

for any $l \in [\![1, d_V]\!]$. We now turn our attention to establishing (43). By reasoning as before and by using the continuity of the exponential and the inner product in its first argument, it is straightforward to show that as mappings of the form $\alpha_l : B \to C(\bar{D}; \mathbb{R}^{d_V})$ and $\alpha_l^{(N)} : B \to C(\bar{D}; \mathbb{R}^{d_V})$ where B is compact, α_l and $\alpha_l^{(N)}$ are continuous and hence the sequence $\{\alpha_l^{(N)}\}$ is uniformly equicontinuous. Therefore, we can find moduli of continuity ω_1, ω_2 such that

$$\|\alpha_l(u) - \alpha_l(v)\|_{C(\bar{D})} \leq \omega_1\big(\|u - v\|_{C(\bar{D})}\big), \quad \|\alpha_l^{(N)}(u) - \alpha_l^{(N)}(v)\|_{C(\bar{D})} \leq \omega_2\big(\|u - v\|_{C(\bar{D})}\big)$$

for any $u, v \in B$ and $N \in \mathbb{N}$. Fix $\epsilon > 0$. Since B is compact, we can find a number $L = L(\epsilon) \in \mathbb{N}$ and functions $\phi_1, \dots, \phi_L \in B$ such that, for any $u \in B$, there exists $j \in [\![1, L]\!]$ such that

$$\|u - \phi_j\|_{C(\bar{D})} < \epsilon.$$

Furthermore, we can find a number $S = S(\epsilon) \in \mathbb{N}$ such that, for any $j \in [\![1, L]\!]$, we have, for all $N \geq S$,

$$\mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(\phi_j)\|_{C(\bar{D})} < \epsilon.$$

It follows by triangle inequality that

$$\begin{aligned}\mathbb{E}\|\alpha_l(u) - \alpha_l^{(N)}(u)\|_{C(\bar{D})} &\leq \mathbb{E}\|\alpha_l(u) - \alpha_l(\phi_j)\|_{C(\bar{D})} + \mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(u)\|_{C(\bar{D})} \\ &\leq |D|\omega_1(\epsilon) + \mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(\phi_j)\|_{C(\bar{D})} + \mathbb{E}\|\alpha_l^{(N)}(\phi_j) - \alpha_l^{(N)}(u)\|_{C(\bar{D})} \\ &\leq |D|\omega_1(\epsilon) + \epsilon + |D|\omega_2(\epsilon).\end{aligned}$$

Therefore the result follows since the argument is uniform over all $l \in [\![1, d_V]\!]$. ■

A.2 Proof of Cross-Attention Approximation Theorem

We make straightforward modifications to Lemmas 25, 28 and to the proof of Theorem 6 in order to establish Theorem 12.

Lemma 27 *Let $D \subset \mathbb{R}^d$ and $E \subset \mathbb{R}^e$ be bounded open sets and $f \in C(\bar{D} \times \bar{E})$. Then*

$$\mathbb{E}\left\|\int_E f(\cdot, y)\,\mathrm{d}y - \frac{|E|}{N}\sum_{j=1}^N f(\cdot, y_j)\right\|_{L^2} \leq |D|^{1/2}|E|\|f\|_{C(\bar{D}\times\bar{E})}N^{-1/2}$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^N \sim U(\bar{E})$.

Proof Notice that, for any $x \in \bar{D}$,

$$\int_E f(x, y)\,\mathrm{d}y = |E|\mathbb{E}_{y\sim U(\bar{E})}\big[f(x, y)\big].$$

Hence, by expanding the square, we can observe that

$$\mathbb{E}\left(\int_E f(x, y)\,\mathrm{d}y - \frac{|E|}{N}\sum_{j=1}^N f(x, y_j)\right)^2 \leq \frac{|E|^2}{N}\mathbb{E}_{y\sim U(\bar{E})}\big[f(x, y)^2\big].$$

Bounding f in L^∞, and noting that it is a continuous function, gives

$$\begin{aligned}\mathbb{E}\left\|\int_E f(\cdot, y)\,\mathrm{d}y - \frac{|E|}{N}\sum_{j=1}^N f(\cdot, y_j)\right\|_{L^2}^2 &\leq \frac{|E|^2}{N}\int_D \mathbb{E}_{y\sim U(\bar{E})}\big[f(x, y)^2\big]\,\mathrm{d}x \\ &\leq |D||E|^2\|f\|_{C(\bar{D}\times\bar{E})}^2 N^{-1}.\end{aligned}$$

The result follows by Jensen's inequality. ■

As part of the proof of Theorem 12, we first show that the the cross-attention operator is a mapping of the form $\mathsf{C} : L^\infty(D; \mathbb{R}^{d_u}) \times L^\infty(E; \mathbb{R}^{d_v}) \to L^\infty(D; \mathbb{R}^{d_V})$ and hence $\mathsf{A} : C(\bar{D}; \mathbb{R}^{d_u}) \times C(\bar{E}; \mathbb{R}^{d_v}) \to C(\bar{D}; \mathbb{R}^{d_V})$.

Lemma 28 *Let $u \in L^\infty(D;\mathbb{R}^{d_u}) \times L^\infty(E;\mathbb{R}^{d_v})$, then it holds that $\mathsf{A}(u) \in L^\infty(D;\mathbb{R}^{d_V})$. Furthermore, for $u \in C(\bar{D};\mathbb{R}^{d_u}) \times C(\bar{E};\mathbb{R}^{d_v})$ it holds that $\mathsf{A}(u) \in C(\bar{D};\mathbb{R}^{d_V})$.*

