NON-EQUISPACED FOURIER NEURAL SOLVERS FOR PDES

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently proposed neural resolution-invariant models, despite their effectiveness and efficiency, usually require equispaced spatial points of data for solving partial differential equations. However, sampling in spatial domain is sometimes inevitably non-equispaced in real-world systems, limiting their applicability. In this paper, we propose a Non-equispaced Fourier PDE Solver (NFS) with adaptive interpolation on resampled equispaced points and a variant of Fourier Neural Operators as its components. Experimental results on complex PDEs demonstrate its advantages in accuracy and efficiency. Compared with the spatially-equispaced benchmark methods, it achieves superior performance with 42.85% improvements on MAE, and is able to handle non-equispaced data with a tiny loss of accuracy. Besides, NFS as a model with mesh invariant inference ability, can successfully model turbulent flows in non-equispaced scenarios, with a minor deviation of the error on unseen spatial points.

1 INTRODUCTION

Solving the partial differential equations (PDEs) holds the key to revealing the underlying mechanisms and forecasting the future evolution of the systems. Recently, data-driven neural PDE solvers revolutionize this field by providing fast and accurate solutions for PDEs. Unlike approaches designed to model one specific instance of PDE (E & Yu, 2017; Bar & Sochen, 2019; Smith et al., 2020; Pan & Duraisamy, 2020; Raissi et al., 2020), neural operators (Guo et al., 2016; Sirignano & Spiliopoulos, 2018; Bhatnagar et al., 2019; KHOO et al., 2020; Li et al., 2020b;d; Bhattacharya et al., 2021; Brandstetter et al., 2022; Lin et al., 2022) directly learn the mapping between infinite-dimensional spaces of functions. They remedy the mesh-dependent nature of the finite-dimensional operators by producing a single set of network parameters that may be used with different discretizations.

However, a problem still exist – discretization-invariant modeling for non-equispaced data. On one hand, classical vision models and graph spatio-temporal models are not discretization-invariant, while the infinite neural operator like FNO (Li et al., 2020c) is. On the other hand, despite computational efficiency, vision models including FNO are equispace-necessary, and limited to handling images as 2-d regular grids. Therefore, two properties should be available in neural PDE solvers: *(1)* discretization-invariance and *(2)* equispace-unnecessity, and recently proposed methods can be classified into four types according to the two properties, as shown in Fig. 1.

As discussed, although the equispace-necessary methods enjoy fast parallel computation and low prediction error, they lack the ability to handle the spatially non-equispaced data. For these reasons, we aim to design a mesh-invariant model (defined in Fig. 1) called Non-equispaced Fourier neural Solver (NFS) with comparably low cost of computation and high accuracy, by lending the powerful expressivity of FNO and vision models to efficiently solve the complex PDE systems.

2 BACKGROUND AND RELATED WORK

2.1 PROBLEM STATEMENT

Let $D \in \mathbb{R}^d$ be the bounded and open spatial domain where n_s-point discretization of the domain D written as $\boldsymbol{X} = \{\boldsymbol{x}_i = (x_i^{(1)}, \ldots, x_i^{(d)}) : 1 \leq i \leq n_s\}$ are sampled. The observation of input function $a \in \mathcal{A}(D; \mathbb{R}^{d_a})$ and output $u \in \mathcal{U}(D; \mathbb{R}^{d_u})$ on the n_s points are denoted by $\{a(\boldsymbol{x}_i), u(\boldsymbol{x}_i)\}_{i=1}^{n_s}$, where $\mathcal{A}(D; \mathbb{R}^{d_a})$ and $\mathcal{U}(D; \mathbb{R}^{d_u})$ are separable Banach spaces of function taking values in $\mathbb{R}^{d_a}$ and $\mathbb{R}^{d_u}$

