ISART™ 2026
BOULDER, CO | AUGUST 11-13, 2026

Physics-Informed Machine Learning: Methods, Frameworks, and Applications in Computational Electromagnetics

Raj Shukla
Division of Applied Mathematics
Brown University

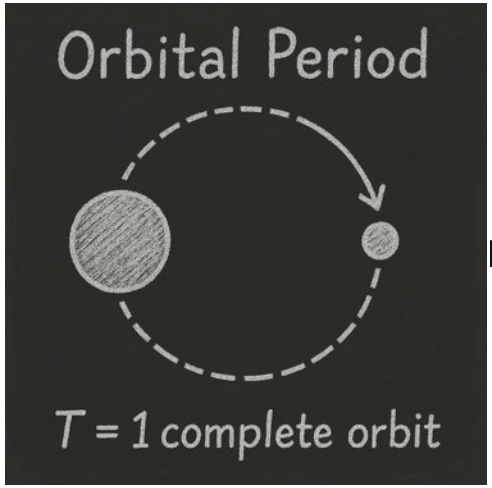
Disclaimer

This presentation does not imply the endorsement or approval by the United States Government or the National Telecommunications and Information Administration (NTIA), and does not represent the views of the U.S. Government or NTIA.

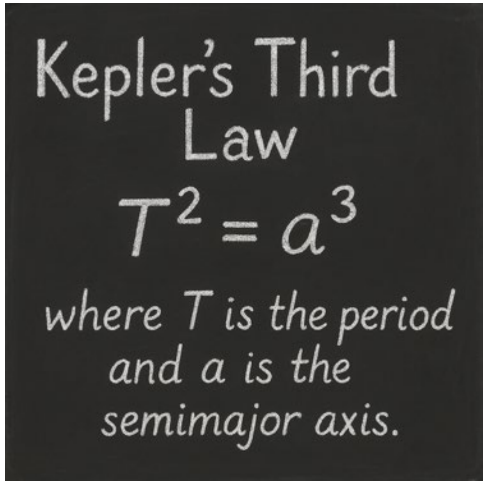
Why Data and Physics?

- Johannes Kepler discovered his famous third law of planetary motion, (period $)^2 \propto ($ radius $)^3$, from searching for patterns in thirty years of data produced by **Tycho Brahe's** unaided eye. Kepler did not discover this by searching through all conceivable relationships with a genetic algorithm on a computer, but by his own geometrical intuition". **And it was Kepler's Third Law, not an apple, that led Isaac Newton to discover the law of gravitation**" (Hawking, 2004).

