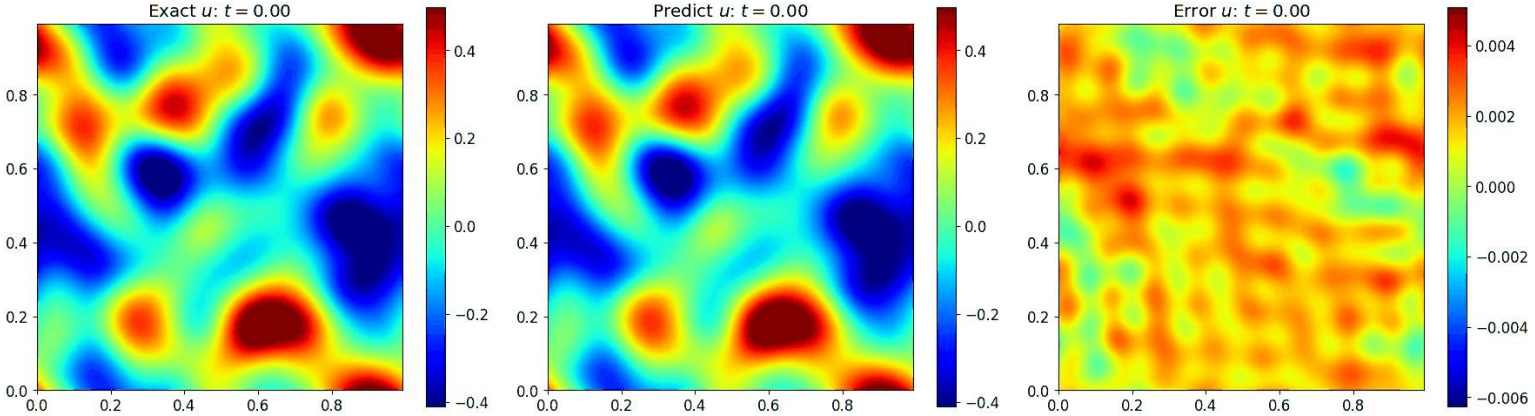
# Physics-Informed Machine Learning



# Administrivia

- **Friday, May 15th** (Today): W2
- **Friday, May 22nd**: Project Milestone Report
- **Monday, May 25th**: A4

# Outline

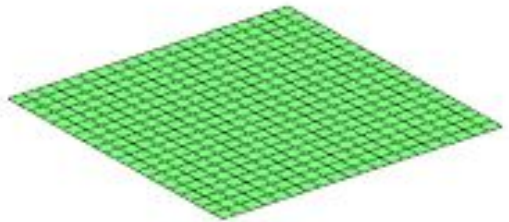
- **Background/Motivation**
- Physics-Informed Neural Networks (PINNs)
- Fourier Neural Operators (FNOs)
- Physics-Informed Neural Operators (PINOs)
- Frontiers of PIML Research

# Why care about Physics-Informed Machine Learning (PIML)?

# Engineering *is* Partial Differential Equations

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

2D Wave Equation



# Solving PDEs is hard

To solve the equation on a rectangular domain $0 \leq x \leq a$ and $0 \leq y \leq b$ with fixed boundaries ($u = 0$ at edges), we assume a product solution:

$$u(x, y, t) = X(x)Y(y)T(t)$$

Substituting this into the wave equation and dividing by $c^2XYT$ **separates the variables** into three ordinary differential equations (ODEs):

1. Spatial Parts: $X''(x) = -k_x^2 X(x)$ and $Y''(y) = -k_y^2 Y(y)$. For fixed boundaries, the solutions are sine function:

$$X_m(x) = \sin\left(\frac{m\pi x}{a}\right) \text{ for } m = 1, 2, 3, \ldots, \qquad Y_n(y) = \sin\left(\frac{n\pi y}{b}\right) \text{ for } n = 1, 2, 3, \ldots$$

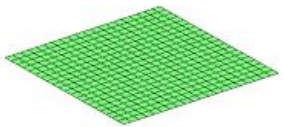
2. Temporal Part:

$$T''(t) = -c^2\lambda_{mn}T(t) \text{ where } \lambda_{mn} = (k_x^2 + k_y^2) = \pi^2\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)$$

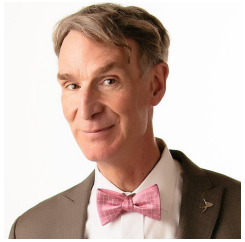
3. General Solution: The full displacement is a **superposition** of these modes

$$\boxed{u(x, y, t) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \sin\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{b}\right)\left[A_{mn}\cos(\omega_{mn}t) + B_{mn}\sin(\omega_{mn}t)\right]}$$

for $\omega_{mn} = c\sqrt{\lambda_{mn}}$ the **natural frequencies** of vibration.



# like… *really* hard



$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}-\frac{1}{R_e}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=0,$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}-\frac{1}{R_e}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)=0,$$

$$u(x,y,0)=u_0(x,y)=\sin(\pi x)\cos(\pi y),$$
$$v(x,y,0)=v_0(x,y)=\cos(\pi x)\sin(\pi y),$$

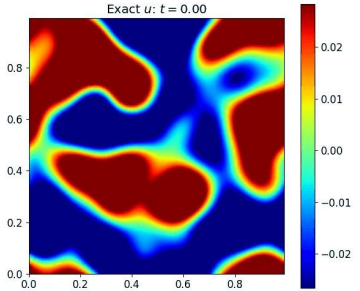
$$u(0,y,t)=u(1,y,t)=v(x,0,t)=v(x,1,t)=0.$$

Then, according to Hopf–Cole transformation, the solution of the Burgers' equation can be written as:

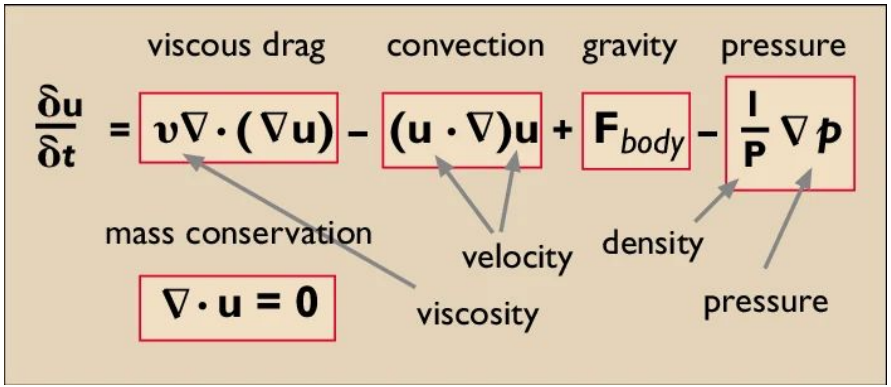
$$u(x,y,t)=2\pi\mu\frac{\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}nA_{mn}C_{nm}E_{mn}(t)\sin(n\pi x)\cos(m\pi y)}{\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}A_{mn}C_{nm}E_{mn}(t)\cos(n\pi x)\cos(m\pi y)},$$

$$v(x,y,t)=2\pi\mu\frac{\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}mA_{mn}C_{nm}E_{mn}(t)\cos(n\pi x)\sin(m\pi y)}{\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}A_{mn}C_{nm}E_{mn}(t)\cos(n\pi x)\cos(m\pi y)}.$$

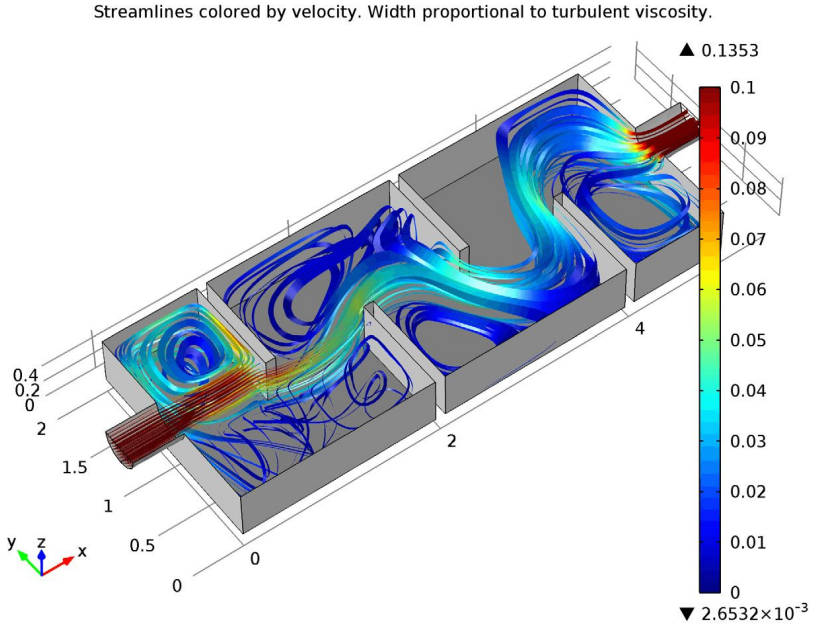
Gao Q, "An analytical solution for two and three dimensional nonlinear Burgers' equation", AMM 2017