Proof The result follows readily from the fact that q is a probability density function. For completeness, we show the boundedness of the normalization constant of the function q. Indeed we note that,

$$\begin{aligned}\int_E \exp\big(\langle Qu(x), Kv(s)\rangle\big)\,\mathrm{d}s &\le \int_E \exp\big(|Q||K||u(x)||v(s)|\big)\mathrm{d}s \\ &\le \int_E \exp\Big(\frac{1}{2}|Q|^2|K|^2|u(x)|^2\Big)\exp\Big(\frac{1}{2}|v(s)|^2\Big)\mathrm{d}s \\ &\le |E|\exp\big(R_1\|u\|^2_{L^\infty}\big)\exp\big(R_2\|v\|^2_{L^\infty}\big) \\ &\le |E|\exp\Big(R\big(\|u\|^2_{L^\infty} + \|v\|^2_{L^\infty}\big)\Big),\end{aligned}$$

for some constant $R \coloneqq \max\{R_1, R_2\}$. From definition, it is clear that

$$\int_E q(y;u,v,x)\,\mathrm{d}y = 1,$$

hence q is indeed a valid probability density function. It hence follows that

$$\|\mathsf{C}(u,v)\|_{L^\infty(D;\mathbb{R}^{d_V})} \le |V|\|v\|_{L^\infty(E;\mathbb{R}^{d_v})}$$

hence $\mathsf{C} : L^\infty(D;\mathbb{R}^{d_u}) \times L^\infty(E;\mathbb{R}^{d_v}) \to L^\infty(D;\mathbb{R}^{d_V})$. Similarly, it follows that $\mathsf{C} : C(\bar{D};\mathbb{R}^{d_u}) \times C(\bar{E};\mathbb{R}^{d_v}) \to C(\bar{D};\mathbb{R}^{d_V})$. We note that the continuity is preserved by C under the continuity of the inner product in its first argument and the continuity of the exponential. ■

We are now ready to establish the full result of Theorem 12 which states that

$$\lim_{N\to\infty} \sup_{(u,v)\in B} \mathbb{E}\left\|\mathsf{C}(u,v) - \frac{\sum_{j=1}^N \exp\big(\langle Qu(\cdot), Kv(y_j)\rangle\big)Vv(y_j)}{\sum_{\ell=1}^N \exp\big(\langle Qu(\cdot), Kv(y_\ell)\rangle\big)}\right\|_{C(\bar{D};\mathbb{R}^{d_V})} = 0,$$

with the expectation taken over i.i.d. sequences $\{y_j\}_{j=1}^N \sim \mathsf{unif}(\bar{E})$.

Proof [Proof of Theorem 12] Before commencing the proof, we first establish useful shorthand notation. Letting $(u,v) \in B$ we define,

$$f(x,y;u,v) \coloneqq \exp\big(\langle Qu(x), Kv(y)\rangle\big), \quad g_l(x,y;u,v) \coloneqq \exp\big(\langle Qu(x), Kv(y)\rangle\big)\big(Vv(y)\big)_l$$

for all $x \in \bar{D}, y \in \bar{E}$ and $l \in [\![1, d_V]\!]$. For $(u,v) \in B$ we also define

$$a(x;u,v) \coloneqq \int_E f(x,y;u,v)\,\mathrm{d}y, \qquad a^{(N)}(x;u,v) \coloneqq \frac{|E|}{N}\sum_{j=1}^N f(x,y_j;u,v),$$

and

$$b_l(x;u,v) \coloneqq \int_E g_l(x,y;u,v)\,\mathrm{d}y, \qquad b_l^{(N)}(x;u,v) \coloneqq \frac{|E|}{N}\sum_{j=1}^N g_l(x,y_j;u,v)$$

for any $x \in \bar{D}$ and any $l \in [\![1, d_V]\!]$. We set $\alpha_l = b_l/a$ and $\alpha_l^{(N)} = b_l^{(N)}/a^{(N)}$. Given this notation, to prove the result we must show

$$\lim_{N\to\infty} \sup_{(u,v)\in B} \mathbb{E}\left\|\alpha_l(\cdot\,;u,v) - \alpha_l^{(N)}(\cdot\,;u,v)\right\|_{C(\bar{D};\mathbb{R})} = 0, \tag{50}$$

for any $l \in [\![1, d_V]\!]$. We divide the proof in the following key steps. We first show that for fixed $(u,v) \in B$, it holds that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,;u,v) - \alpha_l^{(N)}(\cdot\,;u,v)\|_{L^2} = 0, \tag{51}$$

for any $l \in [\![1, d_V]\!]$. By using compactness and applying the Arzelà-Ascoli theorem, we then show that for $(u,v) \in B$

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,;u,v) - \alpha_l^{(N)}(\cdot\,;u,v)\|_{C(\bar{D})} = 0, \tag{52}$$

for any $l \in [\![1, d_V]\!]$. We then use the compactness of B and continuity of the mappings $(u,v) \mapsto \alpha_\ell(\cdot\,;u,v)$ and $(u,v) \mapsto \alpha_\ell^{(N)}(\cdot\,;u,v)$ to deduce the result given by (50). We now focus on establishing (51). Since B is bounded, there exists a constant $M > 0$ such that,

$$\sup_{(u,v)\in B} \|u\|_{L^\infty} + \|v\|_{L^\infty} \le M.$$

Therefore, we have

$$\sup_{(u,v)\in B} \max\{\|f\|, \|g_1\|, \dots, \|g_m\|\} \le \max\big\{\exp\big(RM^2\big), M|V|\exp\big(RM^2\big)\big\} := J,$$

where $\|\cdot\| = \|\cdot\|_{C(\bar{D}\times\bar{E})}$. Clearly, it holds that

$$\sup_{(u,v)\in B} \max\{\|a\|_{L^2}, \|a^{(N)}\|_{L^2}\} \le |D|^{1/2}|E|J.$$

Notice that, since $|\langle Qu(x), Kv(s)\rangle| \le RM^2$, we have $f(x,y) \ge \exp\big(-RM^2\big)$. It follows that

$$\inf_{(u,v)\in B} \min\{\|a\|_{L^2}, \|a^{(N)}\|_{L^2}\} \ge |D|^{1/2}|E|\exp\big(-RM^2\big) := |D|^{1/2}|E|I > 0.$$

Similarly,

$$\sup_{(u,v)\in B} \max_{l\in[d_K]}\{\|b_l\|_{L^2}, \|b_l^{(N)}\|_{L^2}\} \le |D|^{1/2}|E|J.$$

