Published as a conference paper at ICLR 2021

FOURIER $\boxed{\text{NEURAL OPERATOR}}$ FOR PARAMETRIC PARTIAL DIFFERENTIAL EQUATIONS

Zongyi Li
zongyili@caltech.edu

Nikola Kovachki
nkovachki@caltech.edu

Kamyar Azizzadenesheli
kamyar@purdue.edu

Burigede Liu
bgl@caltech.edu

Kaushik Bhattacharya
bhatta@caltech.edu

Andrew Stuart
astuart@caltech.edu

Anima Anandkumar
anima@caltech.edu

Main Message

- Necessity of having mappings between function spaces
- Introduction to a line of research called “Neural Operators”
- Introduction to Fourier Neural Operators

Mapping between function spaces is important!

- Most real-world phenomena are governed by Partial Differential Equations (PDEs), and to model or to predict them we need to solve these PDEs.

- To solve PDEs we need to provide them boundary and initial conditions. The initial condition and boundary conditions are functions, and the prediction is also a function.

Examples of real-world PDEs

Wave Equation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

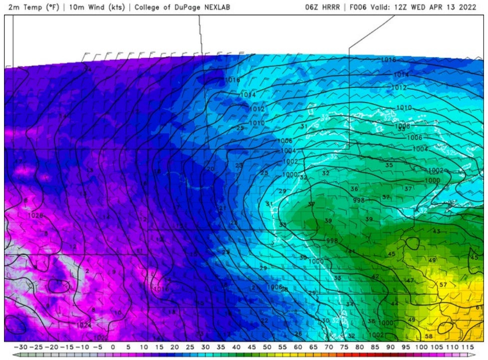
Schrödinger equation in quantum-mechanical system:

$$i\hbar \frac{\partial}{\partial t}\Psi(x,t) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x,t)\right]\Psi(x,t)$$

Navier–Stokes equations for describing the motion of fluids:

$$\rho \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho\, \mathbf{g}$$

For example, in the case of using Navier Stokes for **Weather Forecasting**, the current measurements of the state of atmosphere are our initial conditions and the characteristics of the land and an oceans is our boundary condition.



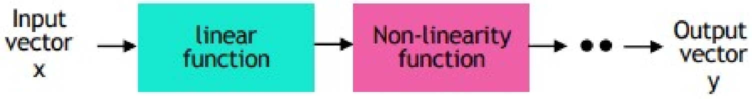
Mapping between function spaces is important!

- Currently, we use numerical methods to solve PDEs, and the solution is unique for each boundary and initial condition.
- Often it takes a lot of effort to solve PDEs and even after simulation, it cannot be used for other predictions or simulations.
- This mapping is done by our PDEs how to do it with Neural?
- And to do that, a proper NN architecture is required plus a training data set of input functions and output functions!
- So, this is what motivated this line of research to develop instant mapping from these functions to the solution function.

Neural Networks vs Neural Operators

Neural networks

$$f:\ x \to y$$
$$R^n \to R^m$$



$$\begin{aligned} v_0 &= x \\ v_{l+1} &= \sigma\big(A_l v_l + b_l\big), \quad l = 0, \dots, L-1 \\ y &= A_L v_L + b_L \end{aligned}$$

$$\sigma : \mathbb{R} \to \mathbb{R}, \quad A_l \in \mathbb{R}^{d_{l+1} \times d_l}, \quad b_l \in \mathbb{R}^{d_{l+1}}$$

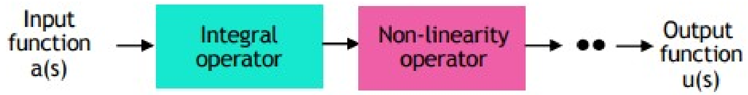
Slide source: {2}

Neural Networks vs Neural Operators

Neural operators

F: $a(s) \rightarrow u(s)$

$(R^n \rightarrow R^m) \rightarrow (R^n \rightarrow R^m)$



$$v_0(s) = P(x(s), s)$$

$$v_{l+1}(s) = \sigma\left(W_l v_l(s) + \int_D \kappa_l(s,z) v_l(z)\, dz + b_l(s)\right), \quad l = 0, \ldots, L-1$$

