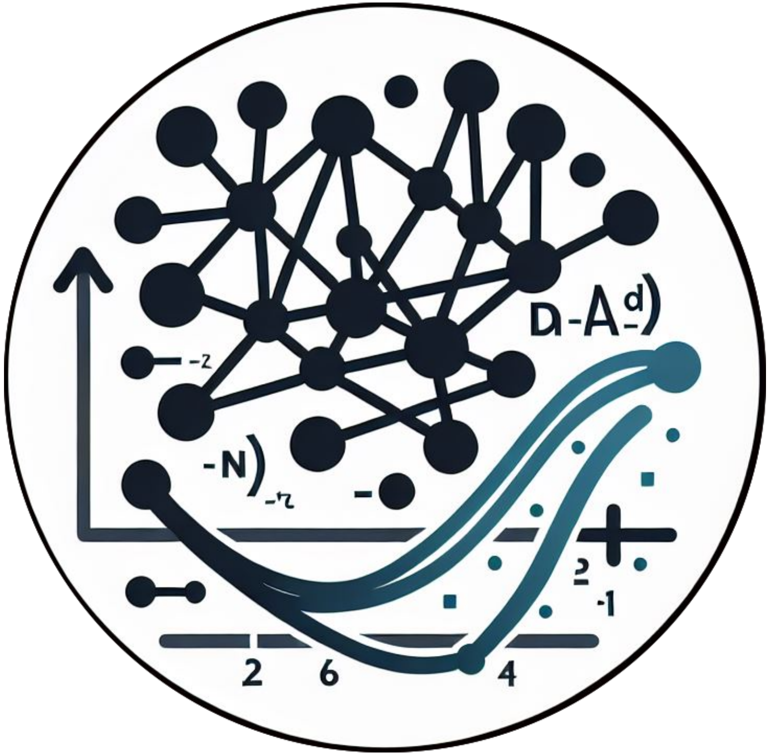
Fourier Neural Operators



Deep Learning for Engineers

Andrew Ning
aning@byu.edu

Fourier 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

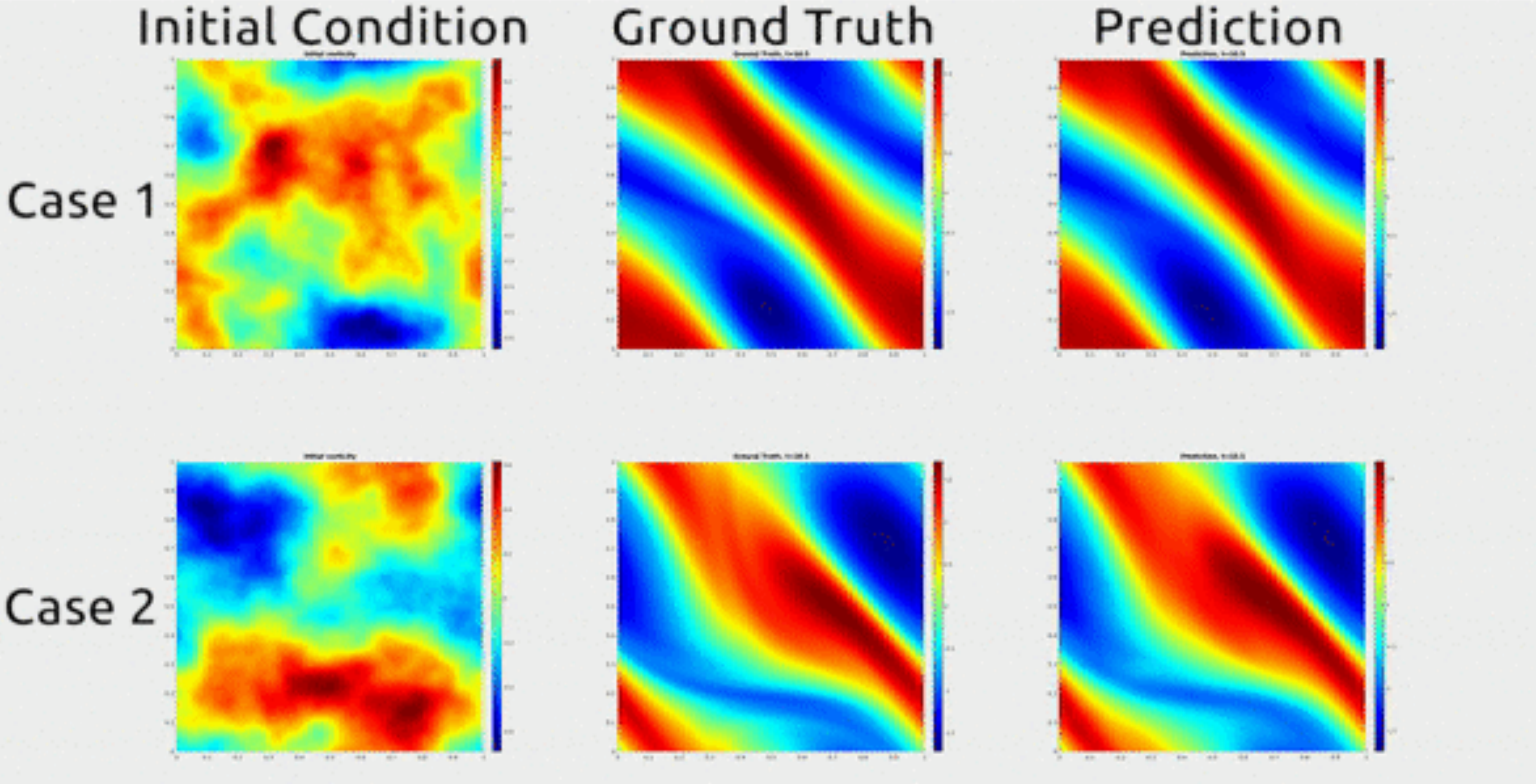
Andrew Stuart
astuart@caltech.edu

Anima Anandkumar
anima@caltech.edu

Abstract

The classical development of neural networks has primarily focused on learning mappings between finite-dimensional Euclidean spaces. Recently, this has been generalized to neural operators that learn mappings between function spaces. For partial differential equations (PDEs), neural operators directly learn the mapping

input		output	example
x	function $x \mapsto f(x)$	y	$x \mapsto x^2$
f	functional $f \mapsto f(\cdot)$	y	$f \mapsto \int_a^b f(x)\,dx$
f	operator $f \mapsto g$	g	$f \mapsto \nabla f$



Initial Condition
Ground Truth
Prediction
Case 1
Case 2

Linear Boundary Value Problems

$$Lu = f$$

linear operator.