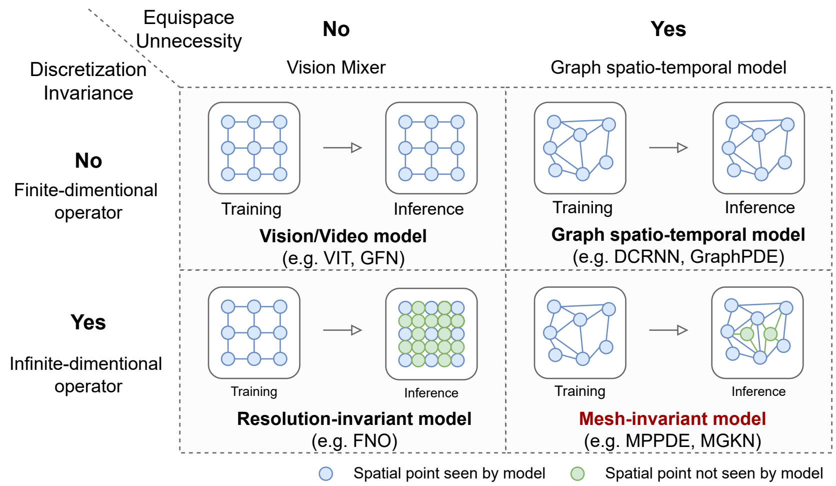

Figure 1: Four types of methods with or without the two concluded limitations.

respectively. Suppose $\boldsymbol{x} \sim \mu$ is i.i.d. sampled from the probability measure μ supported on D. An infinite-dimensional neural operator $\mathcal{G}_\theta : \mathcal{A}(D; \mathbb{R}^{d_a}) \to \mathcal{U}(D; \mathbb{R}^{d_u})$ parameterized by $\theta \in \Theta$, aims to build an approximation so that $\mathcal{G}_\theta(a) \approx u$. To establish a mesh-invariant operator, $\boldsymbol{X}$ can be non-equispaced, and the learned $\mathcal{G}_\theta$ should be transferred to an arbitary discretization $\boldsymbol{X}' \in D$, where $\boldsymbol{x} \in \boldsymbol{X}'$ can be not necessarily contained in $\boldsymbol{X}$. Because we focus on spatially non-equispaced points, when the PDE system is time-dependent, we assume that timestamps $\{t_j\}$ are uniformly sampled.

2.2 Method Preliminaries

Kernel integral operator method (Li et al., 2020a) is a family of infinite-dimensional operators, in which $(\mathcal{G}_\theta(a))(\boldsymbol{x}) = Q \circ v^{\mathrm{T}} \circ \cdots \circ v^1 \circ P(a)(\boldsymbol{x})$ is formulated as an iterative architecture. A higher-dimensional representation function is first obtained by $v^0 = P(a) \in \mathcal{U}(D; \mathbb{R}^{d_v})$, where P is a shallow fully-connected network. It is updated by

$$v^{\mathrm{t}+1}(\boldsymbol{x}) := \sigma(W v^{\mathrm{t}}(\boldsymbol{x}) + \mathcal{K}_\phi(a) v^{\mathrm{t}}(\boldsymbol{x})), \qquad \forall \boldsymbol{x} \in D \tag{1}$$

where $\mathcal{K}_\phi : \mathcal{A} \to \mathcal{L}(\mathcal{U})$ is a kernel integral operator mapping, mapping a to bounded linear operators, with parameters ϕ. W is a linear transform and σ is a non-linear activation function. After the final iteration, Q projects $v^{\mathrm{T}}(\boldsymbol{x})$ back to $\mathcal{U}(D; \mathbb{R}^{d_u})$.

Fourier Neural Operator (FNO) (Li et al., 2020d) as a member in kernel integral operator methodsLi et al. (2020a), updates the representation by applying the convolution theorem as:

$$\mathcal{K}_\phi(a) v(\boldsymbol{x}) = \mathcal{F}^{-1}(\mathcal{F}(\kappa_\phi) \cdot \mathcal{F}(v))(\boldsymbol{x}) = \mathcal{F}^{-1}(R_\phi \cdot \mathcal{F}(v))(\boldsymbol{x}), \tag{2}$$

The discrete Fourier transform of $f : D \to \mathbb{R}^{d_f}$ is denoted by $\mathcal{F}(f)(\boldsymbol{k}) \in \mathbb{C}^{d_f}$, with $\mathcal{F}^{-1}$ as its inverse. R_ϕ as the Fourier transform of a periodic kernel function κ_ϕ, is directly learned as the parameters in the updating process. Because the sampled spatial points are equispaced in FNO, it can efficiently conduct fast Fourier transform (FFT) and its inverse (IFFT) to get the Fourier series.