Data



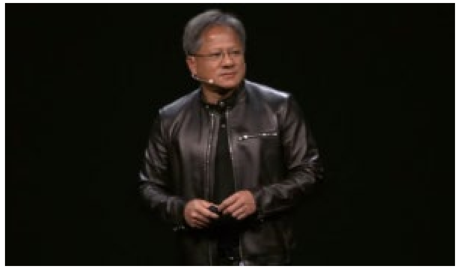
Physics



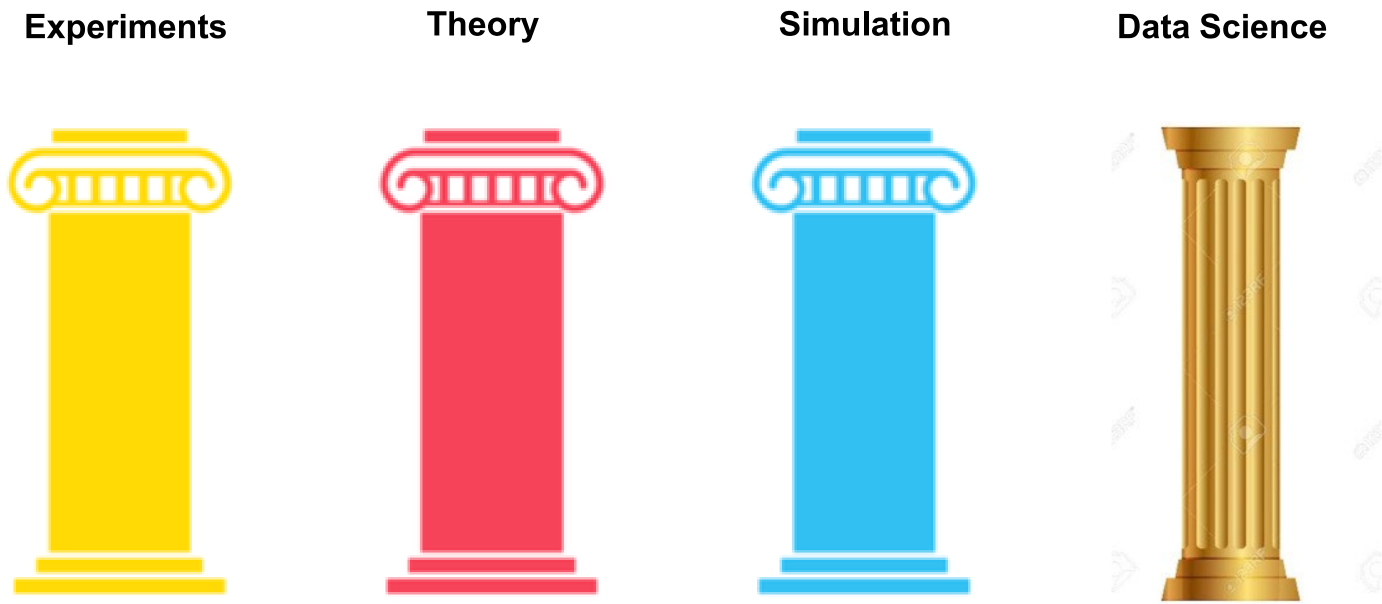
- Likewise, Planck's law was not derived from first principles, **but was a symbolic form fit to data (Planck, 1900).** This symbolic relationship would inspire the development of Quantum Mechanics.

The Four Pillars of Scientific Methods

"Data science (AI+HPC) is the fastest growing field of computer science today. Because of several different factors, it has become the fourth pillar of the scientific method."