$$L = a(x)\frac{d^2}{dx^2} + b(x)\frac{d}{dx} + c(x)$$

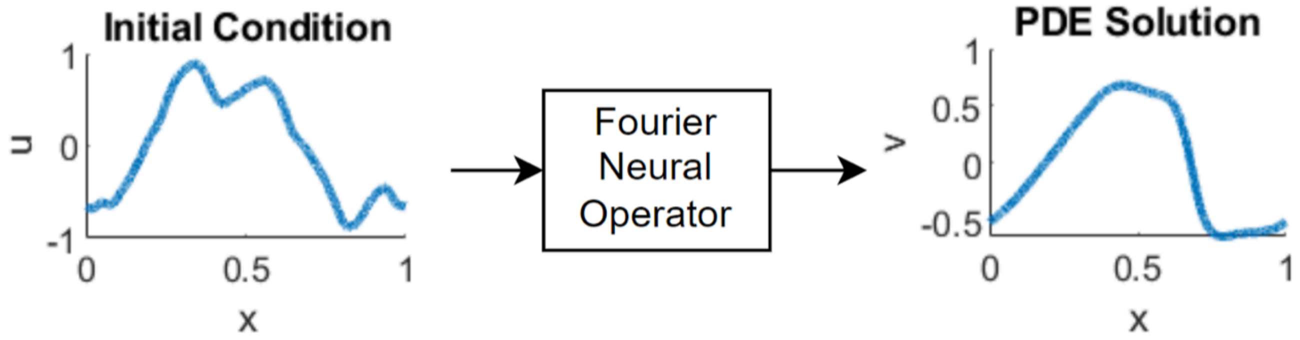
Solution for Linear PDE or BVP

$$Lu = f$$

$$u(x) = \int_D G(x,y) f(y)\, dy$$

Examples: Burgers

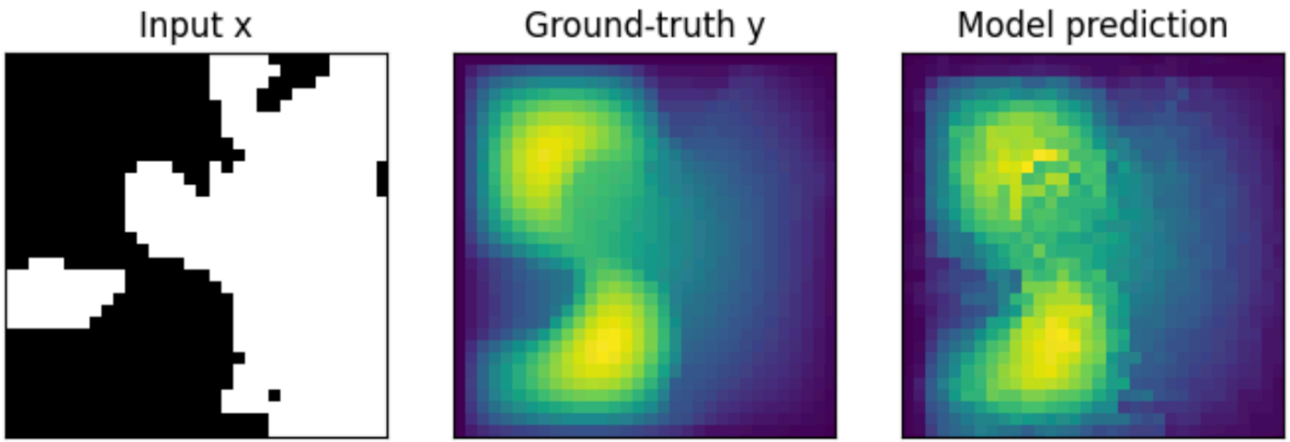
$$\partial_t u(x,t) + \partial_x(u^2(x,t)/2) = \nu\partial_{xx}u(x,t), \qquad x \in (0,1), t \in (0,1]$$
$$u(x,0) = u_0(x), \qquad x \in (0,1)$$



https://www.mathworks.com/help/deeplearning/ug/solve-pde-using-fourier-neural-operator.html

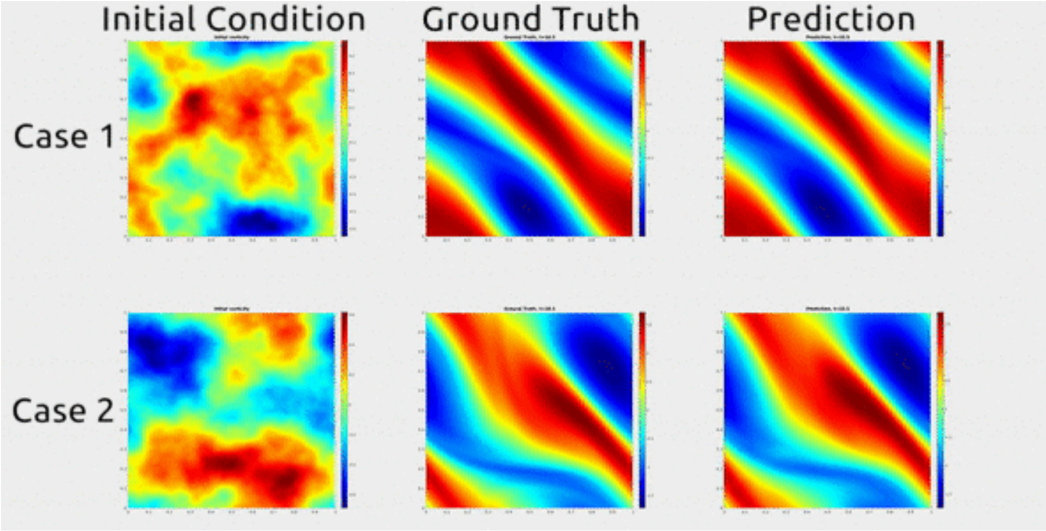
Examples: Darcy

$$-\nabla \cdot (a(x)\nabla u(x)) = f(x) \qquad x \in (0,1)^2$$
$$u(x) = 0 \qquad x \in \partial(0,1)^2$$