Vision Mixers (Dosovitskiy et al., 2020; Tolstikhin et al., 2021; Rao et al., 2021; Guibas et al., 2021) are a line of models with a stack of (token mixing) - (channel mixing) - (token mixing) as their network structure for vision tasks. The defined tokens are equivalent to equispaced spatial points in the former definition In specific, VIT uses a non-Mercer kernel function (Wright & Gonzalez, 2021) κ_ϕ to adaptively learn the pattern of message-passing through the iterative updating process

$$\begin{gathered} v^{\mathrm{t}+1}(\boldsymbol{x}) = \sigma(\mathrm{ChannelMix} \circ \mathrm{TokenMix}(v^{\mathrm{t}}(\boldsymbol{x}))); \\ \mathrm{TokenMix}(v(\boldsymbol{x})) = \sum_i \kappa_\phi(\boldsymbol{x}, \boldsymbol{x}_i, v(\boldsymbol{x}), v(\boldsymbol{x}_i)) \cdot v(\boldsymbol{x}_i); \quad \mathrm{ChannelMix}(v(\boldsymbol{x})) = W v(\boldsymbol{x}), \end{gathered} \tag{3}$$

where W is a linear transform called channel mixing layer because it transforms the input on the channel of an image whose dimension is equivalent to function dimension d_f. We omit the residual connection in Eq. (1) for simplicity. Note that the FNO can be regarded as a member of the family

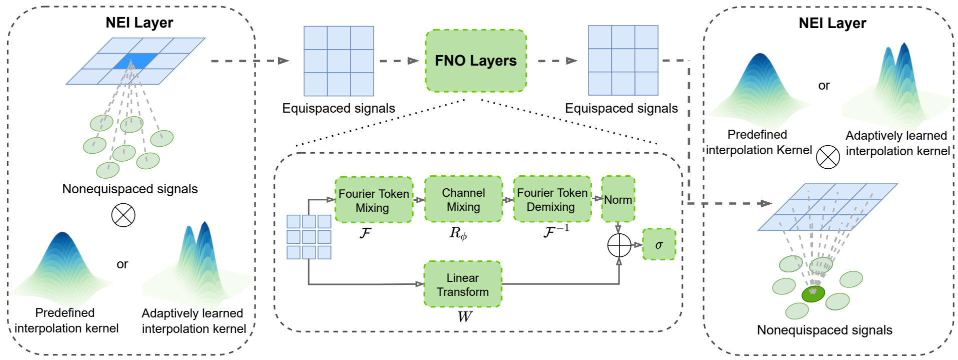

Figure 2: The architecture of NFS: In non-equispaced interpolation (NEI) layers, the kernels are adaptively learned rather than predefined, and the interpolated equispaced signals are processed through a stack of FNO layers with the same structure of Vision Mixers.

of Vision Mixers (See Appendix A). However, the powerful fitting ability and efficiency of Vision Mixers are limited to being applied to non-equispaced spatial points.

Graph spatio-temporal models (Seo et al., 2016; Li et al., 2018; Bai et al., 2020; Lin et al., 2021) as a solution to model non-equispaced spatial points, model interaction patterns among spatial points in a graph message-passing way, whose mechanism is similar to the token mixing in Vision Mixers. However, they suffer from high computational complexity and unsatisfactory accuracy.

3 Proposed Method

3.1 Non-equispaced Fourier Transform

Non-equispaced FFTs usually rely on a mixture of interpolation and the judicious use of FFT, where the calculations of interpolation are no more than $O(n_s \log n_s)$ operations (Kalamkar et al., 2012; Cheema et al., 2017). For example, Gaussian-based interpolation (Kestur et al., 2010) is commonly used. Denote $\mathcal{F}$ as equispaced FFT in particular, and $\mathcal{H}$ as the interpolation operator, and the proposed non-equispaced FFT is written as

$$(\mathcal{F} \circ \mathcal{H}(f))(\boldsymbol{k}) \approx \sqrt{\frac{\pi}{\tau}} e^{\tau<\boldsymbol{k},\boldsymbol{k}>} \sum_{j=1}^{m_s} e^{-2i\pi<\boldsymbol{k},\boldsymbol{x}_j>} \sum_{i=1}^{n_s} f(\boldsymbol{x}_i) h_\tau(\boldsymbol{x}_i - \boldsymbol{x}_j). \tag{4}$$

$\mathcal{H}(f)(\boldsymbol{x}_j) = \sum_{i=1}^{n_s} f(\boldsymbol{x}_i) h_\tau(\boldsymbol{x}_i - \boldsymbol{x}_j)$ interpolates values on resampled points via convolution with the periodic heat kernel $h_\tau(\boldsymbol{x} - \boldsymbol{y}) = \sum_{\boldsymbol{l} \in \mathbb{Z}^d} e^{-(\boldsymbol{x}-\boldsymbol{y})^2/4\tau}$, with τ as a constant.