Applying Lemma 27, we find

$$\mathbb{E}\left\|\frac{b_l}{a} - \frac{b_l^{(N)}}{a^{(N)}}\right\|_{L^2} \le \mathbb{E}\frac{\|a^{(N)}\|_{L^2}\|b_l - b_l^{(N)}\|_{L^2} + \|b_l^{(N)}\|_{L^2}\|a - a^{(N)}\|_{L^2}}{\|a^{(N)}\|_{L^2}\|a\|_{L^2}} \tag{53a}$$

$$\le \frac{2J^2}{I^2}N^{-1/2}. \tag{53b}$$

Setting $\alpha_l = b_l/a$ and $\alpha_l^{(N)} = b_l^{(N)}/a^{(N)}$, from (53a) we deduce that for $(u,v) \in B$ it holds that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,;u,v) - \alpha_l^{(N)}(\cdot\,;u,v)\|_{L^2} = 0, \tag{54}$$

for any $l \in [\![1, d_V]\!]$. We now proceed to the second step of the proof, i.e. showing that (52) holds. Using similar reasoning as before, we notice that

$$\inf_{(u,v)\in B} \min\{\|a\|_{C(\bar{D})}, \|a^{(N)}\|_{C(\bar{D})}\} \ge |E|I, \tag{55a}$$

$$\sup_{(u,v)\in B} \max_{l\in[m]}\{\|b_l\|_{C(\bar{D})}, \|b_l^{(N)}\|_{C(\bar{D})}\} \le |E|J, \tag{55b}$$

hence the sequence $\{\alpha_l^{(N)}\}$ is uniformly bounded in N. Now, for fixed $(u,v) \in B$ the sequence $\{\alpha_l^{(N)}(\cdot\,; u,v)\}$ is also uniformly equicontinuous with probability 1 over the choice $\{y_j\}_{j=1}^N \sim U(\bar{D})$. Indeed, we note that

$$\begin{aligned}
\left|\frac{b_l^{(N)}(r)}{a^{(N)}(r)} - \frac{b_l^{(N)}(t)}{a^{(N)}(t)}\right| &= \left|\frac{\sum_{j=1}^N g_l(r,y_j)}{\sum_{k=1}^N f(r,y_k)} - \frac{\sum_{j=1}^N g_l(t,y_j)}{\sum_{k=1}^N f(t,y_k)}\right| && \text{(56a)}\\
&\leq \frac{1}{N^2 I}\left|\sum_{j,n} g_l(r,y_j)f(t,y_n) - g_l(t,y_j)f(r,y_n)\right| && \text{(56b)}\\
&\leq \frac{J}{N^2 I}\sum_{j,n}\big(|g_l(r,y_j) - g_l(t,y_j)| + |f(t,y_n) - f(r,y_n)|\big). && \text{(56c)}
\end{aligned}$$

The result then follows from the uniform continuity of f and g_l in their first arguments, due to the compactness of $\bar{D}$. Therefore, since $\bar{D} \subset \mathbb{R}^d$ is compact, by the Arzelà–Ascoli theorem there exists a subsequence of indices $(N_k)_{k\in\mathbb{N}}$ such that $\alpha_l^{(N_k)}$ converges in $C(\bar{D})$ to some $\alpha_l^{(\infty)}$. Since,

$$\mathbb{E}\|\alpha_l^{(\infty)} - \alpha_l^{(N_k)}\|_{L^2} \leq |D|^{1/2}\mathbb{E}\|\alpha_l^{(\infty)} - \alpha_l^{(N_k)}\|_{C(\bar{D})},$$

by uniqueness of limits, it is readily observed that $\alpha_l^{(\infty)} = \alpha_l$. Therefore, since the limit is independent of the subsequence, for fixed $(u,v) \in B$ it holds that

$$\lim_{N\to\infty} \mathbb{E}\|\alpha_l(\cdot\,; u,v) - \alpha_l^{(N)}(\cdot\,; u,v)\|_{C(\bar{D})} = 0,$$

for any $l \in [\![1, d_V]\!]$. We now turn our attention to establishing (50). By reasoning as before and by using the continuity of the exponential and of the inner product on the product space, it is straightforward to show that as mappings of the form $\alpha_l : B \to C(\bar{D}; \mathbb{R}^{d_V})$ and $\alpha_l^{(N)} : B \to C(\bar{D}; \mathbb{R}^{d_V})$ where B is compact, α_l and $\alpha_l^{(N)}$ are continuous and hence the sequence $\{\alpha_l^{(N)}\}$ is uniformly equicontinuous. Therefore, we can find moduli of continuity ω_1, ω_2 such that

$$\begin{aligned}
\|\alpha_l(u,v) - \alpha_l(\widetilde{u},\widetilde{v})\|_{C(\bar{D})} &\leq \omega_1\big(\|(u,v) - (\widetilde{u},\widetilde{v})\|_{C(\bar{D})\times C(\bar{E})}\big),\\
\|\alpha_l^{(N)}(u,v) - \alpha_l^{(N)}(\widetilde{u},\widetilde{v})\|_{C(\bar{D})} &\leq \omega_2\big(\|(u,v) - (\widetilde{u},\widetilde{v})\|_{C(\bar{D})\times C(\bar{E})}\big)
\end{aligned}$$

for any $(u,v), (\widetilde{u},\widetilde{v}) \in B$ and $N \in \mathbb{N}$. Fix $\epsilon > 0$. Since B is compact, we can find a number $L = L(\epsilon) \in \mathbb{N}$ and functions $\phi_1, \dots, \phi_L \in B$ such that, for any $(u,v) \in B$, there exists $j \in [\![1, L]\!]$ such that

$$\|(u,v) - \phi_j\|_{C(\bar{D})\times C(\bar{E})} < \epsilon.$$

Furthermore, we can find a number $S = S(\epsilon) \in \mathbb{N}$ such that, for any $j \in [\![1, L]\!]$, we have, for all $N \geq S$,

$$\mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(\phi_j)\|_{C(\bar{D})} < \epsilon.$$

It follows by triangle inequality that

$$\begin{aligned}
\mathbb{E}\|\alpha_l(u,v) - \alpha_l^{(N)}(u,v)\|_{C(\bar{D})} &\leq \mathbb{E}\|\alpha_l(u,v) - \alpha_l(\phi_j)\|_{C(\bar{D})} + \mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(u,v)\|_{C(\bar{D})}\\
&\leq |D|\omega_1(\epsilon) + \mathbb{E}\|\alpha_l(\phi_j) - \alpha_l^{(N)}(\phi_j)\|_{C(\bar{D})} + \mathbb{E}\|\alpha_l^{(N)}(\phi_j) - \alpha_l^{(N)}(u,v)\|_{C(\bar{D})}\\
&\leq |E|\omega_1(\epsilon) + \epsilon + |E|\omega_2(\epsilon).
\end{aligned}$$

Therefore the result follows since our argument is uniform over all $l \in [\![1, d_V]\!]$. ■