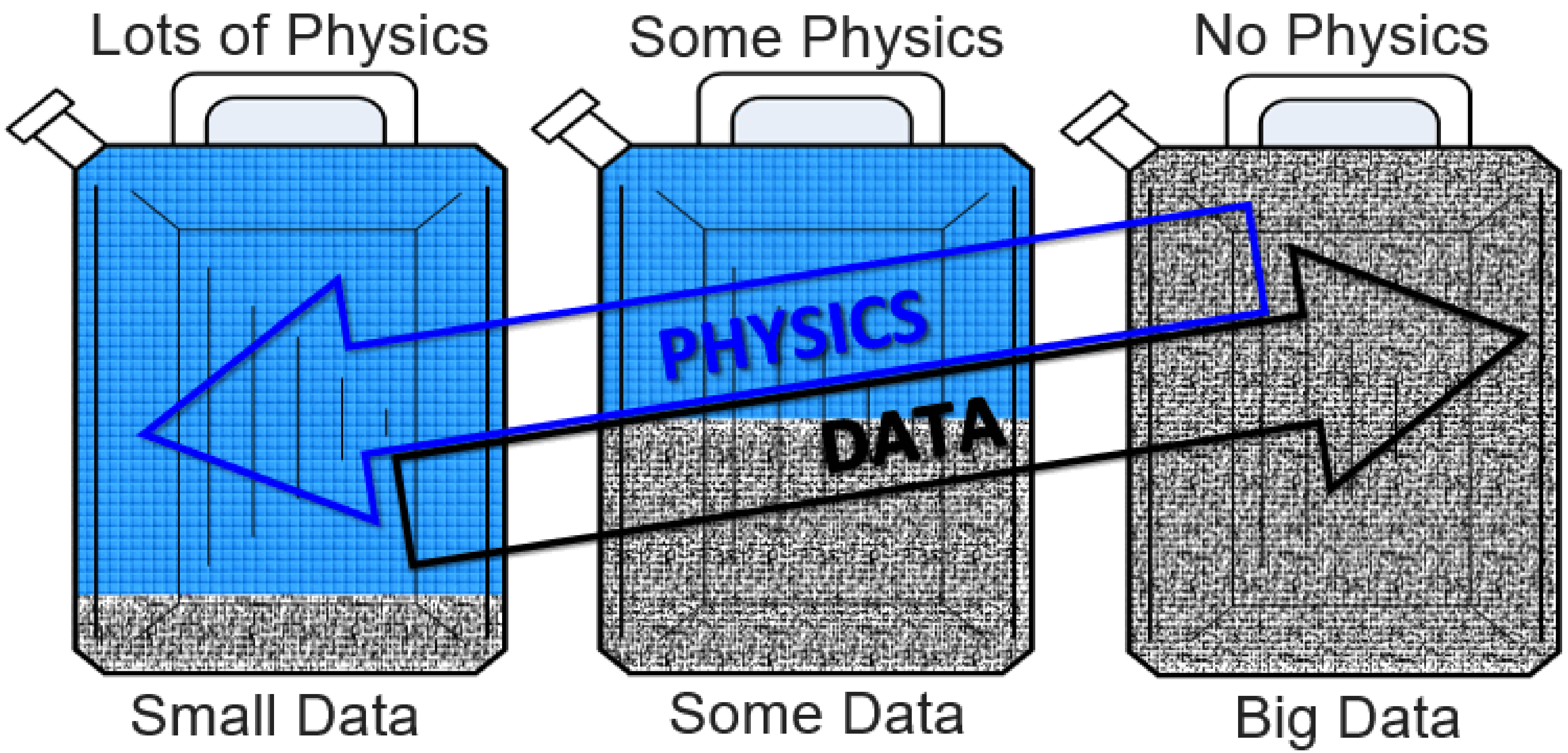


Jensen Huang, NVIDIA



Data + Physical Laws

Three scenarios of
Physics-Informed Learning Machines

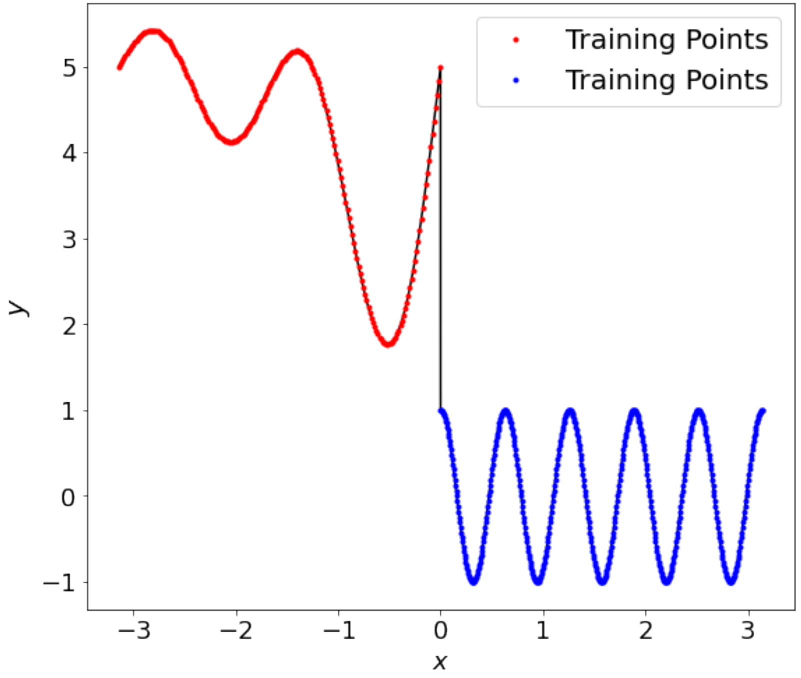