3.2 Non-equispaced Fourier Neural PDE Solver

Non-equispaced interpolation. To harness the effectiveness of FNO, we use non-equispaced Fourier token mixing instead of the equispaced one. It generalizes the equispaced FFT in Eq. (2) as

$$\tilde{\mathcal{F}}(v) = (\mathcal{F} \circ \mathcal{H}_\eta(a))(v). \tag{5}$$

We denote $\mathcal{H}_\eta : \mathcal{A} \to \mathcal{L}(\mathcal{U})$ as the interpolation operator mapping, which maps parametric function to a bounded interpolation operator. $\mathcal{H}_\eta(a)$ gets the interpolated values on m_s resampled equispaced points via the convolution with kernel h_η as

$$(\mathcal{H}_\eta(a)v)(\boldsymbol{x}_j) = \frac{1}{n_s} \sum_{i=1}^{n_s} v(\boldsymbol{x}_i) h_\eta(\boldsymbol{x}_j - \boldsymbol{x}_i, \boldsymbol{x}_i, a(\boldsymbol{x}_i)), \tag{6}$$

where $\boldsymbol{x}_j$ lies on resampled equispaced grids. Another $\mathcal{H}'_\zeta$ interpolates back on the n_s non-equispaced ones in the same way via the convolution with kernel h_ζ. To reduce the operations to no more than $O(n_s \log n_s)$, the summation is restricted in the neighborhood of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, such that $|\mathcal{N}(\boldsymbol{x}_i)| \approx |\mathcal{N}(\boldsymbol{x}_j)| \le c \log n_s$ with c as a predefined constant determining the neighborhood size of spatial points. We formulate the kernel with a shallow feed-forward neural network. Thanks to the universal approximation of neural networks, the following theorem assures that the interpolation operator can approximate the representation function v arbitrarily well. (For detailed proof, see Appendix. B.) Empirical observations on the convergence of interpolation operators are given in Appendix C.

Theorem 3.1 (Approximation Theorem of the Adaptive Interpolation)**.** *Assume the setting of* ***Theorem B.2*** *in Appendix. B is satisfied.* μ *is the probability measure supported on* D*. For* $v \in \mathcal{U}$*, suppose* $\mathcal{U} = L^p(D; \mathbb{R}^{d_v})$*, for any* $1 < p < \infty$*. Then, given* $\epsilon > 0$*, there exist a neural network* $h_\eta : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^{d_a} \to \mathbb{R}^{d_v}$*, such that* $||\hat{v} - v||_{\mathcal{U}} \leq \epsilon$*, where* $\hat{v}(\boldsymbol{x}) = \int_D h_\eta(\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{x}, a(\boldsymbol{y}))v(\boldsymbol{y})d\mu(\boldsymbol{y})$*.*

3.3 EXPERIMENTS

Benchmarks and protocols. For finite-dimensional operators, we choose Vision Mixers including VIT (Dosovitskiy et al., 2020), GFN (Rao et al., 2021) and MLPMIXER (Tolstikhin et al., 2021) as equispaced problem solvers, with DEEPONET-V and DEEPONET-U as two variants for DeepONet(Lu et al., 2021) and graph spatio-temporal models including DCRNN (Li et al., 2018), AGCRN (Bai et al., 2020) and GCGRU (Seo et al., 2016) as non-equispaced problem solvers. For infinite-dimensional operators, the state-of-the-art FNO (Li et al., 2020d) for equispaced problems and MPPDE (Brandstetter et al., 2022) for non-equispaced problems are chosen. A brief introduction to these models is shown in Appendix. B.1. The widely-used metrics - Mean Absolute Error (MAE) is deployed to measure the performance. The reported mean and standard deviation of metrics are obtained through 5 independent trials.

Data. We choose 4 equations in experiments: Korteweg de Vries (KdV) and Burgers' equation for 1-d problem and Darcy Flow and Navier-Stokes (NS) equation for a viscous, incompressible fluid in vorticity form on the unit torus for 2-d PDEs. The total number of instances is 1200, with percentages of 0.7, 0.1 and 0.2 for training, validating and testing, respectively. When evaluating their performance in equispaced scenarios of different resolutions, we can downsample the resolution for training to low-resolution data, e.g. 64×64 in NS equation. To evaluate their performance in non-equispaced scenarios of different meshes, we randomly choose n_s spatial points for training. Details are given in Appendix. B.2.