$$y(s) = Q(v_L(s))$$

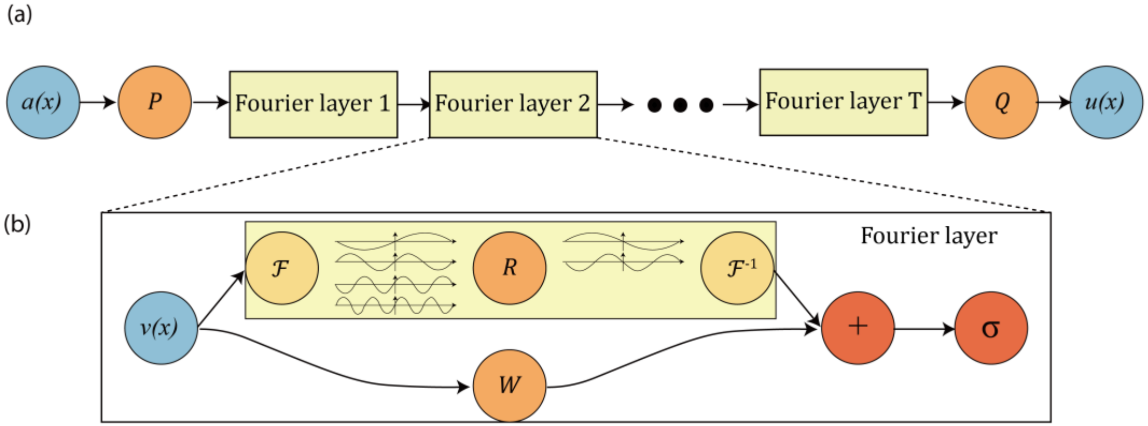
$$P : \mathbb{R}^{m+n} \rightarrow \mathbb{R}^{d_0}, \quad W_l \in \mathbb{R}^{d_{l+1} \times d_l}, \quad \kappa_l : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{d_{l+1} \times d_l}, \quad b_l : \mathbb{R}^n \rightarrow \mathbb{R}^{d_{l+1}}, \quad Q : \mathbb{R}^L \rightarrow \mathbb{R}^r$$

Slide source: {2}

Fourier Neural Operators

$$\kappa_\phi(x, y) = \kappa_\phi(x - y)$$

$$\mathcal{F}^{-1}\Big(R_\phi \cdot (\mathcal{F} v_t)\Big)(x)$$

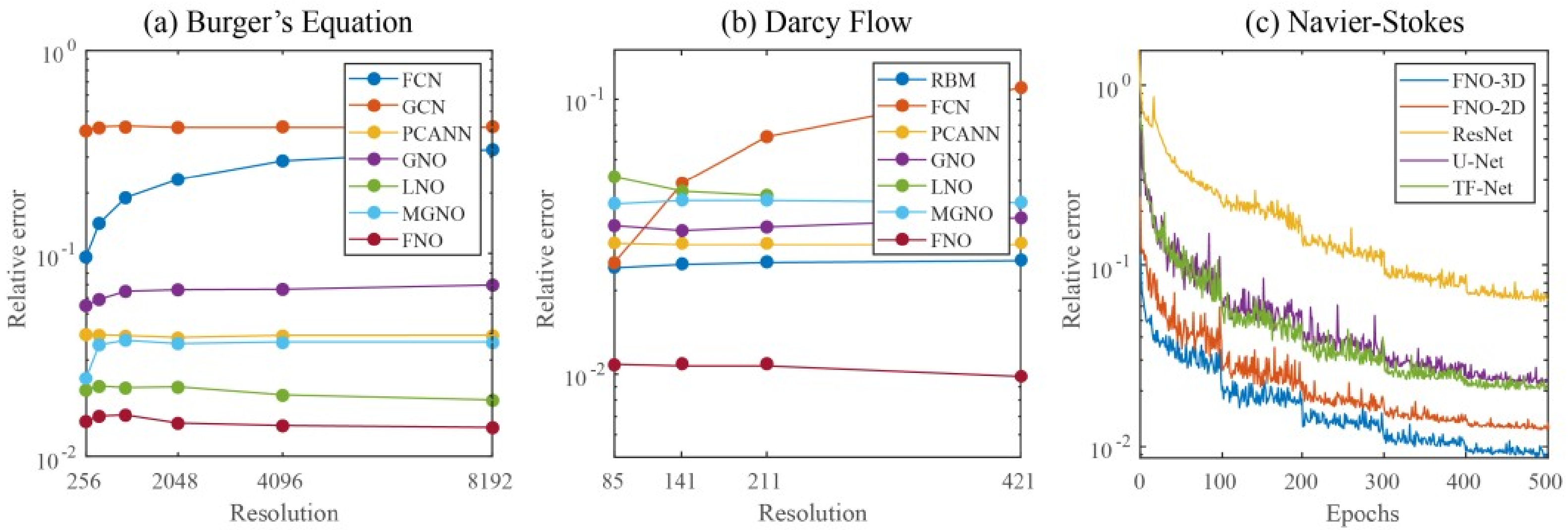


(a) The full architecture of neural operator: start from input a. 1. Lift to a higher dimension channel space by a neural network P. 2. Apply four layers of integral operators and activation functions. 3. Project back to the target dimension by a neural network Q. Output u. **(b) Fourier layers**: Start from input v. On top: apply the Fourier transform $\mathcal{F}$; a linear transform R on the lower Fourier modes and filters out the higher modes; then apply the inverse Fourier transform $\mathcal{F}^{-1}$. On the bottom: apply a local linear transform W.