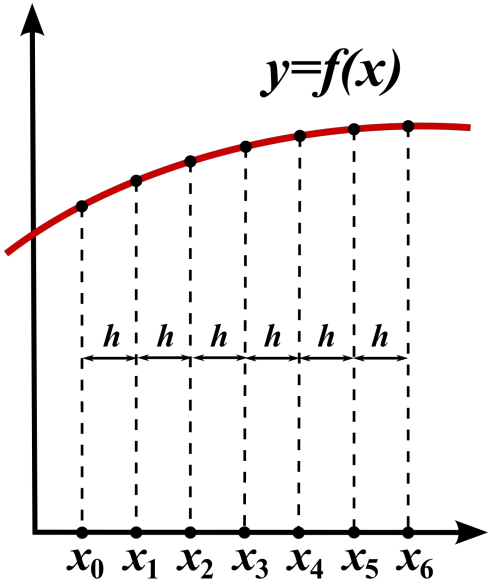
# and oftentimes *impossible*



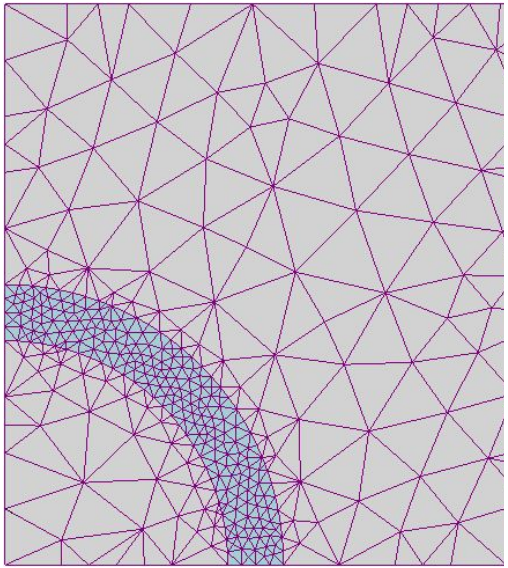

Navier-Stokes Equations



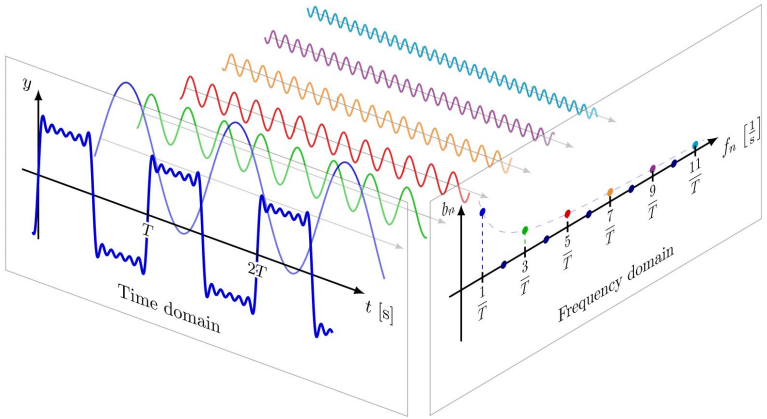
# Numerical Methods to the rescue!




Finite difference method



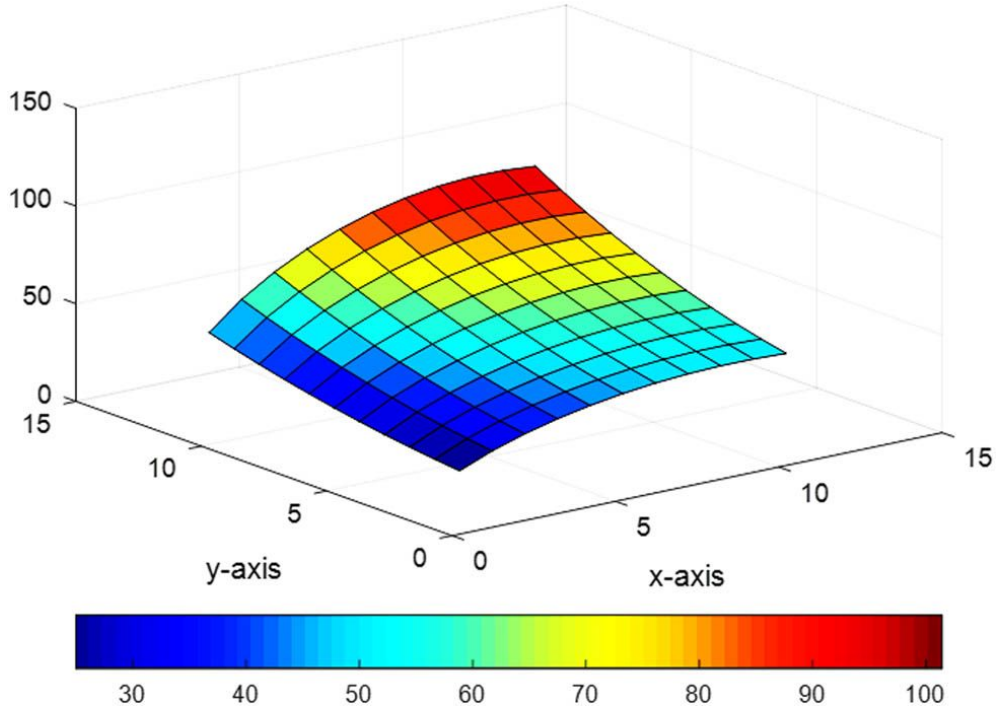
Finite element method



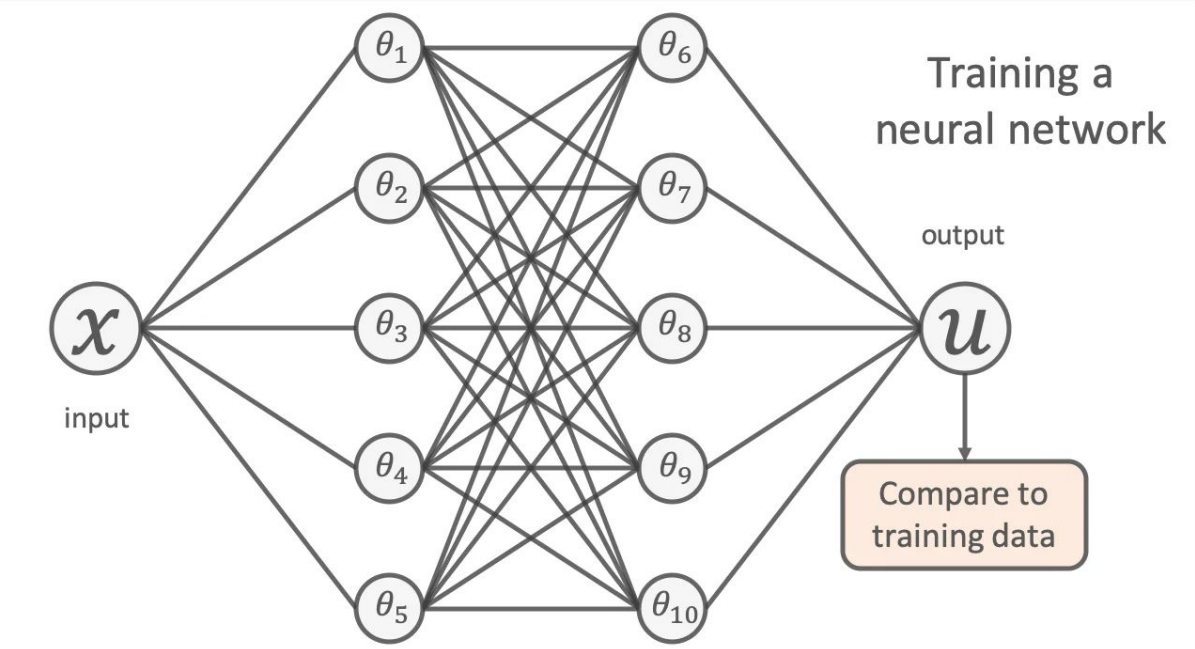

Spectral Methods

# Numerical Methods have problems

1. Slow
2. Discretization-Dependent
3. Inaccurate

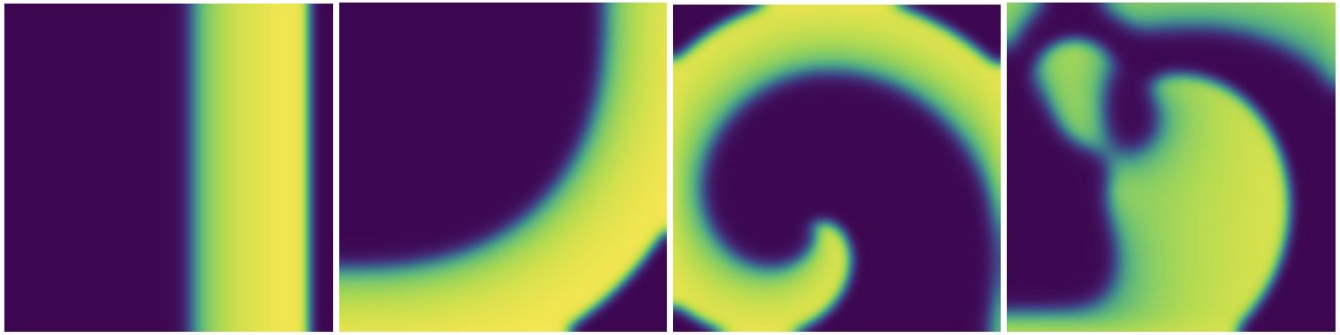


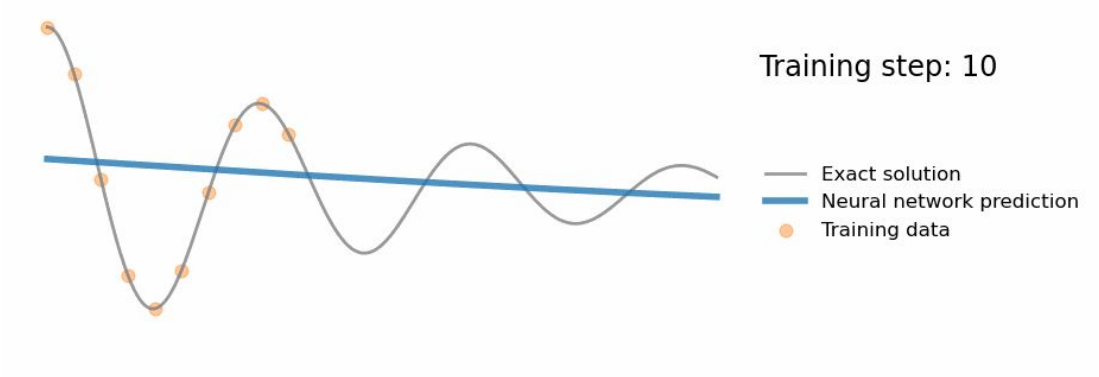
# Neural Networks to the rescue!