Regression of a Discontinuous/Oscillatory Function

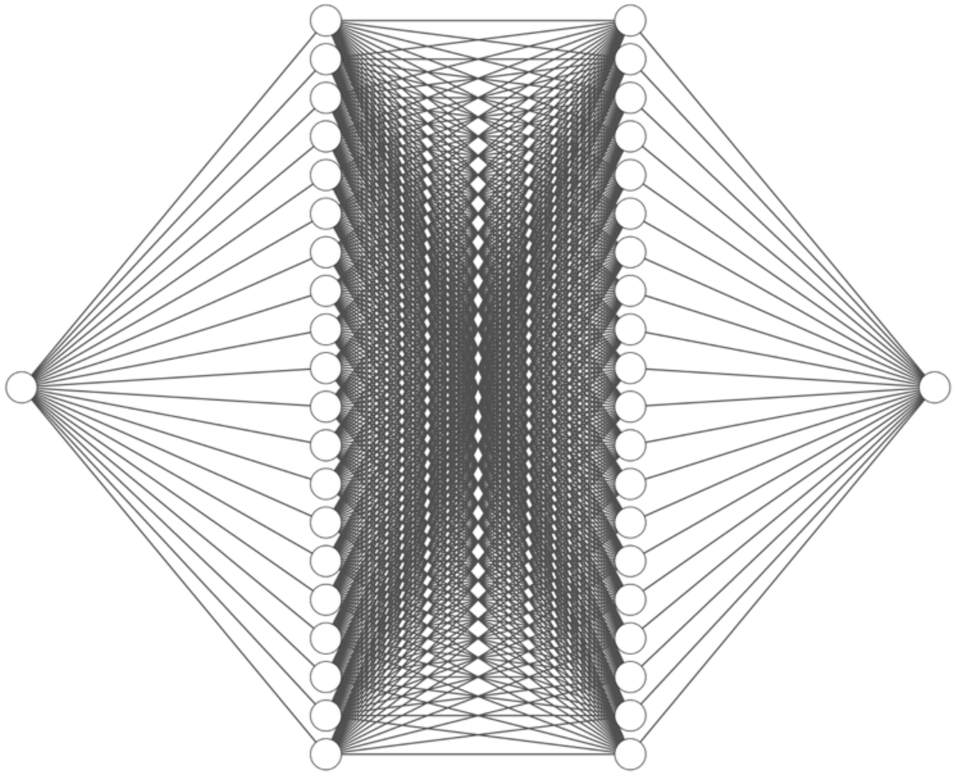
- **Long Tail, Hierarchical training, Spectral Bias, Discontinuity**