Table 1: MAE($\times 10^{-3}$) comparison with vision mixer benchmarks.

	Burgers' ($n_t = 10$)			Darcy Flow			NS ($n_t = 1$)			NS ($n_t = 10$)		
r	512	512	1024	64	128	256	64	64	128	64	64	128
n'_t	10	40	20	1	1	1	10	40	20	10	40	20
VIT	0.5042	2.4269	1.5327	0.5073	0.9865	1.1078	9.3797	22.8565	15.7398	3.9609	12.3433	9.3010
MLPMIXER	0.1973	0.4210	0.3303	0.4970	0.8909	0.9125	7.5246	15.8632	14.9360	3.1530	7.9291	7.7410
GFN	0.2383	0.4187	0.3500	0.4739	0.8659	0.9618	3.5524	10.2250	6.3976	1.7396	5.4464	3.1261
FNO	0.0978	0.1815	**0.1430**	0.4289	0.7086	0.9075	3.3425	8.9857	4.4627	2.4076	7.6979	3.7001
DEEPONET-U	0.4471	1.9624	0.6541	0.3753	0.9488	0.9692	7.4912	16.0440	14.3476	3.4436	10.2950	7.1394
DEEPONET-V	0.4782	2.1707	1.6131	0.5119	0.9614	1.3216	8.6986	18.5561	16.0587	3.9745	12.3314	9.3471
NFS	**0.0958**	**0.1708**	0.1474	**0.1497**	**0.2254**	**0.4216**	**1.7425**	**4.7882**	**2.6988**	**0.8636**	**3.1122**	**1.8406**

Performance comparison. In this part, for time-dependent PDEs, our target is to map the observed physical quantities from initial condition $u(\boldsymbol{X}, \boldsymbol{T}) \in \mathbb{R}^{n_s \times n_t}$, where $\boldsymbol{T} = \{t_i : t_i < T\}_{1 \leq i \leq n_t}$, to quantities at some later time $u(\boldsymbol{X}, \boldsymbol{T'}) \in \mathbb{R}^{n_s \times n'_t}$, where $\boldsymbol{T'} = \{t_i : T < t_i < T'\}_{1 \leq i \leq n'_t}$. We set the input timestamp number n_t as 1 (initial state to future dynamics) and 10 (sequence to sequence), and prediction horizon n'_t as 10, 20 and 40 as short-, mid- and long-term settings. For Darcy Flows, which are independent of time, we directly build an operator to map a to u. In equispaced scenarios, the resolution is denoted by $r^d = n_s$, where d is the spatial dimension. In non-equispaced scenarios, the spatial points number is denoted by n_s. The comparison of methods are shown in Table. 1, Table. B3 and Fig. B2, and detailed results are given in Appendix. B.3. It can be concluded that *(1)* All of the evaluated Vision Mixers are able to model the dynamical systems effectively, in spite of FNO as the only discretization-invariant model; *(2)* In equispaced scenarios, the proposed NFS obtains the lowest error in most 1-d PDE settings, and in solving 2-d PDEs, its superiority over other Vision Mixers are significant, with 42.85% improvements on MAE according to the trials of NS ($r = 64, n_t = 10, n'_t = 40$). *(3)* In non-equispaced scenarios, the evaluated graph spatio-temporal models' performance is unsatisfactory, especially in NS equations. In comparison, NFS achieves comparable high accuracy to the equispaced scenarios, for instance, according to columns of NS ($r = 64, n_t = 10, n'_t = 40$) with ($n_s = 4096, n_t = 10, n'_t = 40$).

Mesh-invariance evaluation and Architecture Analysis. For the evaluation of mesh-invariance and further analysis on model architecture with more visualization of results, see Appendix. B in detail. We can conclude that *(1)* The errors on unseen meshes are larger than the errors on seen meshes, but acceptable, since they are even lower than other models' prediction error on seen meshes. *(2)* Larger n'_s leads to higher prediction error because a large number of unseen points are likely to disturb the learned token mixing patterns.

REFERENCES

James Atwood and Don Towsley. Diffusion-convolutional neural networks, 2016.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting, 2020.

Leah Bar and Nir Sochen. Unsupervised deep learning algorithm for pde-based forward and inverse problems, 2019.