Examples: Navier Stokes

$$
\begin{aligned}
\partial_t w(x,t) + u(x,t)\cdot\nabla w(x,t) &= \nu\Delta w(x,t) + f(x), && x\in(0,1)^2, t\in(0,T] \\
\nabla\cdot u(x,t) &= 0, && x\in(0,1)^2, t\in[0,T] \\
w(x,0) &= w_0(x), && x\in(0,1)^2
\end{aligned}
$$



Motivate form of a Neural Net solution

$$u(x) = \int_D G(x,y) f(y)\, dy$$

$a(x)$ → $v_0 \to v_1 \mapsto v_2 \mapsto \cdots \mapsto v_n$ → $u(x)$

$$v_{t+1}(x) = \int_D K_\phi(x,y)\, v_t(y)\, dy$$

(K_ϕ ← N.N.)

$$v_{t+1}(x) = \sigma\left(W v_t(y) + \int_D K_\phi(x,y)\, v_t(y)\, dy \right)$$

(σ ← activation function)

$a(x) \rightarrow \cdots \rightarrow u(x)$

$v_0 \rightarrow v_1 \cdots v_T$

lift to higher dimension: $a(x) \xrightarrow{NN} v_0$

project back to a lower dimension: $v_T \xrightarrow{NN}$

not an autoencoder

1x1 convolution

Fourier Transform

assume

$$\int_D K_\theta(x,y)\,v_t(y)\,dy = \int K_\theta(x-y)\,v_t(y)\,dy$$

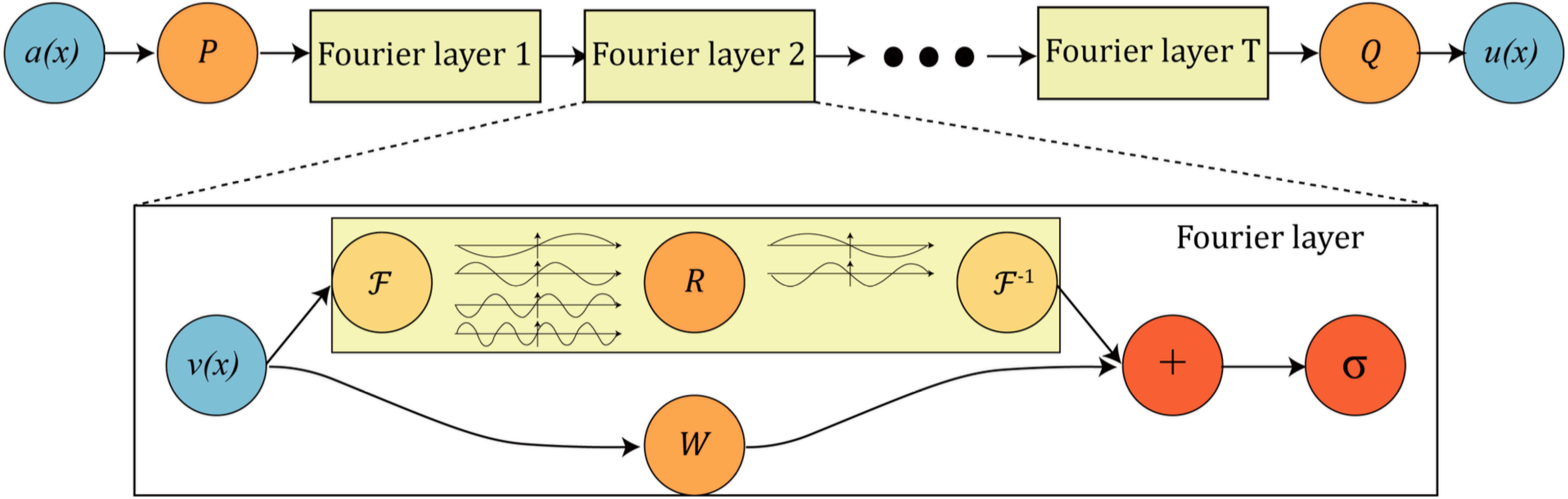
$$= K_\theta * v_t$$

← convolution

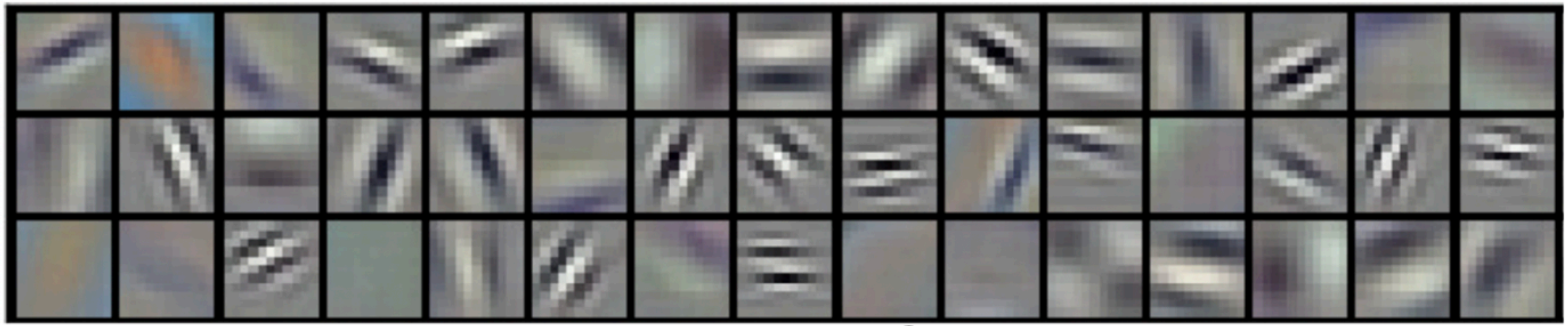
$$= \mathcal{F}^{-1}\left(\mathcal{F}(K_\theta)\cdot\mathcal{F}(v_t)\right)$$

$$= \mathcal{F}^{-1}\left(R_\theta\cdot\mathcal{F}(v_t)\right)$$

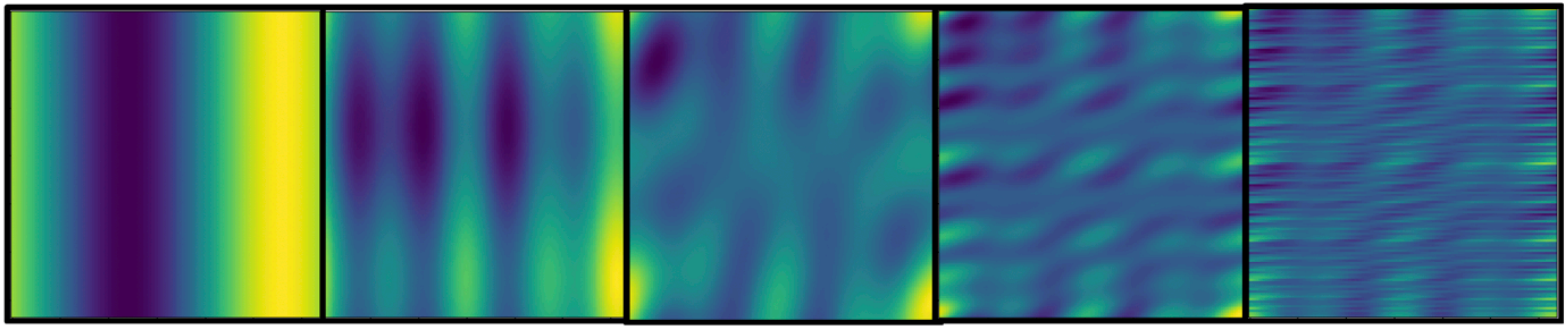
Summary



a(x)
P
Fourier layer 1
Fourier layer 2
Fourier layer T
Q
u(x)
Fourier layer
v(x)
$\mathcal{F}$
R
$\mathcal{F}^{-1}$
W
+
σ