$$y = 5 + \sum_{k=1}^{4} \sin(kx), \;\; x < 0$$

$$y = \cos(10x), \;\; x \geq 0$$



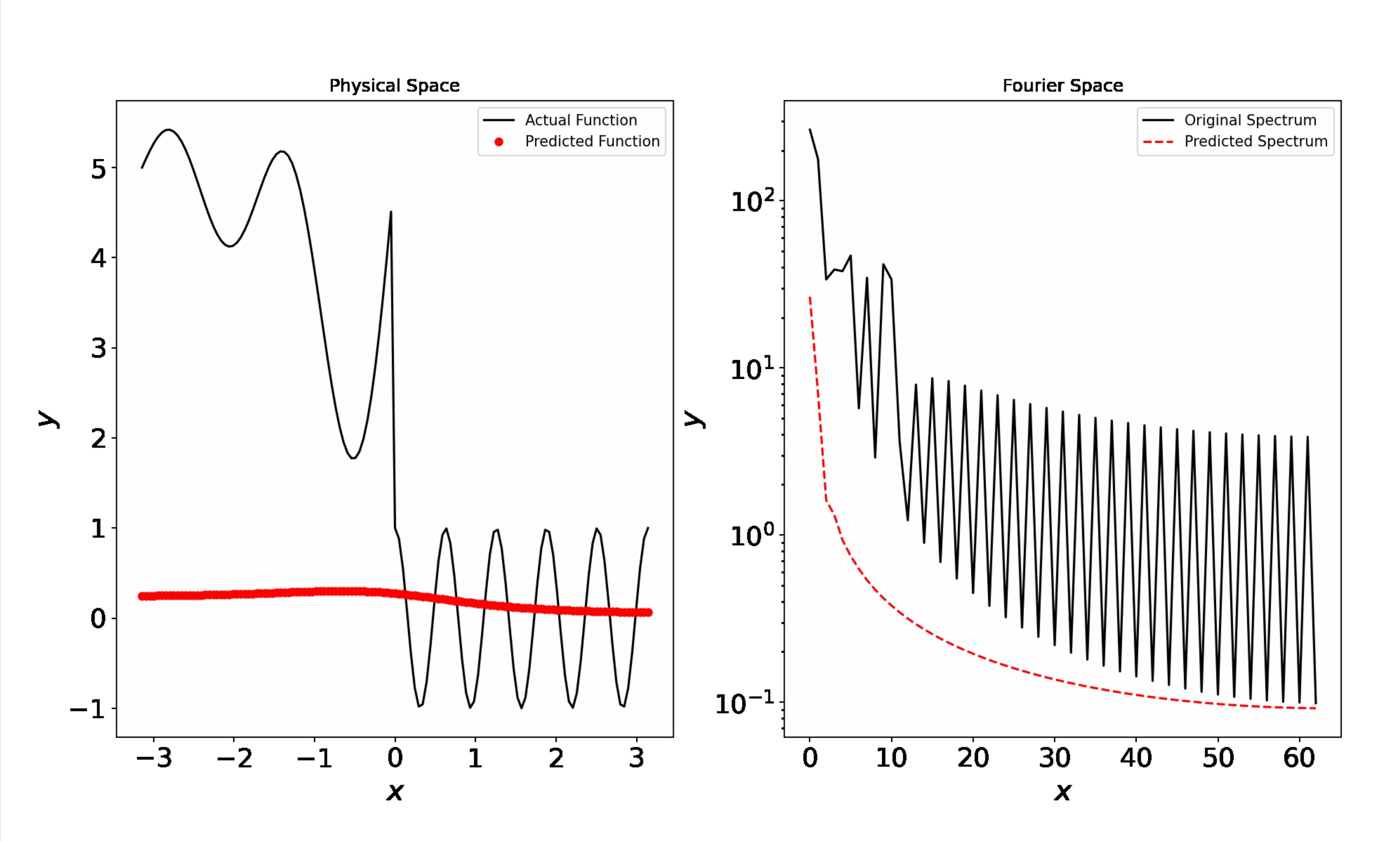
x



$y = \mathcal{N}_\theta(x)$