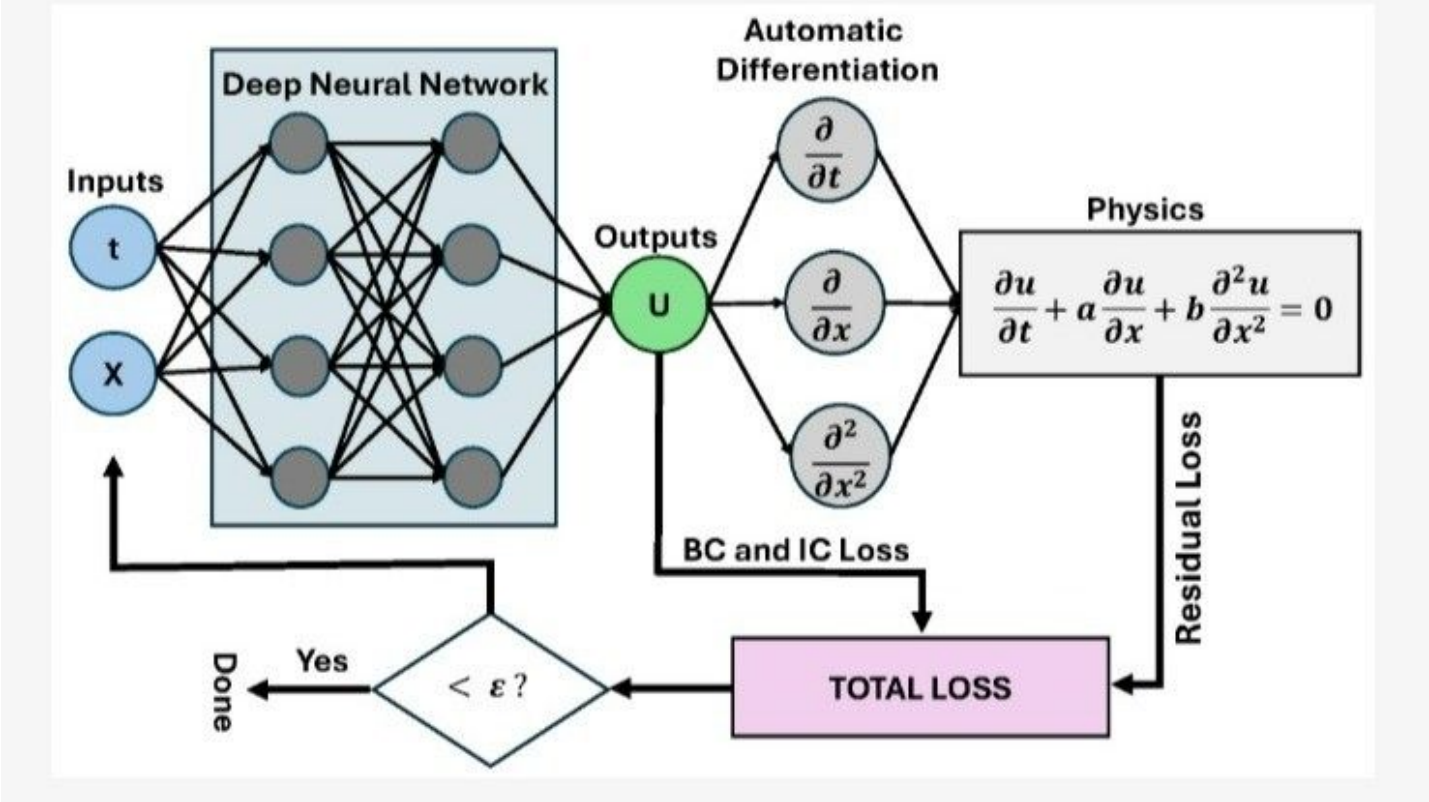

$$\min \frac{1}{N} \sum_{i}^{N} (u_{\mathrm{NN}}(x_i;\theta) - u_{\mathrm{true}}(x_i))^2$$

# Too much data…



# Physics to the rescue!



# Outline

- Background/Motivation
- **Physics-Informed Neural Networks (PINNs)**
- Fourier Neural Operators (FNOs)
- Physics-Informed Neural Operators (PINOs)
- Frontiers of PIML Research

Raissi et. al, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”, Journal of Computational Physics 2019

# Example: Damped Harmonic Oscillator

$\delta < \omega_0$, where $\delta = \frac{\mu}{2m}, \omega_0 = \sqrt{\frac{k}{m}}$

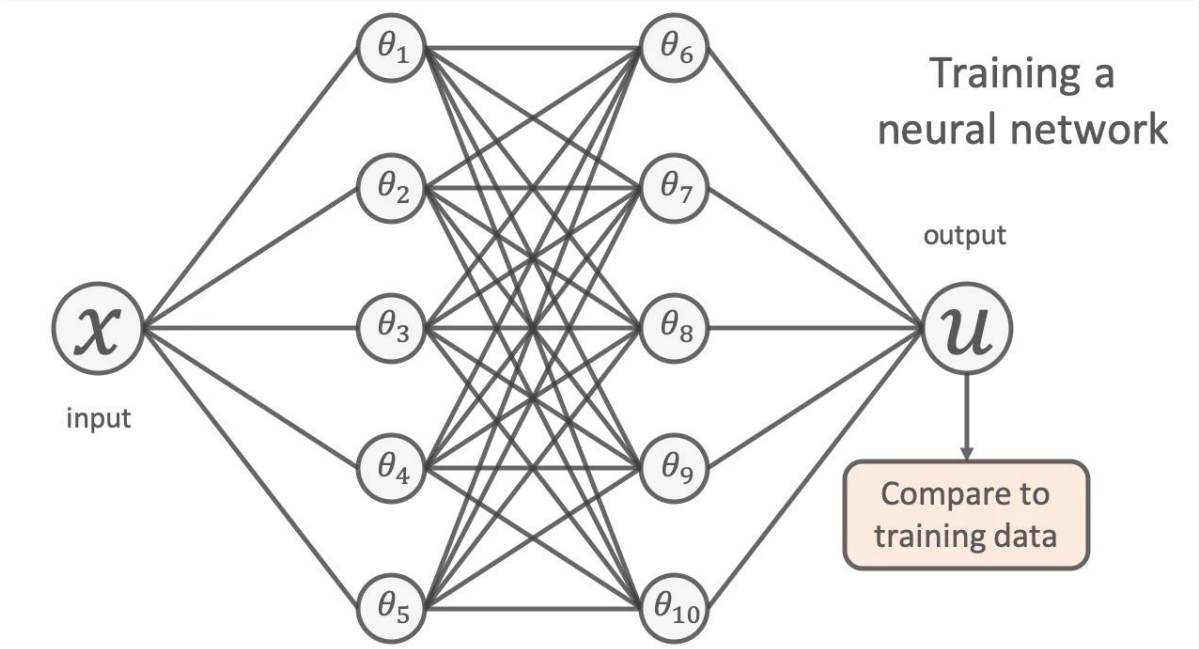
Initial condition: $u(t=0) = 1, \frac{du}{dt}(t=0) = 0$

$$m\frac{d^2u}{dx^2} + \mu\frac{du}{dx} + ku = 0$$

**Solution:** $u(t) = e^{-\delta t}(2A\cos(\phi + \omega t))$, with $\omega = \sqrt{\omega_0^2 - \delta^2}$

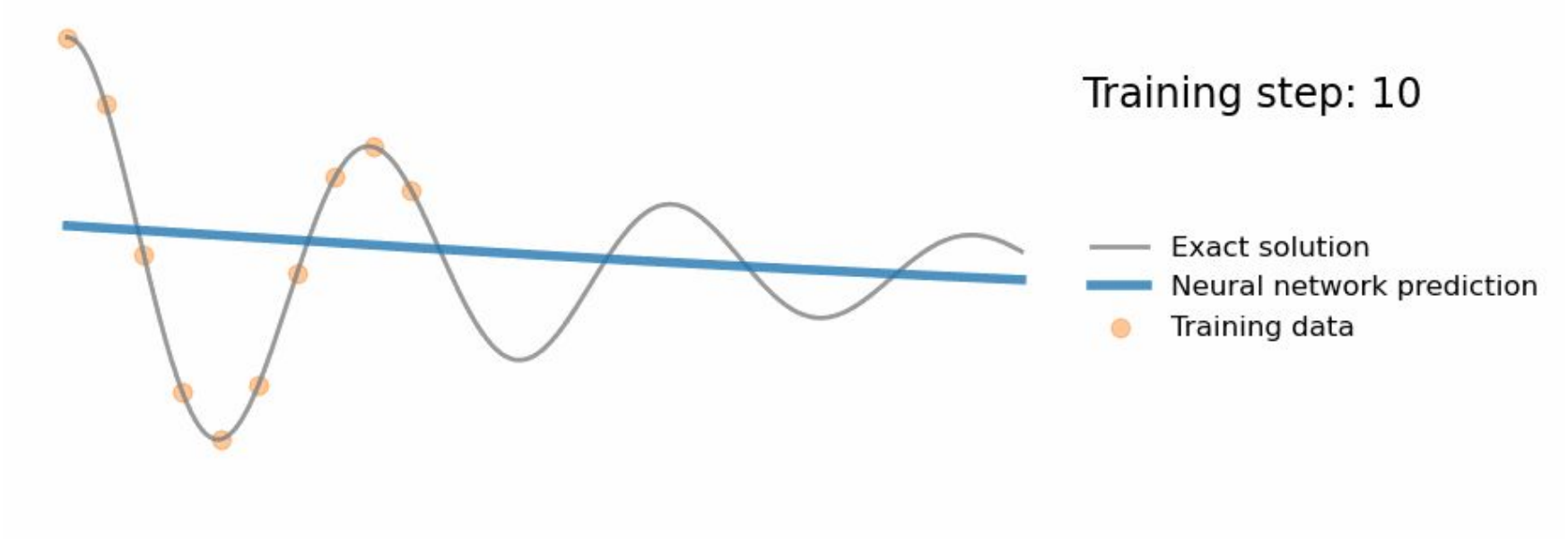
https://benmoseley.blog/my-research/so-what-is-a-physics-informed-neural-network/