Saakaar Bhatnagar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, and Shailendra Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64(2):525–545, jun 2019. doi: 10.1007/s00466-019-01740-0. URL https://doi.org/10.1007%2Fs00466-019-01740-0.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model reduction and neural networks for parametric pdes, 2021.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers, 2022. URL https://arxiv.org/abs/2202.03376.

Umer I. Cheema, Gregory Nash, Rashid Ansari, and Ashfaq Khokhar. Memory-optimized re-gridding architecture for non-uniform fast fourier transform. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(7):1853–1864, 2017. doi: 10.1109/TCSI.2017.2681723.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995. doi: 10.1109/72.392253.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers, 2021. URL https://arxiv.org/abs/2104.13840.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering, 2017.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth, 2021. URL https://arxiv.org/abs/2103.03404.

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, 2020. URL https://arxiv.org/abs/2010.11929.

Weinan E and Bing Yu. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems, 2017.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers, 2021. URL https://arxiv.org/abs/2111.13587.

Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 481–490, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939738. URL https://doi.org/10.1145/2939672.2939738.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208.

Dhiraj D. Kalamkar, Joshua D. Trzaskoz, Srinivas Sridharan, Mikhail Smelyanskiy, Daehyun Kim, Armando Manduca, Yunhong Shu, Matt A. Bernstein, Bharat Kaul, and Pradeep Dubey. High performance non-uniform fft on modern x86-based multi-core systems. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium*, pp. 449–460, 2012. doi: 10.1109/IPDPS.2012.49.

Srinidhi Kestur, Sungho Park, Kevin M. Irick, and Vijaykrishnan Narayanan. Accelerating the nonuniform fast fourier transform using fpgas. In *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, pp. 19–26, 2010. doi: 10.1109/FCCM.2010.13.

YUEHAW KHOO, JIANFENG LU, and LEXING YING. Solving parametric PDE problems with artificial neural networks. *European Journal of Applied Mathematics*, 32(3):421–435, jul 2020. doi: 10.1017/s0956792520000182. URL https://doi.org/10.1017%2Fs0956792520000182.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces, 2021. URL http://tensorlab.cms.caltech.edu/users/anima/pubs/GraphPDE_Journal.pdf.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2018.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations, 2020a. URL https://arxiv.org/abs/2003.03485.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations, 2020b.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations, 2020c.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2020d. URL https://arxiv.org/abs/2010.08895.

Haitao Lin, Zhangyang Gao, Yongjie Xu, Lirong Wu, Ling Li, and Stan. Z. Li. Conditional local convolution for spatio-temporal meteorological forecasting, 2021.

Haitao Lin, Guojiang Zhao, Lirong Wu, and Stan Z. Li. Stonet: A neural-operator-driven spatio-temporal network, 2022. URL https://arxiv.org/abs/2204.08414.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. URL https://arxiv.org/abs/2103.14030.

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

Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer, 2021. URL https://arxiv.org/abs/2105.07926.

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers, 2021. URL https://arxiv.org/abs/2105.10497.

Shaowu Pan and Karthik Duraisamy. Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. *SIAM Journal on Applied Dynamical Systems*, 19 (1):480–509, Jan 2020. ISSN 1536-0040. doi: 10.1137/19m1267246. URL http://dx.doi.org/10.1137/19M1267246.

Namuk Park and Songkuk Kim. How do vision transformers work?, 2022. URL https://arxiv.org/abs/2202.06709.

Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification, 2021. URL https://arxiv.org/abs/2107.00645.

Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks, 2016.

Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, dec 2018. doi: 10.1016/j.jcp.2018.08.029. URL https://doi.org/10.1016%2Fj.jcp.2018.08.029.

Jonathan D. Smith, Kamyar Azizzadenesheli, and Zachary E. Ross. Eikonet: Solving the eikonal equation with deep neural networks, 2020.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021. URL https://arxiv.org/abs/2105.01601.

Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L. Griffiths. Are convolutional neural networks or transformers more like human vision?, 2021. URL https://arxiv.org/abs/2105.07197.

Matthew A. Wright and Joseph E. Gonzalez. Transformers are deep infinite-dimensional non-mercer binary kernel machines, 2021. URL https://arxiv.org/abs/2106.01506.

Baosong Yang, Longyue Wang, Derek Wong, Lidia S. Chao, and Zhaopeng Tu. Convolutional self-attention networks, 2019. URL https://arxiv.org/abs/1904.03107.