Appendix B. Transformer Neural Operators: The Discrete Setting

We describe how the transformer neural operators described in Section 4 are implemented in the discrete setting. Namely, in practice we will have access to evaluations of the input function $u \in \mathcal{U}\big(D;\mathbb{R}^{d_u}\big)$ at a finite set of N discretization points $\{x_i\}_{i=1}^N \subset D$. To highlight how the neural operator methodology is applied in practical settings it is useful, abusing notation, to view the input to the neural operators as $u \in \mathbb{R}^{N\times d_u}$. In order to distinguish between functions in $\mathcal{U}\big(D;\mathbb{R}^{d_u}\big)$, and discrete representations of these functions at a finite set of grid points, in this section we use different fonts for a function $\mathsf{u} \in \mathcal{U}\big(D;\mathbb{R}^{d_u}\big)$ and its discrete analogue $u \in \mathbb{R}^{N\times d_u}$. We highlight that in practice, it is of interest to apply approximations of the attention operators A and A_P defined on the continuum, to $u \in \mathbb{R}^{N\times d_u}$. Through an abuse of notation, we will write $\mathsf{A}_\bullet(u)$ to denote the finite-dimensional approximation of $\mathsf{A}_\bullet$ acting on matrices, and similarly for $\mathsf{E}_L, \mathsf{F}_{\mathrm{NN}}$ and F_{LN}.

In Subsection B.1 we describe the transformer neural operator in the discrete setting, in Subsection B.2 the vision transformer neural operator and in Subsection B.3 the Fourier attention neural operator.

B.1 Transformer Neural Operator

In this section, we describe how the transformer neural operator from Subsection 4.2 is implemented in the discrete setting. The input to the model is therefore a tensor $u \in \mathbb{R}^{N\times d_u}$, which is then concatenated with the discretization points $\{x_i\}_{i=1}^N \subset D$, to form the tensor $u_{\mathrm{in}} \in \mathbb{R}^{N\times(d_u+d)}$. The learnable linear transformation $W_{\mathrm{in}}^\top \in \mathbb{R}^{(d_u+d)\times d_{\mathrm{model}}}$ then acts on u_{in} to yield

$$v^{(0)} := u_{\mathrm{in}} W_{\mathrm{in}}^\top,$$

which is then fed through the transformer encoder. In practice we solve the recurrence relation in (10) on a finite set of N grid points that arise naturally from the discrete input u_{in}. The multi-head attention operator $\mathsf{A}_{\mathrm{MultiHead}}$ is defined by the application of the linear transformation $W_{\mathrm{MultiHead}}^\top \in \mathbb{R}^{Hd_K\times d_{\mathrm{model}}}$ to the concatenation of $H \in \mathbb{N}$ self-attention operations. The concatenation of the outputs of the $H \in \mathbb{N}$ self-attention operations results in a $\mathbb{R}^{N\times Hd_K}$ tensor, so that

$$\mathsf{A}_{\mathrm{MultiHead}}(v) := \big(\mathsf{A}(v;Q_1,K_1,V_1),\dots,\mathsf{A}(v;Q_H,K_H,V_H)\big)W_{\mathrm{MultiHead}}^\top \in \mathbb{R}^{N\times d_{\mathrm{model}}}, \tag{57}$$

for any $v \in \mathbb{R}^{N\times d_{\mathrm{model}}}$. We recall that the self-attention operator in question, A, is defined in Definition 5 for functions. In the discrete setting, where $v \in \mathbb{R}^{N\times d_{\mathrm{model}}}$, we compute the expectation and the integral arising from the normalization constant using the following trapezoidal approximations:

$$\mathsf{A}(v)_j := \frac{1}{2}\sum_{k=2}^{N}\Big(Vv_k\cdot p(k;v,j) + Vv_{k-1}\cdot p(k-1;v,j)\Big)\cdot\Delta x_k, \tag{58}$$

for any $j = 1,\dots,N$, with

$$p(k;v,j) \approx \frac{\exp\Big(\big\langle Qv_j, Kv_k\big\rangle_{\mathbb{R}^{d_K}}\Big)}{\frac{1}{2}\sum_{\ell=2}^{N}\Big(\exp\big(\langle Qv_j,Kv_\ell\rangle_{\mathbb{R}^{d_K}}\big)+\exp\big(\langle Qv_j,Kv_{\ell-1}\rangle_{\mathbb{R}^{d_K}}\big)\Big)\Delta x_\ell}, \tag{59}$$

where by u_ℓ we have denoted the ℓ'th row of the matrix $v \in \mathbb{R}^{N\times d_{\mathrm{model}}}$, and similarly for $\mathsf{A}(v)_\ell$. In equations (58) and (59) Δx_k denotes

$$\Delta x_k := \prod_{i=1}^{d} |(x_k)_i - (x_{k-1})_i|,$$

for $k = 2,\dots,N$, where $(x_k)_i$ is the i'th component of the vector x_k. This trapezoidal approximation allows for a mathematically consistent implementation of attention for irregularly sampled data, as demonstrated in Section 6.

The linear transformations $W_1^\top, W_2^\top \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$ are applied pointwise. It is also straightforward to apply the definitions for $\mathsf{F}_{\text{LayerNorm}}$ in (16) and F_{NN} in (18) to the finite representation of v as both involve pointwise linear transformations. We observe that the recurrence relation (10) is compatible with varying domain discretization lengths N, and will produce an output $v^{(L)}$ of dimension matching the input $v^{(0)}$. The resulting tensor $v^{(L)} \in \mathbb{R}^{N\times d_{\text{model}}}$ is then projected to the output dimension by applying the learnable linear transformation $W_{\text{out}}^\top \in \mathbb{R}^{d_{\text{model}}\times d_z}$, so that the output of the architecture is defined as

$$z \coloneqq v^{(L)} W_{\text{out}}^\top \in \mathbb{R}^{d_{\text{model}}\times d_z}.$$

B.2 Vision Transformer Neural Operator

In this section, we describe how the transformer neural operator from Subsection 4.3 is implemented in the discrete setting. In practice, we will have access to a discretization of the domain D containing N points and evaluations of the function $\mathsf{u} \in \mathcal{U}(D;\mathbb{R}^{d_u})$ at these points; the resolution of the input will be determined by $n_1 \times \ldots \times n_d = N$. The input to the model will therefore be a tensor $u \in \mathbb{R}^{n_1\times\ldots\times n_d\times d_u}$. The input $u \in \mathbb{R}^{n_1\times\ldots\times n_d\times d_u}$ is first concatenated with the discretization points $\{x_i\}_{i=1}^N \subset D$, to form the tensor $u_{\text{in}} \in \mathbb{R}^{n_1\times\ldots\times n_d\times(d_u+d)}$. Denoting by $p_1\times\ldots\times p_d$ the resolution of each patch, application of the reshaping operator $\mathsf{S}_{\text{reshape}}$ results in

$$\mathsf{S}_{\text{reshape}}\big(u_{\text{in}}\big) \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times(d_u+d)}.$$