# *Naive* Neural Network



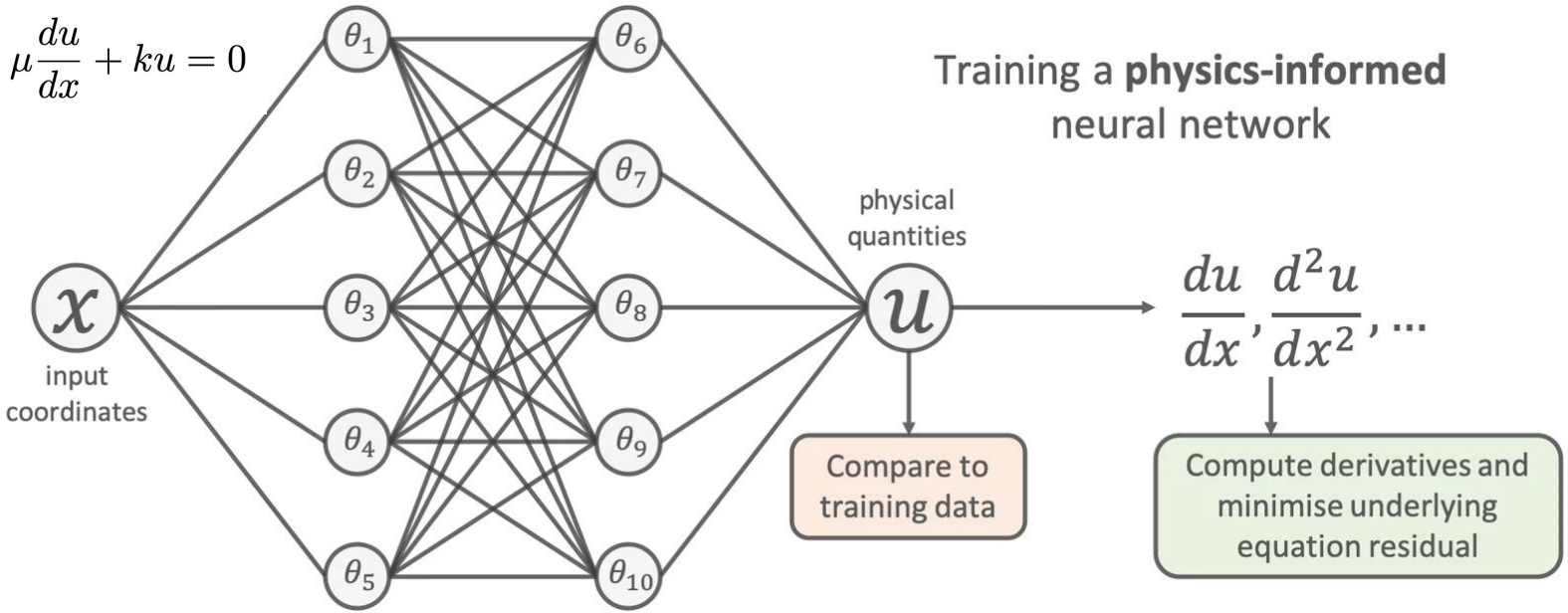

$$\min \frac{1}{N} \sum_{i}^{N} (u_{\mathrm{NN}}(x_i;\theta) - u_{\mathrm{true}}(x_i))^2$$

# Lack of Generalizability



# Idea: Add Physics to the Loss Function

$$m\frac{d^2u}{dx^2} + \mu\frac{du}{dx} + ku = 0$$



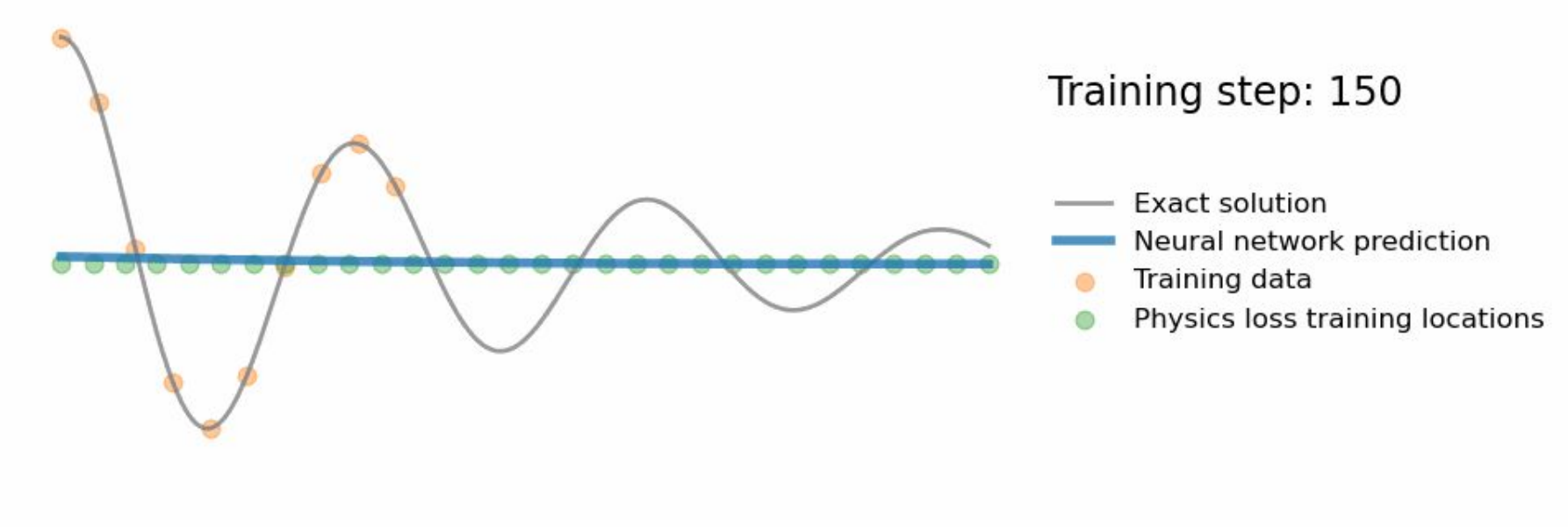

$$\min \frac{1}{N}\sum_{i}^{N}(u_{\mathrm{NN}}(x_i;\theta) - u_{\mathrm{true}}(x_i))^2 + \frac{1}{M}\sum_{j}^{M}\left(\left[m\frac{d^2}{dx^2} + \mu\frac{d}{dx} + k\right]u_{\mathrm{NN}}(x_j;\theta)\right)^2$$

# Code

```
model = FCN(1,1,32,3)
optimizer = torch.optim.Adam(model.parameters(),lr=1e-4)
files = []
for i in range(20000):
    optimizer.zero_grad()

    # compute the "data loss"
    yh = model(x_data)
    loss1 = torch.mean((yh-y_data)**2)# use mean squared error

    # compute the "physics loss"
    yhp = model(x_physics)
    dx  = torch.autograd.grad(yhp, x_physics, torch.ones_like(yhp), create_graph=True)[0]# computes dy/dx
    dx2 = torch.autograd.grad(dx,  x_physics, torch.ones_like(dx),  create_graph=True)[0]# computes d^2y/dx^2
    physics = dx2 + mu*dx + k*yhp# computes the residual of the 1D harmonic oscillator differential equation
    loss2 = (1e-4)*torch.mean(physics**2)

    # backpropagate joint loss
    loss = loss1 + loss2# add two loss terms together
    loss.backward()
    optimizer.step()
```

https://github.com/benmoseley/harmonic-oscillator-pinn

# Results


Training step: 150
Exact solution
Neural network prediction
Training data
Physics loss training locations

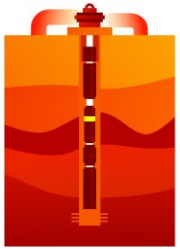
# Applications

## PINNs IN ACTION: VALUE FOR THE INDUSTRY

### CHEMICAL REACTOR

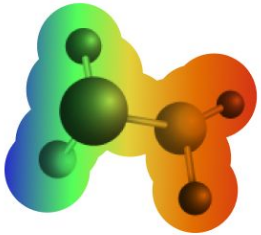
PINNs run as a part of a hybrid model for a chemical reactor running the Fischer-Tropsch process.

Two PINNs are used to:
- Solve an equation for reaction rates.
- Model the distribution of reactants inside a catalytic pellet.



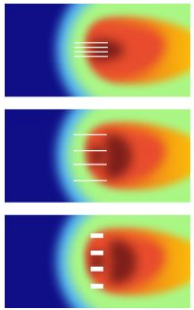
### ELECTRICAL SUBMERSIBLE PUMPS

A digital twin of ESP-lifted oil well, based on PINN formalism, takes advantage of combining available sensor data with fluid flow equations, and the approximative power of neural networks.



### MOLECULAR ELECTROSTATIC POTENTIAL

Our in-house PINN-based predictive pipeline is designed to produce a spatial distribution of molecular electrostatic potential for a wide range of organic molecules.

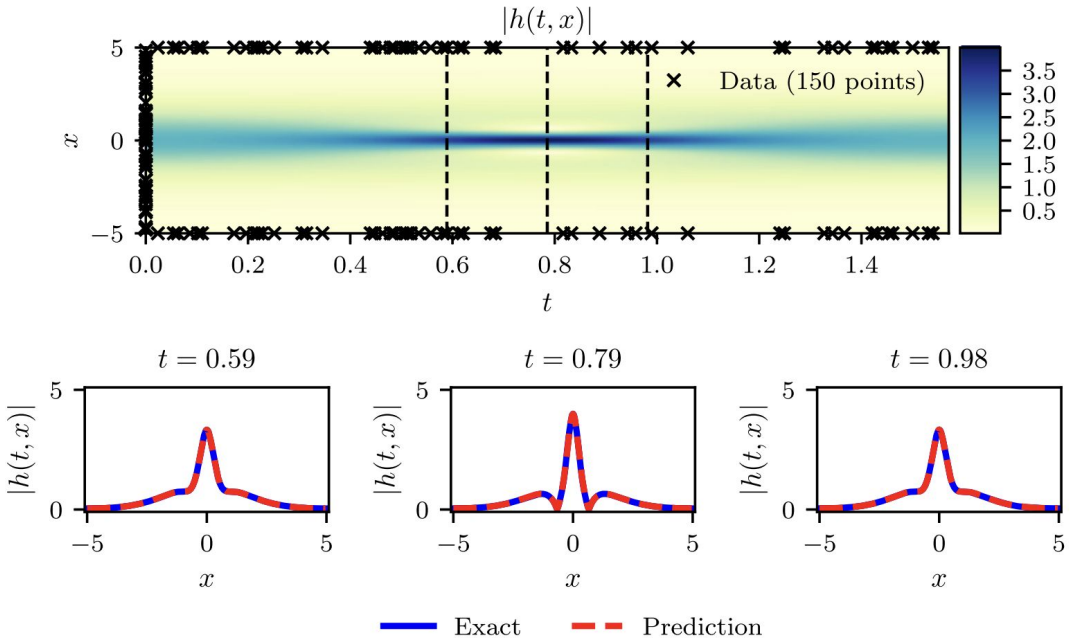


### HEAT SINK DESIGN

Our PINN-driven solution for heat sink design solves heat transfer equations and optimizes heat sink configuration (e.g., temperature minimization on a CPU).