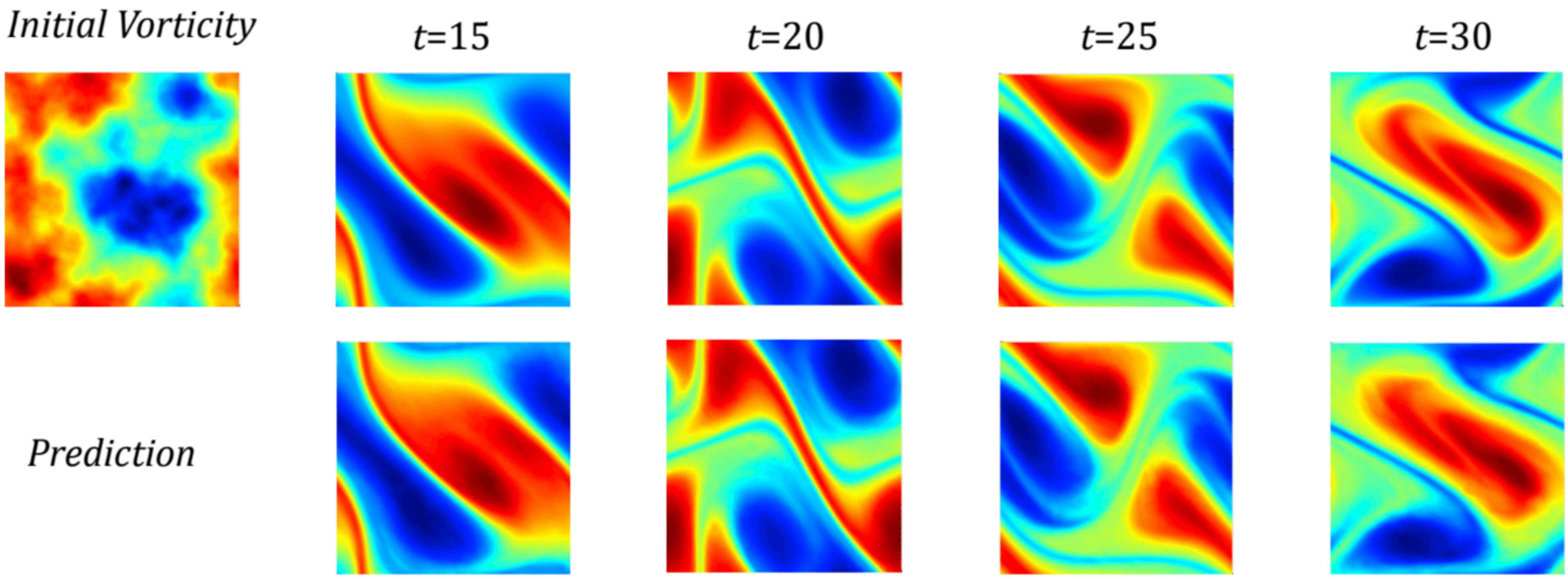
Filters in CNN



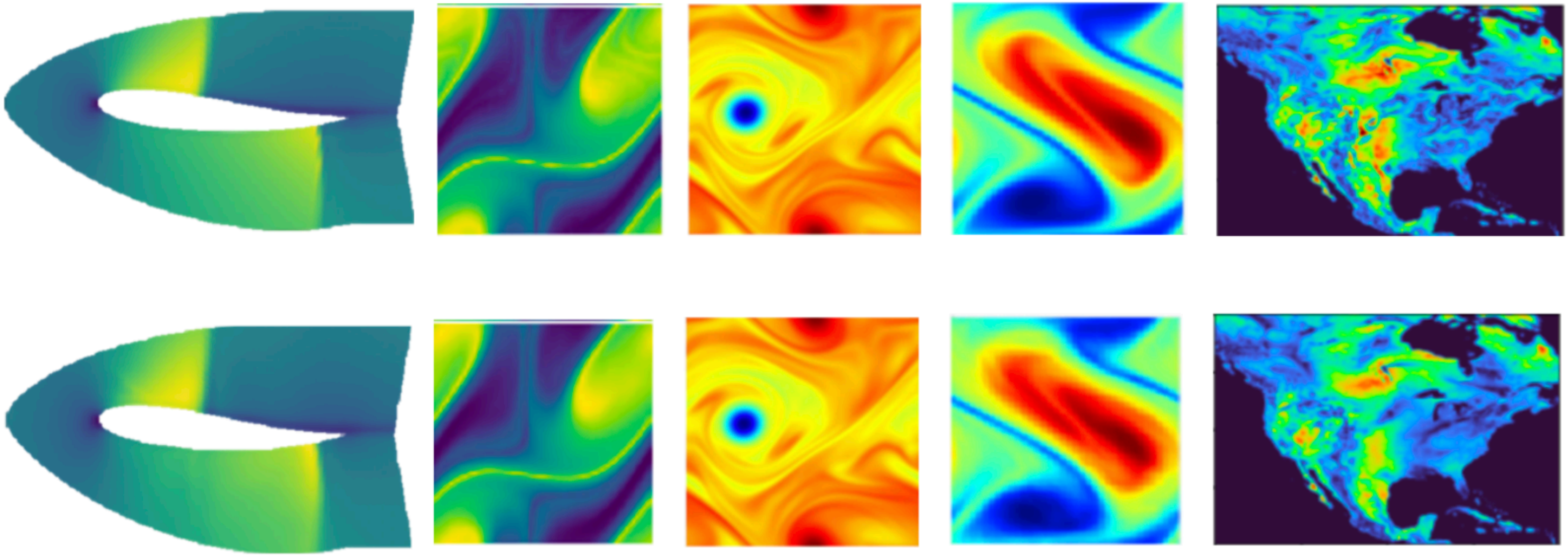
Fourier Filters

https://zongyi-li.github.io/blog/2020/fourier-pde/

Zero-shot Super Resolution