Following Definition 20 and the subsequent discussion, the nonlocal linear operator W_{in} is implemented as

$$\mathsf{W}_{\text{in}}(\theta)u \coloneqq \mathcal{F}^{-1}\Big(R_{\mathsf{W}_{\text{in}}}(\theta)\cdot(\mathcal{F}u)\Big), \tag{60}$$

for any $u \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times(d_u+d)}$, where we have made explicit the dependence on $\theta\in\Theta$ to highlight the learnable components in the implementation. The FFT is computed on the $p_1\times\ldots\times p_d$ dimensions so that the resulting tensor is of the form $\mathcal{F}u \in \mathbb{C}^{P\times p_1\times\ldots\times p_d\times(d_u+d)}$. Since u is convolved with a function that possess only $\big|Z_{k_{\max}}\big|$ Fourier modes, the higher Fourier modes may be truncated so that one may consider $\mathcal{F}u \in \mathbb{C}^{P\times(2k_{\max,1}+1)\times\ldots\times(2k_{\max,d}+1)\times(d_u+d)}$. The learnable tensor is $R_{\mathsf{W}_{\text{in}}} \in \mathbb{C}^{(2k_{\max,1}+1)\times\ldots\times(2k_{\max,d}+1)\times d_{\text{model}}\times(d_u+d)}$ and the operation in Fourier space is defined as

$$\Big(R\cdot(\mathcal{F}u)\Big)_{p,i_1,\ldots,i_d,\ell} = \sum_{k=1}^{d_u+d} R_{p,i_1,\ldots,i_d,\ell,k}\big(\mathcal{F}u\big)_{p,i_1,\ldots,i_d,k}, \tag{61}$$

for $p\in[\![1,P]\!]$, $1\le i_j\le 2k_{\max,j}+1$, $1\le\ell\le d_{\text{model}}$. The inverse FFT is then applied along the dimensions $2k_{\max,1}+1,\ldots,2k_{\max,d}+1$. The resulting tensor defined as $v^{(0)}\in\mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$ is then fed through the transformer encoder $\mathsf{E}_L : \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}} \to \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$.

The multi-head attention is based on the self-attention operator acting on functions from Definition 14. While the expectation is computed with respect to a probability measure on the patch sequence, hence inducing a discrete probability mass function, the discretization of the domain introduces the need to approximate the L^2 inner product appearing in Definition 13. We resort to an approximation of the integral given by

$$\begin{aligned}\big\langle \mathsf{Q}_h\mathsf{v}(j), \mathsf{K}_h\mathsf{v}(k)\big\rangle_{L^2(D',\mathbb{R}^{d_K})} &= \int_{D'} \big(\mathsf{Q}_h\mathsf{v}(j)\big)(x)\cdot\big(\mathsf{K}\mathsf{v}(k)\big)(x)\,\mathrm{d}x &\text{(62a)}\\ &\approx \sum_{x_i\in D'} \Delta x_i\cdot\big(\mathsf{Q}_h\mathsf{v}(j)\big)(x_i)\cdot\big(\mathsf{K}_h\mathsf{v}(k)\big)(x_i) &\text{(62b)}\end{aligned}$$

where we have denoted by Δx_i the volume of the ith cell. We recall that in this case the linear operators $\mathsf{Q}_h, \mathsf{K}_h, \mathsf{V}_h$ are pointwise local operators hence they admit finite dimensional representations

$Q_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, K_h \in \mathbb{R}^{d_K \times d_{\text{model}}}, V_h \in \mathbb{R}^{d_K \times d_{\text{model}}}$, for $h = 1, \dots, H$. In the case of uniform patching and uniform discretization of the domain of size $s^d = N$, which is the setting we experiment with, the approximation reduces to the approximation of the integral of the form

$$\big\langle \mathsf{Q}_h \mathsf{v}(j), \mathsf{K}_h \mathsf{v}(k) \big\rangle_{L^2(D', \mathbb{R}^{d_K})} \approx \frac{1}{N} \sum_{i_1, \dots, i_d = 1}^{s} \sum_{\ell=1}^{d_K} \big(vQ_h^\top\big)_{j \times i_1 \times \dots \times i_d \times \ell} \big(vK_h^\top\big)_{k \times i_1 \times \dots \times i_d \times \ell}. \tag{63}$$

The multi-head attention operator is thus defined by the application of the linear transformation $W_{\text{MultiHead}}^\top \in \mathbb{R}^{Hd_K \times d_{\text{model}}}$ to the concatenation of $H \in \mathbb{N}$ self-attention operations defined by the approximations above. The concatenation of the outputs of the $H \in \mathbb{N}$ self-attention operations results in a $\mathbb{R}^{P \times p_1 \times \dots \times p_d \times Hd_K}$ tensor, so that

$$\mathsf{A}_{\text{MultiHead}}(v) := \big(\mathsf{A}_{\mathsf{P}}(v; Q_1, K_1, V_1), \dots, \mathsf{A}_{\mathsf{P}}(v; Q_H, K_H, V_H)\big) W_{\text{MultiHead}}^\top \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}, \tag{64}$$

for any $v \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}$. The linear transformations $W_1^\top, W_2^\top \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are applied pointwise. The layer normalization operator is applied as defined in (24) to every point in every patch so that $\mathsf{F}_{\text{LayerNorm}}(v) \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}$.

The operator F_{NN} is applied so that $\mathsf{F}_{\text{NN}}(v) \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}$, for any $v \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}$. Indeed, in this finite-dimensional setting F_{NN} is defined as

$$\mathsf{F}_{\text{NN}}(v; W_3, W_4, b_1, b_2) = f\big(vW_4^\top + b_1\big) W_3^\top + b_2, \tag{65}$$

for any $v \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}$, where f is a nonlinear activation function $W_3^\top, W_4^\top \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are learnable local linear operators and $b_1, b_2 \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}$ are learnable in the last dimension.