# Benefits

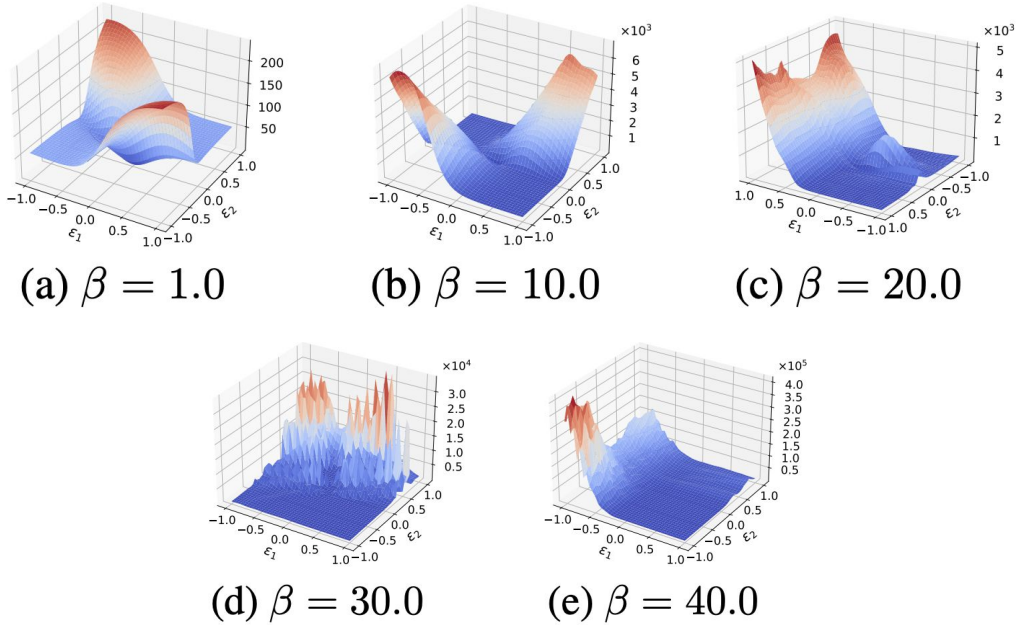
- Mesh-free
- Small data regime (or no data at all)
- Faster inference time (1000x)
- Simple/Intuitive → Extensible
- Inverse Problem




Raissi et. al, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”, Journal of Computational Physics 2019

# Drawbacks

- Encourages, not enforces physics
- Optimization landscape
- Sensitive to hyperparameters
- Traditional methods with grid can be faster
- Solution for one instance



(a) $\beta = 1.0$ (b) $\beta = 10.0$ (c) $\beta = 20.0$

(d) $\beta = 30.0$ (e) $\beta = 40.0$

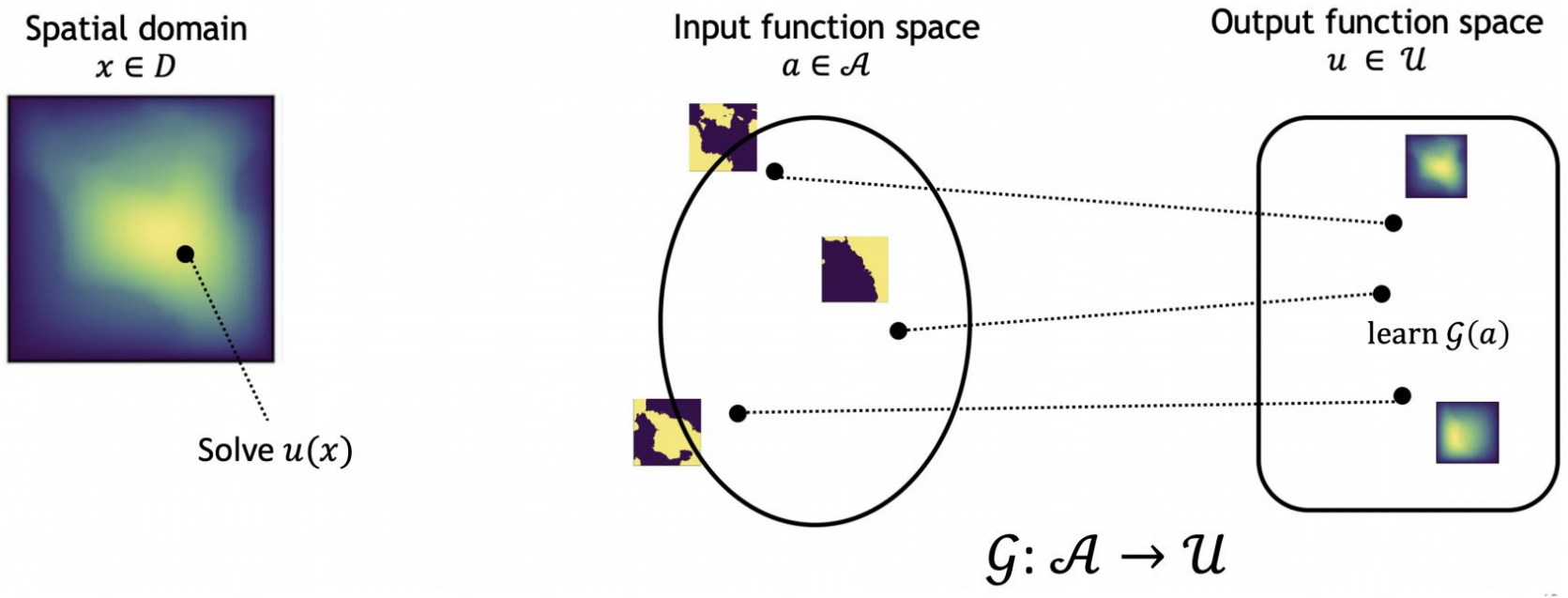
Mahoney et. al, “Characterizing possible failure modes in physics-informed neural networks”, NeurIPS 2021

# Outline

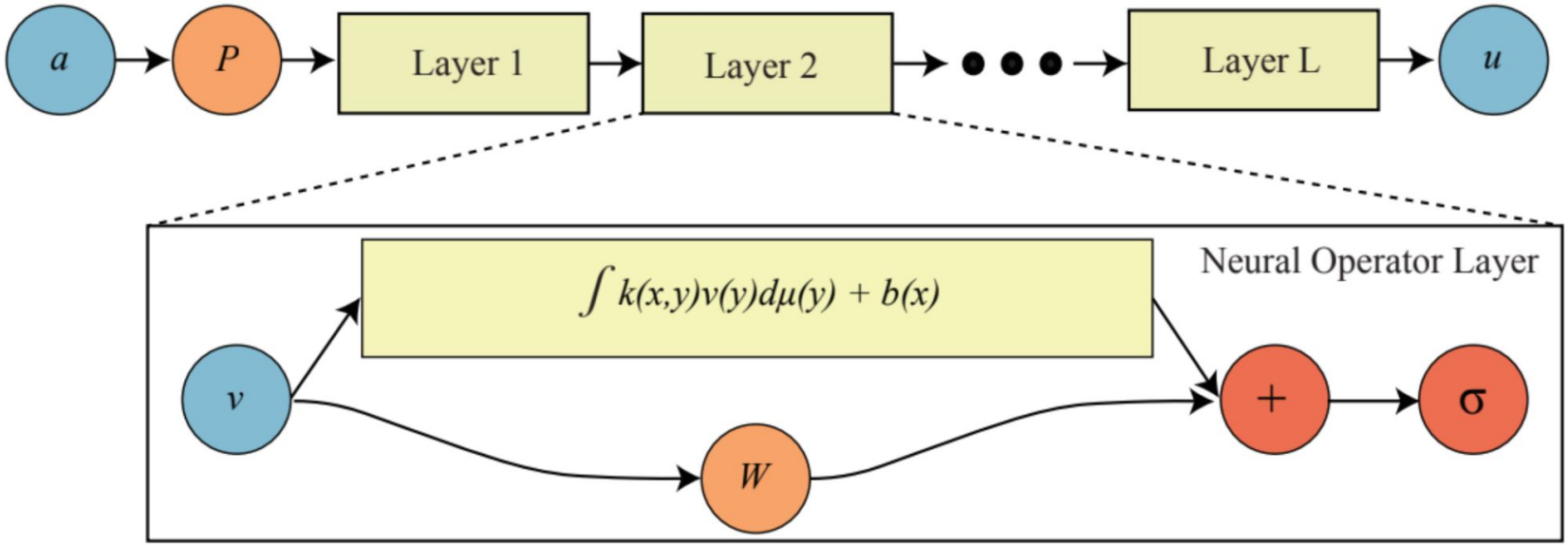
- Background/Motivation
- Physics-Informed Neural Networks (PINNs)
- **Fourier Neural Operators (FNOs)**
- Physics-Informed Neural Operators (PINOs)
- Frontiers of PIML Research

Anandkumar et. al, “Fourier Neural Operator for Parametric Partial Differential Equations”, ICML 2021