Figure 2: **top:** The architecture of the neural operators; **bottom:** Fourier layer.

Slide source: {3}

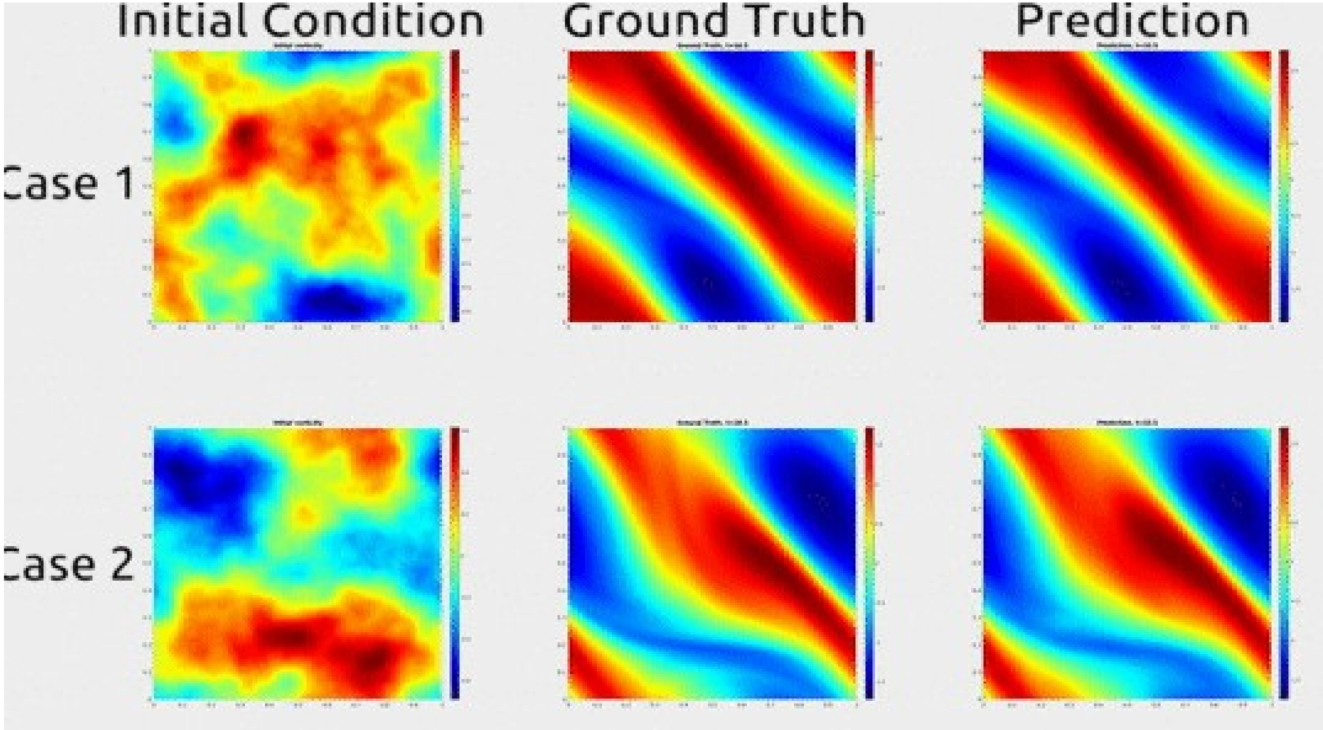
Fourier Neural Operators



Left: benchmarks on Burgers equation; **Mid:** benchmarks on Darcy Flow for different resolutions; **Right:** the learning curves on Navier-Stokes $\nu = 1e{-}3$ with different benchmarks. Train and test on the same resolution. For acronyms, see Section 5; details in Tables 1, 3, 4

Figure 3: Benchmark on Burger's equation, Darcy Flow, and Navier-Stokes

Navier-Stokes



Takeaways

- It is important to map between function spaces!
- Neural Networks are finite dimensional mappings, and Neural Operators are infinite dimensional mappings (of function spaces)!
- **“Fourier” Neural Operators?**
- **Neural Operators relate to Transformers?**

Resources and Links
[1]. Neural Operator
[2]. https://zongyi-li.github.io/neural-operator/Neural-Operator-CS159-0503-2022.pdf
[3]. https://github.com/zongyi-li/fourier_neural_operator