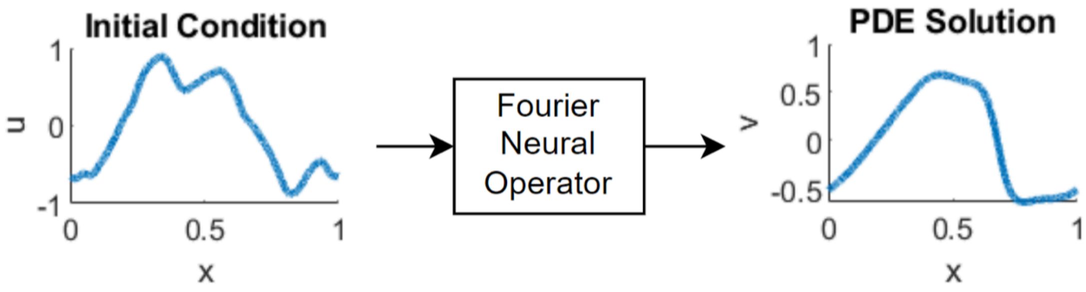


Many Other Variants



Airfoil Flow (Geo-FNO) | Darcy Flow (GANO) | Kolmogorov Flow (PINO) | Navier Stokes (FNO) | Weather model (FourcastNet)