# A Brief Aside: Operator

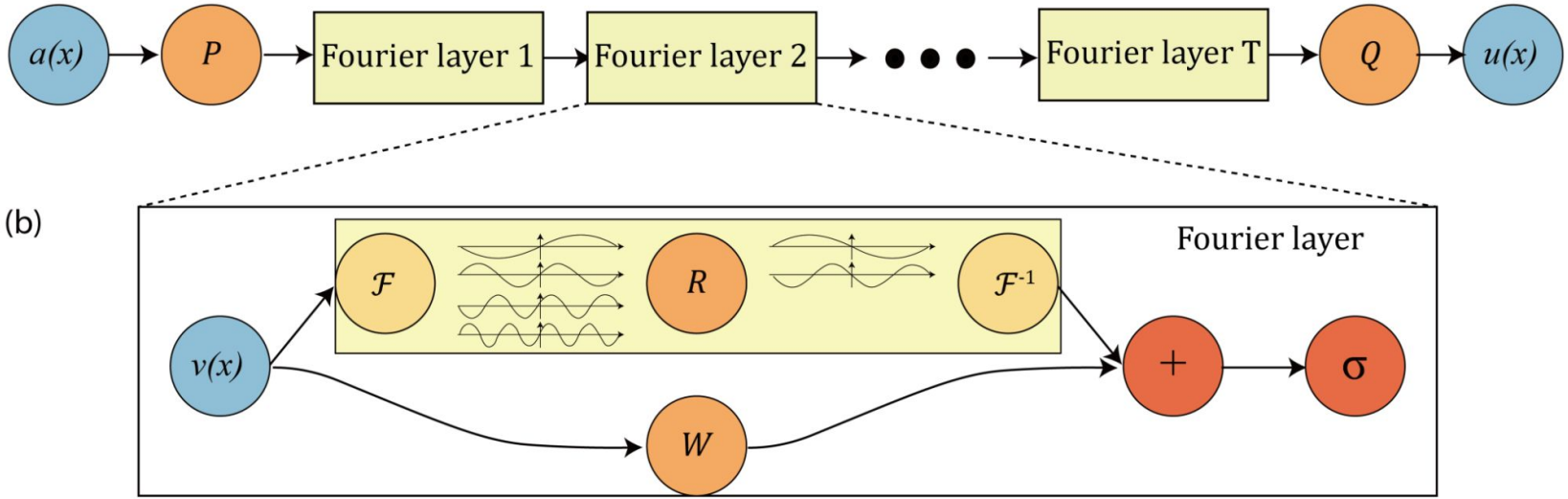


# Neural Operator




Anandkumar et. al, “Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs”, Journal of Machine Learning Research 2023

# Fourier Neural Operator

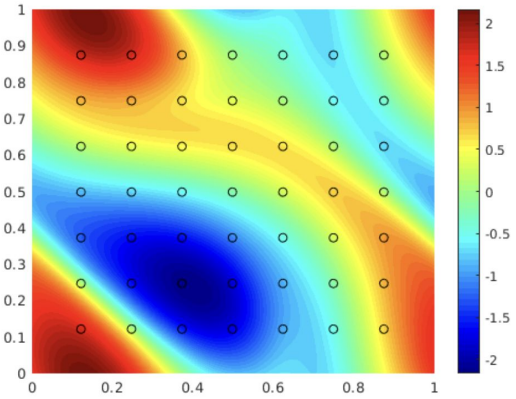


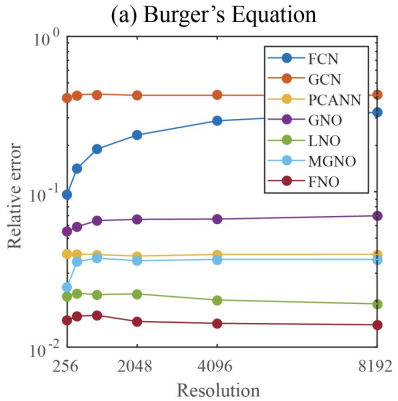

Anandkumar et. al, “Fourier Neural Operator for Parametric Partial Differential Equations”, ICML 2021

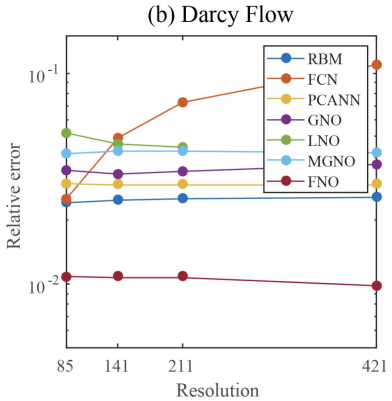
"Operator learning can be taken as an image-to-image problem. The Fourier layer can be viewed as a substitute for the convolution layer." - Zongyi Li

# Benefits: Speed + Resolution Invariance

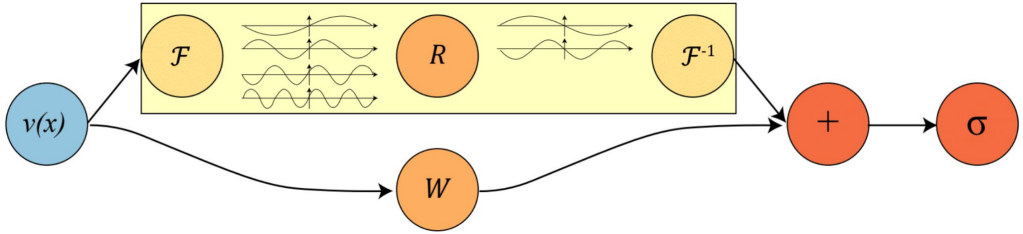
1. Speed!
   - On a 256 × 256 grid, the Fourier neural operator has an inference time of only 0.005s compared to the 2.2s of the pseudo-spectral method used to solve Navier-Stokes
2. Resolution-Invariant
3. Learn family of solutions



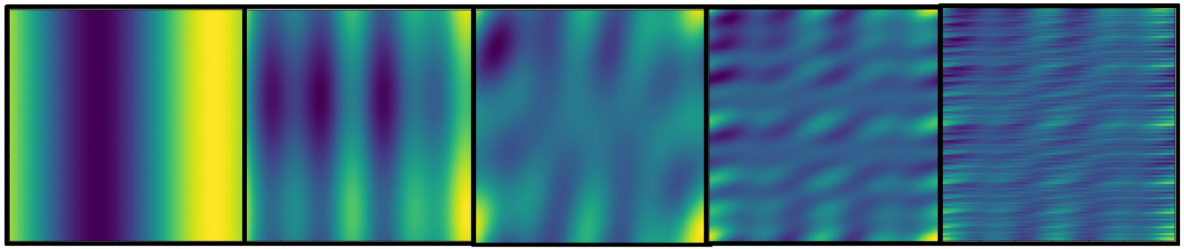



# Benefits: Global Features

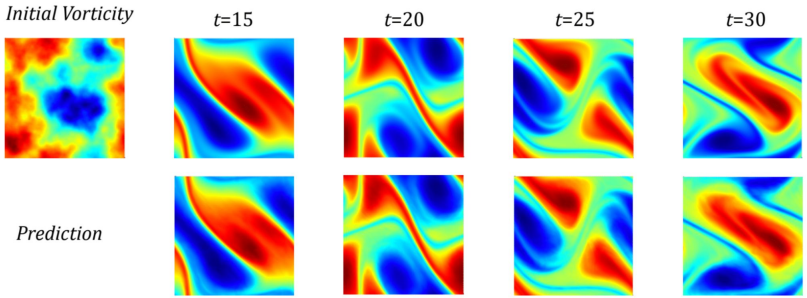



Filters in CNN

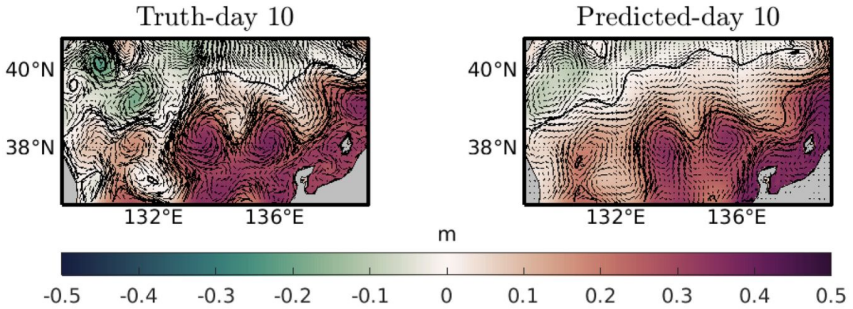


Fourier Filters

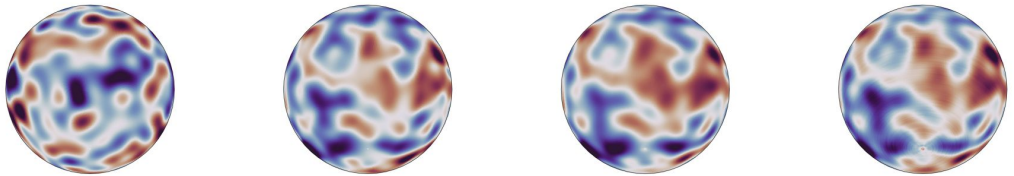
# Applications




Zero-shot super-resolution: Navier-Stokes Equation with viscosity v = 1e−4; Ground truth on top and prediction on bottom; trained on 64 × 64 × 20 dataset; evaluated on 256 × 256 × 80



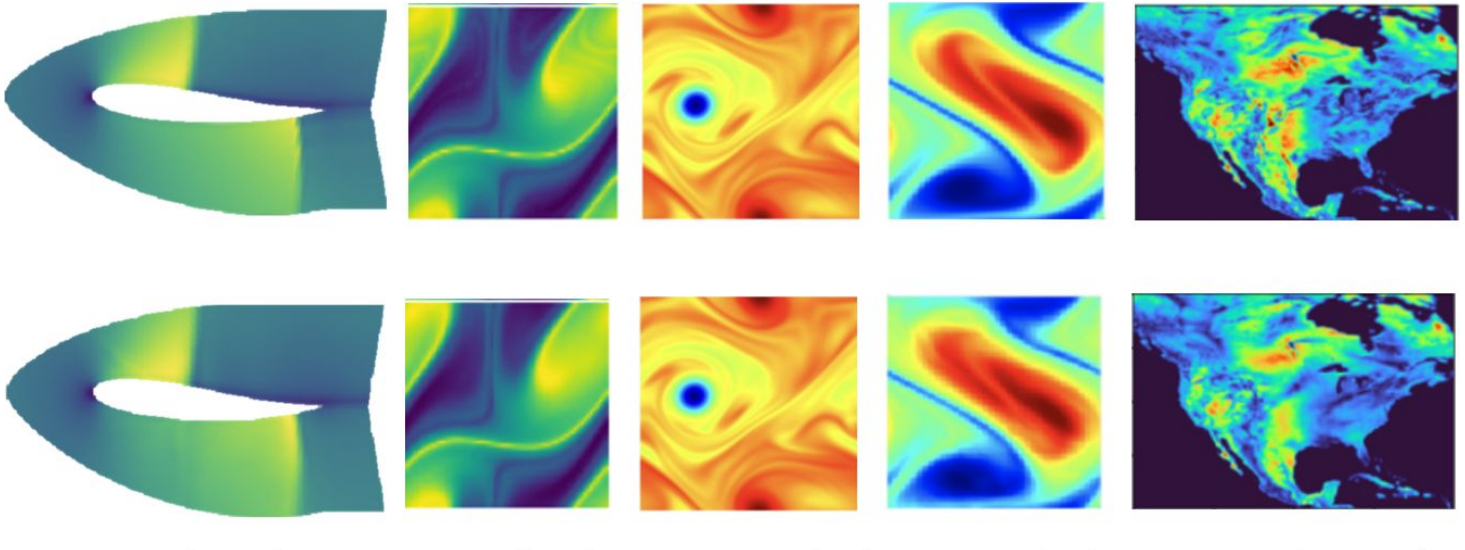

Choi et. al, "Applications of the Fourier neural operator in a regional ocean modeling and prediction", Physical Oceanography 2024