After computing the recurrence for E_L as described in Equation (10), a reshaping operator $\mathsf{S}'_{\text{reshape}}$ is applied to the resulting tensor so that

$$v \mapsto \mathsf{S}'_{\text{reshape}}(v) \in \mathbb{R}^{n_1 \times \dots \times n_d \times d_{\text{model}}}, \tag{66}$$

for any $v \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}$. A pointwise linear transformation $W_{\text{out}}^\top \in \mathbb{R}^{d_{\text{model}} \times d_z}$ is then applied according to

$$v \mapsto vW_{\text{out}}^\top, \tag{67}$$

for any $v \in \mathbb{R}^{n_1 \times \dots \times n_d \times d_{\text{model}}}$.

B.3 Fourier Attention Neural Operator

In this section, we describe how the transformer neural operator from Subsection 4.4 is implemented in the discrete setting. In practice, we consider a discretization of the domain D containing N points and evaluations of the function $u \in \mathcal{U}\big(D; \mathbb{R}^{d_u}\big)$ at these points; the resolution of the input is defined by $n_1 \times \dots \times n_d = N$. The input to the model is hence a tensor $u \in \mathbb{R}^{n_1 \times \dots \times n_d \times d_u}$. The input is concatenated with the discretization points $\{x_i\}_{i=1}^N \subset D$, to form the tensor $u_{\text{in}} \in \mathbb{R}^{n_1 \times \dots \times n_d \times (d_u + d)}$. Denoting by $p_1 \times \dots \times p_d$ the resolution of each patch, application of the reshaping operator $\mathsf{S}_{\text{reshape}}$ results in

$$\mathsf{S}_{\text{reshape}}\big(u_{\text{in}}\big) \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times (d_u + d)}.$$

The learnable linear transformation $W_{\text{in}}^\top \in \mathbb{R}^{(d_u + d) \times d_{\text{model}}}$ is then applied as a map

$$\mathsf{S}_{\text{reshape}}\big(u_{\text{in}}\big) \mapsto \mathsf{S}_{\text{reshape}}\big(u_{\text{in}}\big) W_{\text{in}}^\top \in \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}.$$

The resulting tensor $\widetilde{u}^{(0)}$ is then fed through the transformer encoder

$$\mathsf{E}_L : \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}} \to \mathbb{R}^{P \times p_1 \times \dots \times p_d \times d_{\text{model}}}.$$

In the recurrence relation that defines the encoder, $\mathsf{A}_{\text{MultiHead}}$ is the multi-head attention operator as defined in equation (11). In this setting, letting $H \in \mathbb{N}$ denote the number of attention heads, the multi-head attention operator is parametrized by learnable linear integral operators $\mathsf{Q}_h : \mathbb{R}^{p_1\times\ldots\times p_d\times d_{\text{model}}} \to \mathbb{R}^{p_1\times\ldots\times p_d\times d_K}$, $\mathsf{K}_h : \mathbb{R}^{p_1\times\ldots\times p_d\times d_{\text{model}}} \to \mathbb{R}^{p_1\times\ldots\times p_d\times d_K}$, and $\mathsf{V}_h : \mathbb{R}^{p_1\times\ldots\times p_d\times d_{\text{model}}} \to \mathbb{R}^{p_1\times\ldots\times p_d\times d_K}$ for $h = 1, \ldots, H$. Following Definition 20 and the subsequent discussion we implement the linear integral operators as

$$\mathsf{Q}_h(\theta)v = \mathcal{F}^{-1}\Big(R_{\mathsf{Q}_h}(\theta)\cdot(\mathcal{F}v)\Big), \tag{68a}$$

$$\mathsf{K}_h(\theta)v = \mathcal{F}^{-1}\Big(R_{\mathsf{K}_h}(\theta)\cdot(\mathcal{F}v)\Big), \tag{68b}$$

$$\mathsf{V}_h(\theta)v = \mathcal{F}^{-1}\Big(R_{\mathsf{V}_h}(\theta)\cdot(\mathcal{F}v)\Big), \tag{68c}$$

for $h = 1, \ldots, H$ and any $v \in \mathbb{R}^{p_1\times\ldots\times p_d\times d_{\text{model}}}$, where we have made explicit the dependence on $\theta \in \Theta$ to highlight the learnable components in the implementation. In (68) the FFT is computed on the $p_1 \times \ldots \times p_d$ so that the resulting tensor $\mathcal{F}v \in \mathbb{C}^{p_1\times\ldots\times p_d\times d_{\text{model}}}$. Since v is a function that possess only $\big|Z_{k_{\max}}\big|$ Fourier modes, the higher Fourier modes may be truncated so that one may consider $\mathcal{F}v \in \mathbb{C}^{(2k_{\max,1}+1)\times\ldots\times(2k_{\max,d}+1)\times d_{\text{model}}}$. The learnable tensors are of dimension $R_{\mathsf{Q}_h}, R_{\mathsf{K}_h}, R_{\mathsf{V}_h} \in \mathbb{C}^{(2k_{\max,1}+1)\times\ldots\times(2k_{\max,d}+1)\times d_K\times d_{\text{model}}}$ for $h = 1, \ldots, H$, and the operation in Fourier space is defined as

$$\Big(R\cdot(\mathcal{F}v)\Big)_{i_1,\ldots,i_d,\ell} = \sum_{k=1}^{d_{\text{model}}} R_{i_1,\ldots,i_d,\ell,k}\big(\mathcal{F}v\big)_{i_1,\ldots,i_d,k}, \tag{69}$$

for any $v \in \mathbb{R}^{p_1\times\ldots\times p_d\times d_{\text{model}}}$ and $1 \le i_j \le 2k_{\max,j}+1$, $1 \le \ell \le d_K$. The inverse FFT is computed along the dimensions $2k_{\max,1}+1, \ldots, 2k_{\max,d}+1$ so that $\mathsf{Q}_h v, \mathsf{K}_h v, \mathsf{V}_h v \in \mathbb{R}^{p_1\times\ldots\times p_d\times d_K}$ for $h = 1, \ldots, H$. We recall that in this case the self-attention operator A is defined according to Definition 14. While the expectation is computed with respect to a probability measure on the patch sequence, hence inducing a discrete probability mass function, the discretization of the domain introduces the need to approximate the L^2 inner product appearing in Definition 13. We resort to an approximation of the integral given by (62) and (63). The multi-head attention operator is thus defined by the application of the linear transformation $W_{\text{MultiHead}}^\top \in \mathbb{R}^{Hd_K\times d_{\text{model}}}$ to the concatenation of $H \in \mathbb{N}$ self-attention operations. The concatenation of the outputs of the $H \in \mathbb{N}$ self-attention operations results in a $\mathbb{R}^{P\times p_1\times\ldots\times p_d\times Hd_K}$ tensor, so that