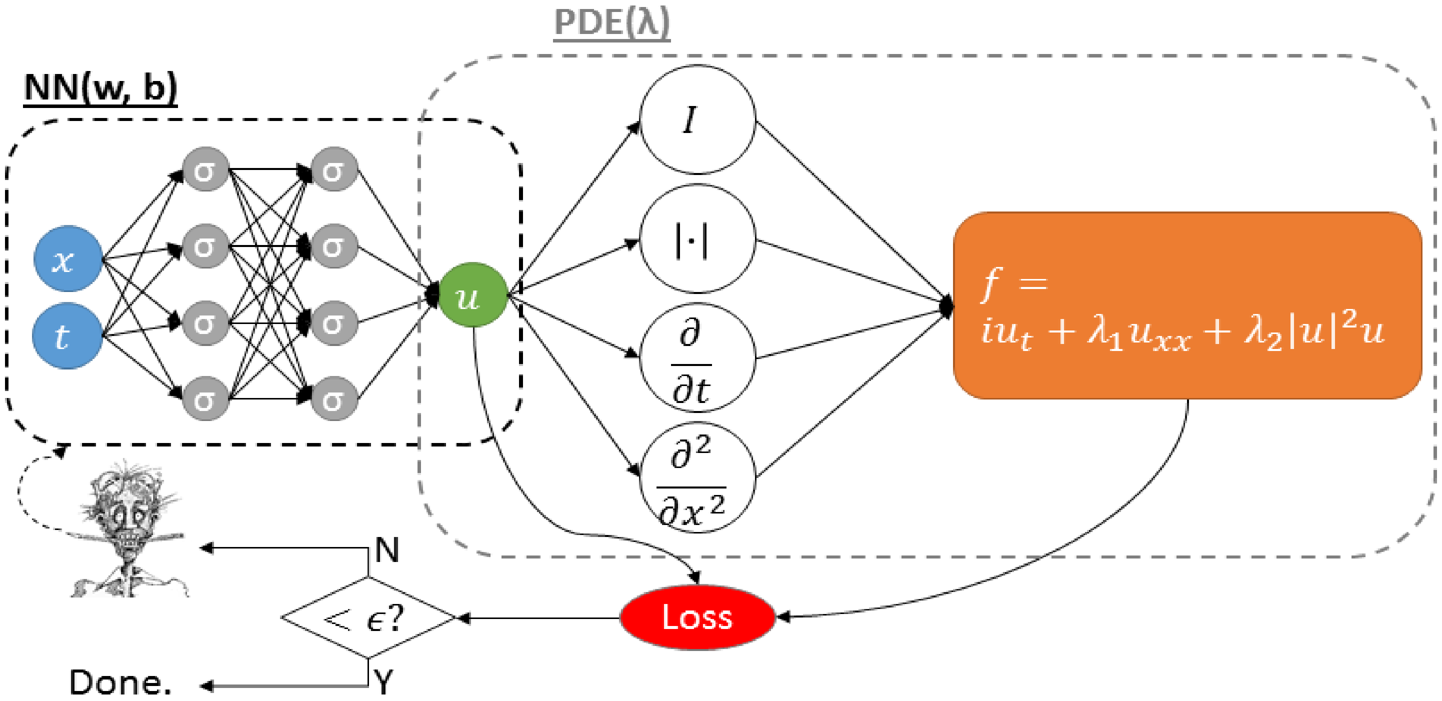
$$N = 20, \;\; L = 2, \;\; \text{and } m = 702$$

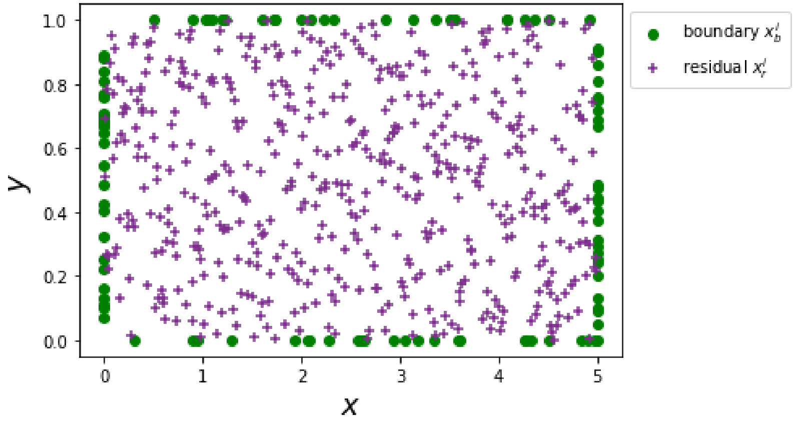
Regression of a Discontinuous/Oscillatory Function