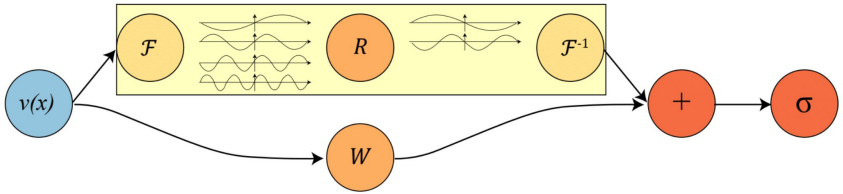
(a) initial condition (b) ground truth, $t = 5h$ (c) SFNO, $t = 5h$ (d) FNO, $t = 10h$

Anandkumar et. al, "Modelling Atmospheric Dynamics with Spherical Fourier Neural Operators", ICLR 2023

# Limitations -- Data Driven



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

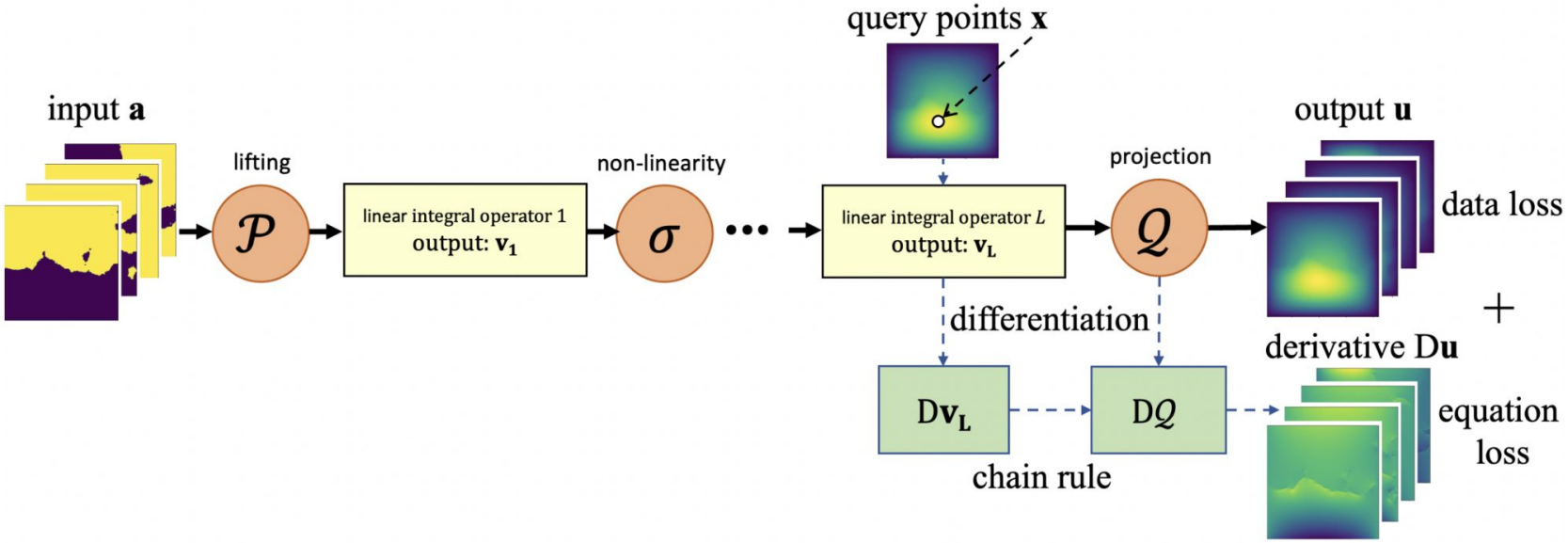


# Outline

- Background/Motivation
- Physics-Informed Neural Networks (PINNs)
- Fourier Neural Operators (FNOs)
- **Physics-Informed Neural Operators (PINOs)**
- Frontiers of PIML Research

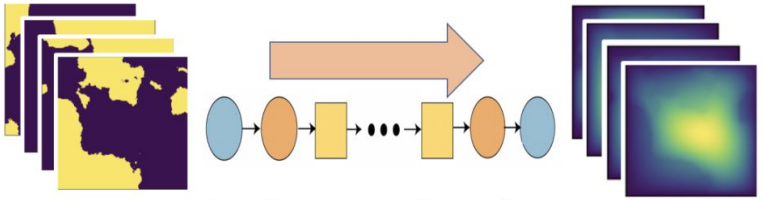
Anandkumar et. al, “Physics-Informed Neural Operator for Learning Partial Differential Equations”, ACM/IMS Journal of Data Science, 2024

# Physics-Informed Neural Operator (PINO)

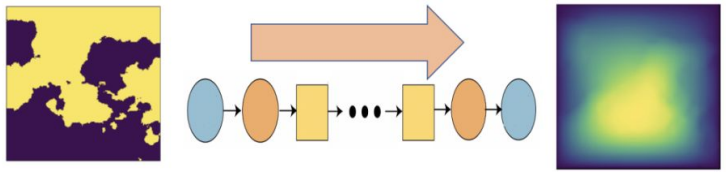



$$\mathcal{L} = \lambda_d \mathcal{L}_{\text{data}} + \lambda_p \mathcal{L}_{\text{pde}}^{\text{fine}} + \lambda_b \mathcal{L}_{\text{bc}}$$

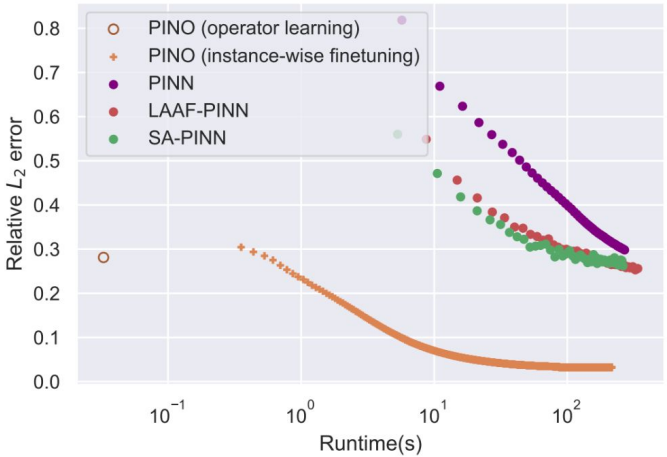
# Instance-wise Fine-tuning

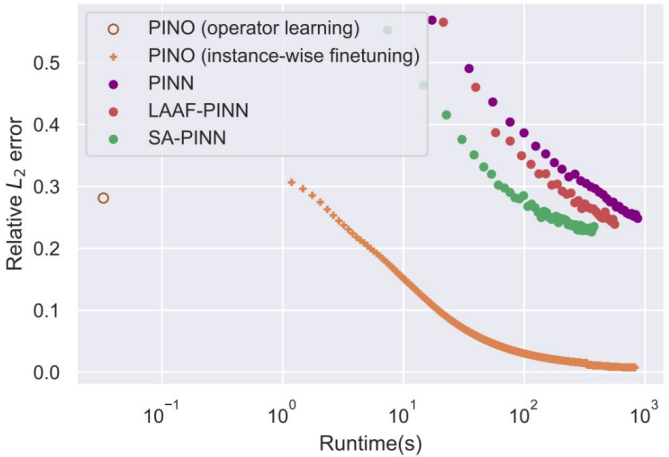


(a) Operator learning



(b) Test-time optimization



# Aside: Inverse Problem

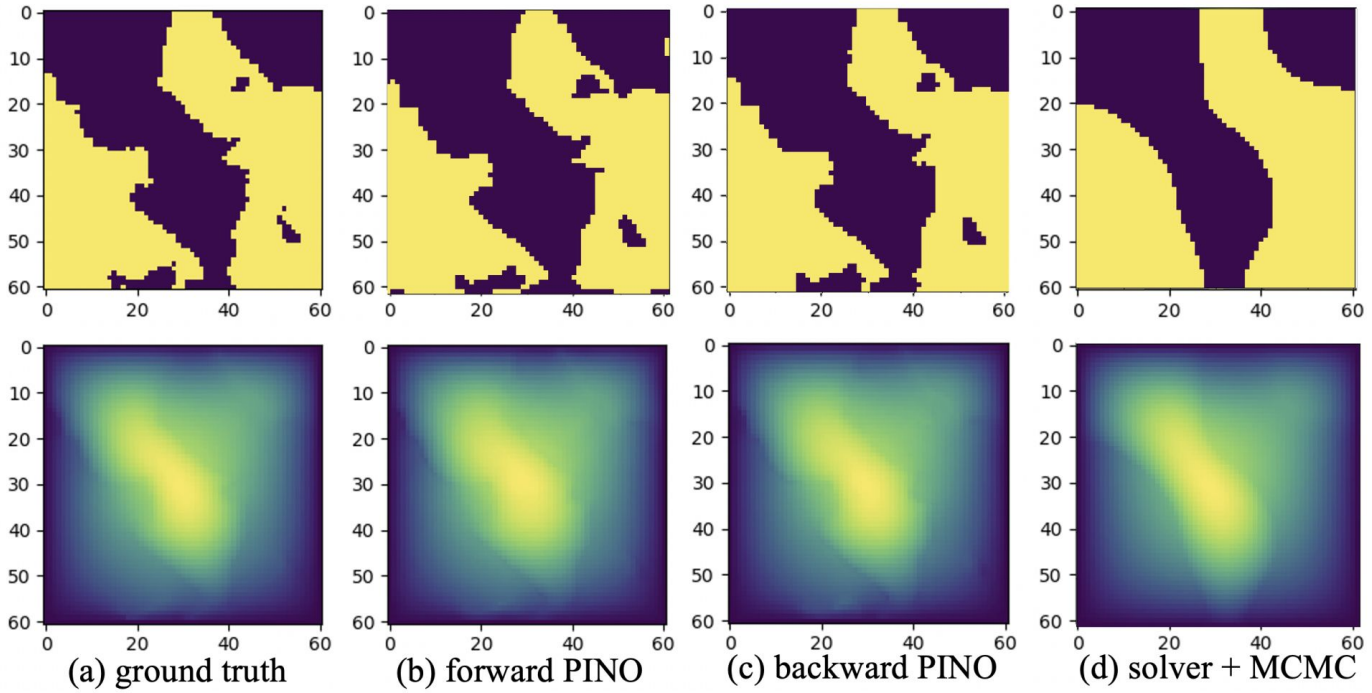
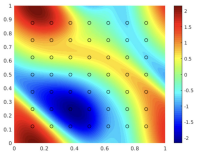