$$\mathsf{A}_{\text{MultiHead}}(v) := \big(\mathsf{A}_{\mathsf{P}}(v;\mathsf{Q}_1,\mathsf{K}_1,\mathsf{V}_1),\ldots,\mathsf{A}_{\mathsf{P}}(v;\mathsf{Q}_H,\mathsf{K}_H,\mathsf{V}_H)\big)W_{\text{MultiHead}}^\top \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}, \tag{70}$$

for any $v \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$. The linear transformations $W_1^\top, W_2^\top \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$ are applied pointwise. The layer normalization operator is applied as defined in (16) to every point in every patch so that $\mathsf{F}_{\text{LayerNorm}}(v) \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$. Finally, the operator F_{NN} is parametrized and applied as in (65) so that $\mathsf{F}_{\text{NN}}(v) \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$, for any $v \in \mathbb{R}^{P\times p_1\times\ldots\times p_d\times d_{\text{model}}}$.

A reshaping operator $\mathsf{S}'_{\text{reshape}}$ to the tensor resulting from the transformer encoder iteration as in Equation (66). A pointwise linear transformation $W_{\text{out}}^\top \in \mathbb{R}^{d_{\text{model}}\times d_z}$ is then applied as in Equation (67).

Appendix C. Complexity of the Transformer Neural Operators

We summarize the expressions for the parameter complexity and evaluation cost (measured in floating point operations, FLOPS) of the neural operators from Section 4 and the Fourier neural operator (FNO) from Li et al. (2021). These are provided in Tables 4 and 5, respectively. In Subsection C.1 and C.2 we provide further details for the computations of the parameter and evaluation cost complexities, respectively.

C.1 Parameter Complexity

Architecture	**Parameter Complexity**
FNO	$(d_u + d) \cdot d_{\text{FNO}} + 2d_{\text{FNO}} \cdot d_{\text{model}} + d_{\text{FNO}} \cdot d_z + 4 \cdot \left(d_{\text{model}}^2 \cdot k_{\max} + d_{\text{model}}^2\right)$
AFNO	$(d_u + d) \cdot d_{\text{model}} + N d_{\text{model}} + d_{\text{model}} d_z + 4 \cdot (8 + 4/b) \cdot d_{\text{model}}^2$
Transformer NO	$(d_u + d) \cdot d_{\text{model}} + d_{\text{model}} \cdot d_z + 6 \cdot \left(6d_{\text{model}}^2 + 2d_{\text{model}}\right)$
ViT NO	$(d_u + d) \cdot d_{\text{model}} \cdot k_{\max} + d_{\text{model}} \cdot d_z + 6 \cdot \left(6d_{\text{model}}^2 + 2d_{\text{model}}\right)$
FANO	$(d_u + d) \cdot d_{\text{model}} + d_{\text{model}} \cdot d_z + 6 \cdot \left(3d_{\text{model}}^2 \cdot k_{\max} + 3d_{\text{model}}^2 + 2d_{\text{model}}\right)$

Table 4: Parameter complexity for the three proposed transformer neural operators, as implemented in practice, compared to the Fourier neural operator (Li et al., 2021). We note that for our implementations, d_{FNO}, which is the latent dimension of the two-layer lifting and projection MLP in FNO, is fixed to be 128, while the hyperparameter b in the AFNO is fixed to be 8. Parameters arising from possible bias terms, parameters arising from skip connections in FNO and parameters arising from normalizations in AFNO are omitted for ease of presentation.

C.2 Evaluation Cost

We note that a matrix vector product Wu for $W \in \mathbb{R}^{r \times r'}$ has cost $2rr'$. We approximate the cost of computing the FFT of a vector in $\mathbb{R}^r$ as $5r \log r$ (Hoop et al., 2022). We approximate the cost of the two-dimensional real FFT, used in implementation, by $(15/2) \cdot N \log(\sqrt{N})$ where N is the total number of grid points. Assuming that the grid is of size $s \times s = N$ for s an even integer, 1d FFTs are first computed for each row. The real FFT requires $s/2$ column 1d FFTs, hence the $(15/2) \cdot N \log \sqrt{N}$ complexity.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint*, arXiv:2106.01506, 2014.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619:1–6, 07 2023.

Shuhao Cao. Choose a transformer: Fourier or Galerkin. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24924–24940. Curran Associates, Inc., 2021.

Gary J. Chandler and Rich R. Kerswell. Invariant recurrent solutions embedded in a turbulent two-dimensional Kolmogorov flow. *Journal of Fluid Mechanics*, 722:554–595, 2013. doi: 10.1017/jfm.2013.122.

A. Chattopadhyay, M. Mustafa, P. Hassanzadeh, E. Bach, and K. Kashinath. Towards physics-inspired data-driven weather forecasting: integrating data assimilation with a deep spatial-transformer-based u-net in a case study with era5. *Geoscientific Model Development*, 15(5):2221–2237, 2022.

Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5659–5667, 2017.

Sibo Cheng, César Quilodrán-Casas, Said Ouala, Alban Farchi, Che Liu, Pierre Tandeo, Ronan Fablet, Didier Lucor, Bertrand Iooss, Julien Brajard, Dunhui Xiao, Tijana Janjic, Weiping Ding, Yike Guo, Alberto Carrassi, Marc Bocquet, and Rossella Arcucci. Machine learning with data