What is a PINN? Physics-Informed Neural Network

We employ two (or more) NNs that share the same parameters





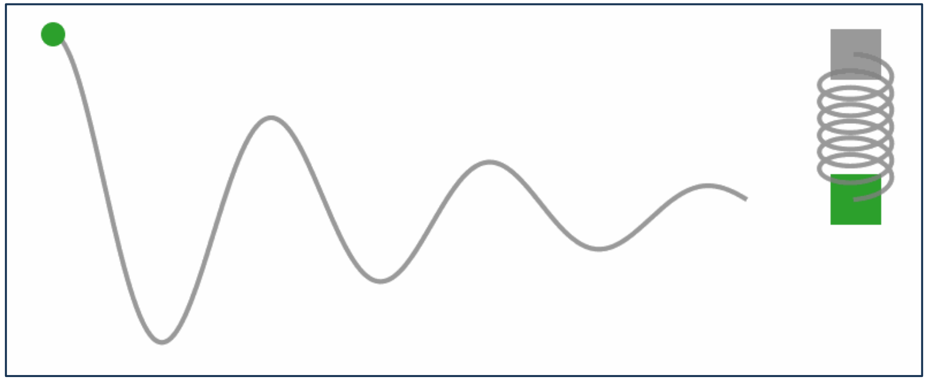
- Minimize:

$$MSE = MSE_u + MSE_f,$$

$$MSE_u = \frac{1}{N_u} \sum_{i=1}^{N_u} |u(t_u^i, x_u^i) - u^i|^2,$$

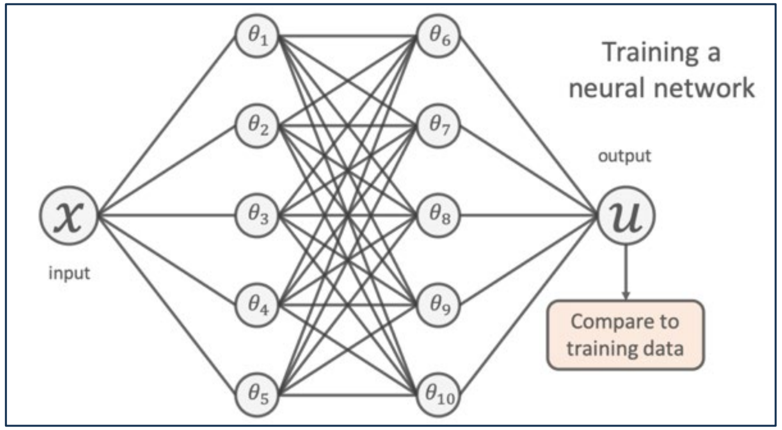
$$MSE_f = \frac{1}{N_f} \sum_{i=1}^{N_f} |f(t_f^i, x_f^i)|^2.$$