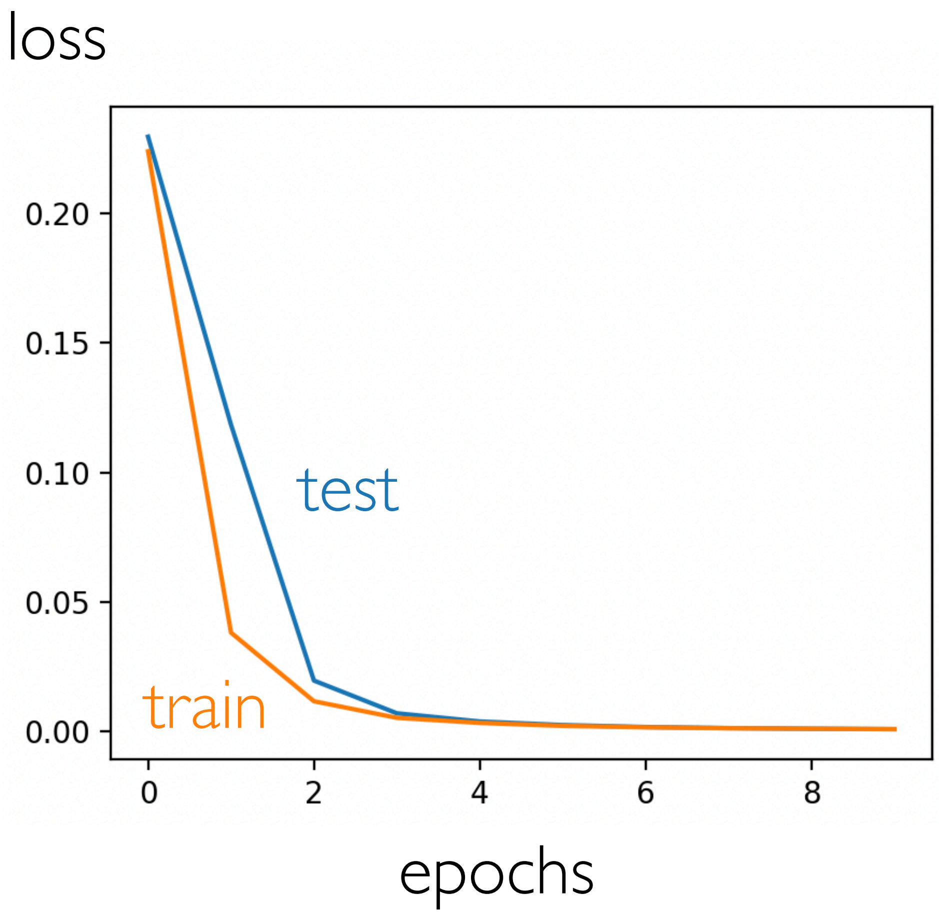
Example

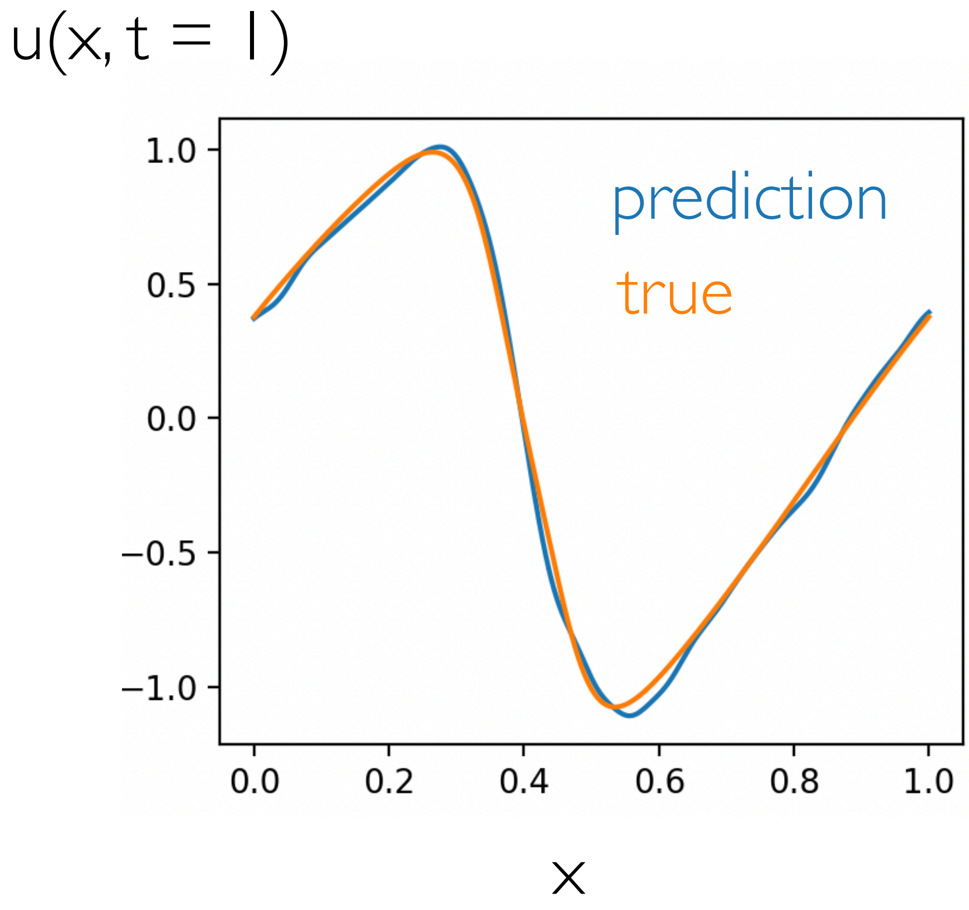


from neuralop.models import FNO

n_modes = 8
in_channels=1
out_channels=1,
hidden_channels=24,
n_layers=5,

train and test on coarse grid
(16 pts from 0 to 1)





16 pt resolution

8192 pt resolution