Architecture	Evaluation Cost
FNO	$2N(d_u + d + 2d_{\text{model}})d_{\text{FNO}} + 2Nd_{\text{FNO}}d_z + 4\big(15d_{\text{model}}N\log(\sqrt{N}) + k_{\max}\cdot(2d_{\text{model}}^2 - d_{\text{model}}) + 2Nd_{\text{model}}^2\big)$
Transformer NO	$2N(d_u + d)d_{\text{model}} + 2Nd_{\text{model}}d_z + 6\big(8d_{\text{model}}^2N + 2N^2d_{\text{model}} + 4d_{\text{model}}^2N\big)$
ViT NO	$(d_u + d + d_{\text{model}})\cdot 15N\log(\sqrt{N/P})/2 + Pk_{\max}d_{\text{model}}\big(2(d_u + d) - 1\big) + 2Nd_{\text{model}}d_z + 6\Big(8d_{\text{model}}^2N + 2d_{\text{model}}NP + 4d_{\text{model}}^2N\Big)$
FANO	$2N(d_u + d)d_{\text{model}} + 2Nd_{\text{model}}d_z + 6\Big(45d_{\text{model}}N\log(\sqrt{N/P}) + 3k_{\max}P\big(2d_{\text{model}}^2 - d_{\text{model}}\big) + 2Nd_{\text{model}}^2 + 2d_{\text{model}}NP + 4Nd_{\text{model}}^2\Big)$
AFNO	$2N(d_u + d)d_{\text{model}} + 2Nd_{\text{model}}d_z + 4\big(16d_{\text{model}}^2 + 15Nd_{\text{model}}\log(\sqrt{N}) + 6N(2d_{\text{model}}^2/b - d_{\text{model}})\big)$

Table 5: Evaluation cost (in FLOPS) for the proposed transformer neural operators, as implemented in practice, compared to FNO (Li et al., 2021). As done in Kaplan et al. (2020) and for ease of presentation, the additional evaluation cost arising from nonlinear activations, the softmax operator and normalizations are omitted. Furthermore, we omit the cost arising from skip connections. Indeed these do not affect the scaling. We further note that for our implementations, d_{FNO}, which is the latent dimension of the two-layer lifting and projection MLP in FNO, is fixed to be 128. In this table we present the expression for the evaluation cost of AFNO without consideration of possible patching, which we do not employ for our comparisons: this in combination with the fact that we do not employ mode truncation is the reason for the Nd_{model}^2 term. In our context, the parameter b in AFNO is set to $b = 8$.

assimilation and uncertainty quantification for dynamical systems: A review. *IEEE/CAA Journal of Automatica Sinica*, 10(6):1361–1387, 2023.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Neural Information Processing Systems 2014 Workshop on Deep Learning, December 2014*, 2014.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Efficient token mixing for transformers via adaptive fourier neural operators. In *International Conference on Learning Representations*, 2022.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Kevin Hayden, Eric Olson, and Edriss S. Titi. Discrete data assimilation in the Lorenz and 2d Navier–Stokes equations. *Physica D: Nonlinear Phenomena*, 240(18):1416–1425, 2011.

Juncai He, Xinliang Liu, and Jinchao Xu. MgNO: Efficient parameterization of linear operators via multigrid. In *The Twelfth International Conference on Learning Representations*, 2024.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 11 1997.

Maarten V. de Hoop, Daniel Zhengyu Huang, Elizabeth Qian, and Andrew M. Stuart. The cost-accuracy trade-off in operator learning with neural networks. *Journal of Machine Learning*, 1(3): 299–341, 2022. ISSN 2790-2048.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint*, arXiv:2001.08361, 2020.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Samuel Lanthaler, Zongyi Li, and Andrew M. Stuart. Nonlocality and nonlinearity implies universality in operator learning. *Constructive Approximation*, Aug 2025. ISSN 1432-0940. doi: 10.1007/s00365-025-09718-3. URL `https://doi.org/10.1007/s00365-025-09718-3`.

Zhijie Li, Tianyuan Liu, Wenhui Peng, Yuan Zelong, and Jianchun Wang. A transformer-based neural operator for large-eddy simulation of turbulence. *Physics of Fluids*, 36, 06 2024. doi: 10.1063/5.0210493.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, Los Alamitos, CA, USA, 10 2021. IEEE Computer Society.

Edward N Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130–141, 1963.

Xihaier Luo, Xiaoning Qian, and Byung-Jun Yoon. Hierarchical neural operator transformer with learnable frequency-aware loss prior for arbitrary-scale super-resolution. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024.

André Martins, António Farinhas, Marcos Treviso, Vlad Niculae, Pedro Aguiar, and Mario Figueiredo. Sparse and continuous attention mechanisms. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20989–21001. Curran Associates, Inc., 2020.

Alexander Moreno, Zhenke Wu, Supriya Nagesh, Walter Dempsey, and James M Rehg. Kernel multimodal continuous attention. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 18046–18059. Curran Associates, Inc., 2022.

Oded Ovadia, Adar Kahana, Panos Stinis, Eli Turkel, Dan Givoli, and George Em Karniadakis. Vito: Vision transformer-operator. *Computer Methods in Applied Mechanics and Engineering*, 428: 117109, 2024. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2024.117109.

Louis M. Pecora and Thomas L. Carroll. Synchronization in chaotic systems. *Phys. Rev. Lett.*, 64: 821–824, 2 1990.

PhysicsNeMo Contributors. Nvidia physicsnemo: An open-source framework for physics-based deep learning in science and engineering. `https://github.com/NVIDIA/physicsnemo`, February 2023. Version released on 2023-02-24.

Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8: 143–195, 1999.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, Jan 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08252-9. URL `https://doi.org/10.1038/s41586-024-08252-9`.

Md Ashiqur Rahman, Robert Joseph George, Mogab Elleithy, Daniel Leibovici, Zongyi Li, Boris Bonev, Colin White, Julius Berner, Raymond A. Yeh, Jean Kossaifi, Kamyar Azizzadenesheli, and Anima Anandkumar. Pretraining codomain attention neural operators for solving multiphysics PDEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Daniel Sanz-Alonso, Andrew Stuart, and Armeen Taeb. *Inverse Problems and Data Assimilation*. London Mathematical Society Student Texts. Cambridge University Press, 2023.

R. Temam. *Infinite-Dimensional Dynamical Systems in Mechanics and Physics*. Applied Mathematical Sciences. Springer New York, 2012. ISBN 9781468403138.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115783. URL `https://www.sciencedirect.com/science/article/pii/S0045782522007393`.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4344–4353, Hong Kong, China, November 2019. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Sifan Wang, Jacob H Seidman, Shyam Sankaran, Hanwen Wang, George J. Pappas, and Paris Perdikaris. CViT: Continuous vision transformer for operator learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. In *International Joint Conference on Artificial Intelligence(IJCAI)*, 2023.

Matthew A Wright and Joseph E. Gonzalez. Transformers are deep infinite-dimensional non-mercer binary kernel machines. *arXiv preprint*, arXiv:2106.01506, 2021.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020.