Figure 3: Darcy inverse problem

# Benefits: Speed + Zero-shot Super-resolution

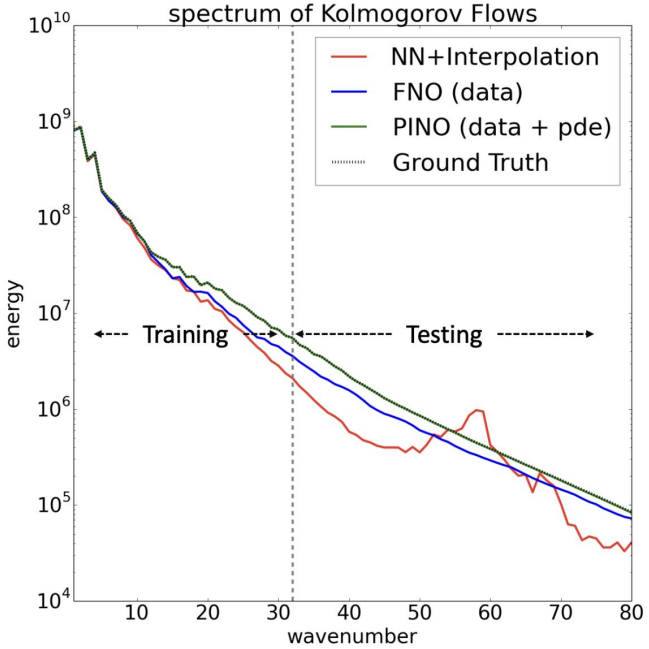
| PDE | Training setting | Error at low data resolution | Error at 2× data resolution | Error at 4× data resolution |
|---|---|---|---|---|
| Burgers | Data | 0.32±0.01% | 3.32±0.02% | 3.76±0.02% |
| | Data & PDE loss | 0.17±0.01% | 0.28±0.01% | 0.38±0.01% |
| Darcy | Data | 5.41±0.12% | 9.01±0.07% | 9.46±0.07% |
| | Data & PDE loss | 5.23±0.12% | 1.56±0.05% | 1.58±0.06% |
| Kolmogorov flow | Data | 8.28%±0.15% | 8.27%±0.15% | 8.30%±0.15% |
| | Data & PDE loss | 6.04%±0.12% | 6.02%±0.12% | 6.01%±0.12% |

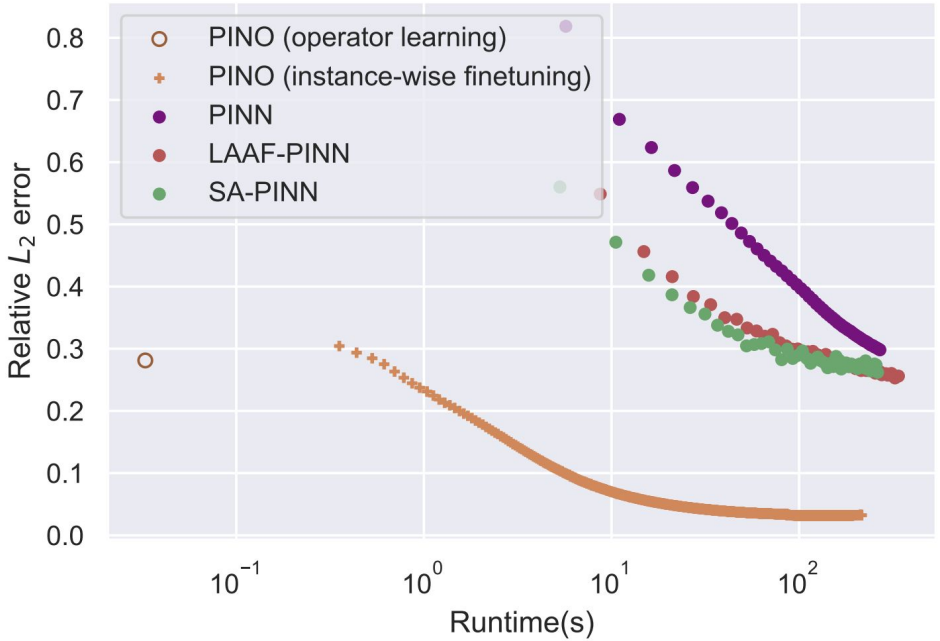


400x faster than traditional GPU-based spectral solver

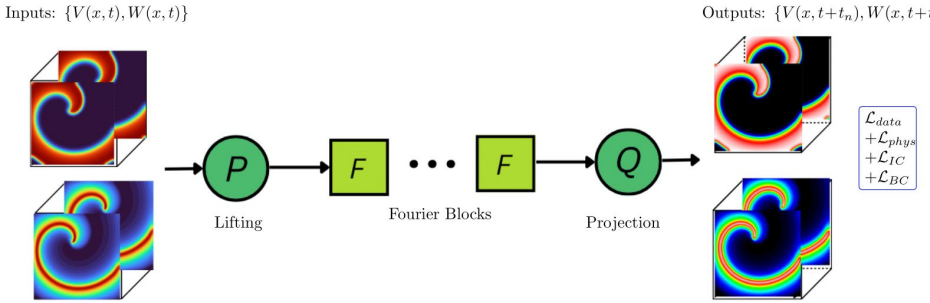
| Method | # data samples | # PDE instances | Solution error $(w)$ | Time cost |
|---|---|---|---|---|
| PINNs | - | - | 18.7% | 4577s |
| PINO | 0 | 0 | **0.9%** | 608s |
| PINO | 0.4k | 0 | **0.9%** | 536s |
| PINO | 0.4k | 160k | **0.9%** | **473**s |

# Benefits: Generalization + Optimization

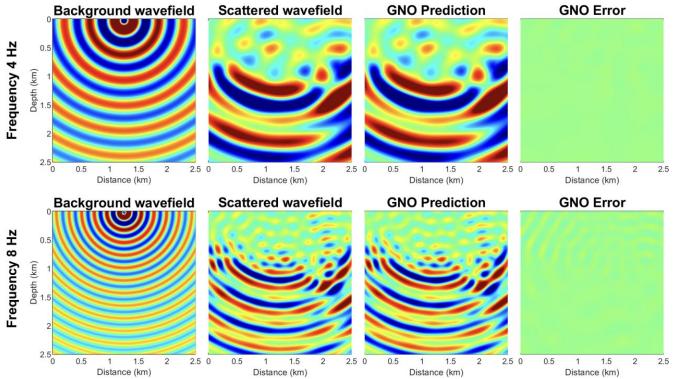


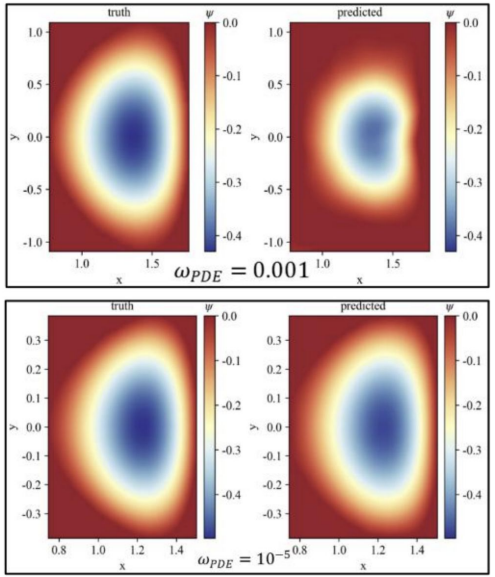
# Applications




Paoletti et. al, "Physics-Informed Neural Operators for Cardiac Electrophysiology", 2025



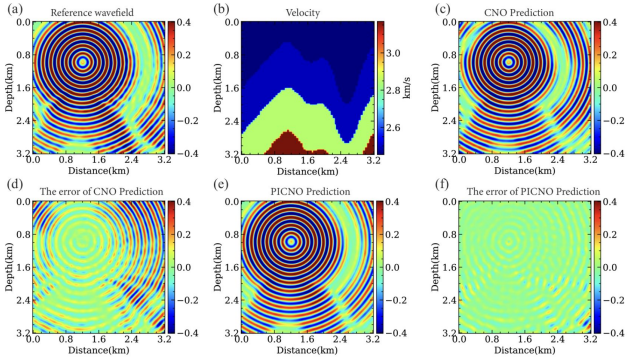

Alkhalifah et. al, "Seismic wavefield solutions via physics-guided generative neural operator", 2025



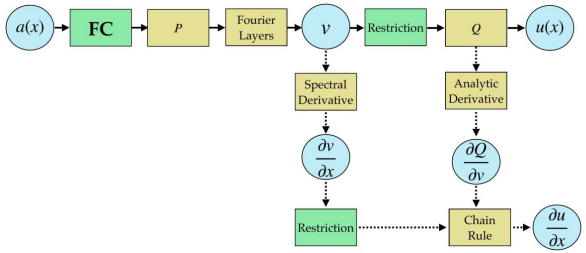

Liu et. al, "Physics-informed Neural Operator Learning for Nonlinear Grad-Shafranov Equation", 2025

# Frontiers

- Improved Architectures and optimization
  - Fourier Continuation PINO (FC-PINO), Layered Fourier Reduced PINO (LFR-PINO), Physics-Informed Convolutional Neural Operator (PICNO)
- Discovery!
- New applications
  - Climate, environmental science, materials, energy, fusion, personal medicine




Maxiao, "An effective physics-informed neural operator framework for predicting wavefields", Machine Learning and Computation 2025




Anandkumar et. al, "FC–PINO: High-Precision Physics-Informed Neural Operators via Fourier Continuation", 2025

# Thank you!