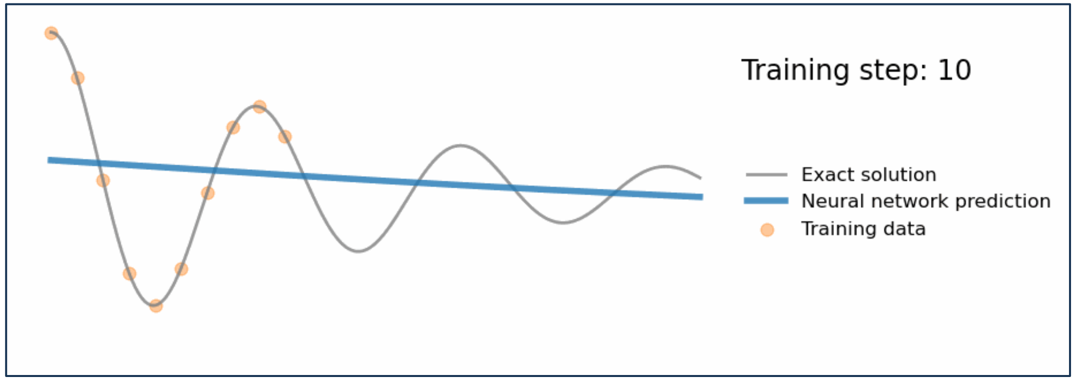
PINN 101



Data

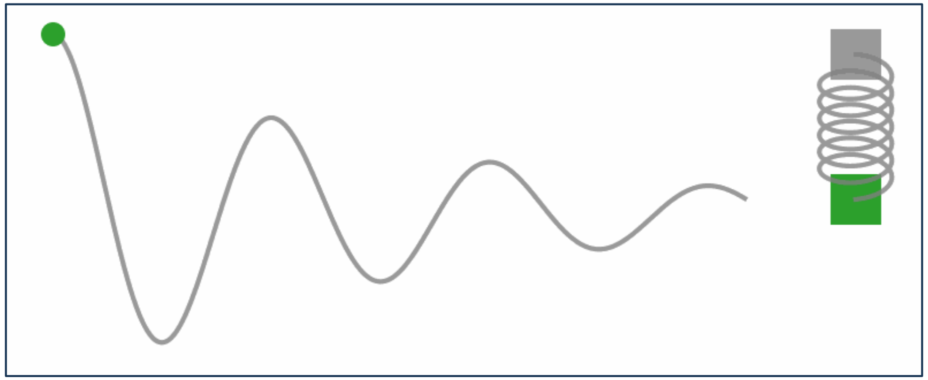
$$m\frac{d^2u}{dx^2} + \mu\frac{du}{dx} + ku = 0$$





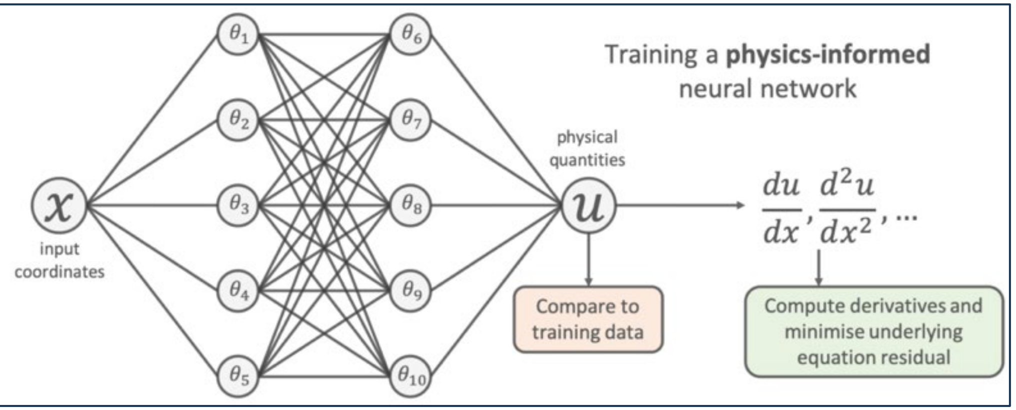
No-Physics

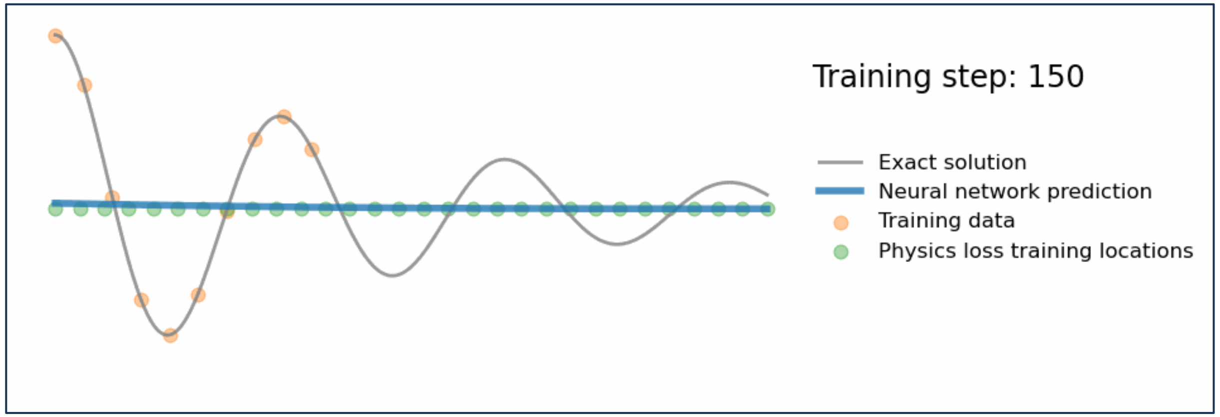
PINN 101



Data

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

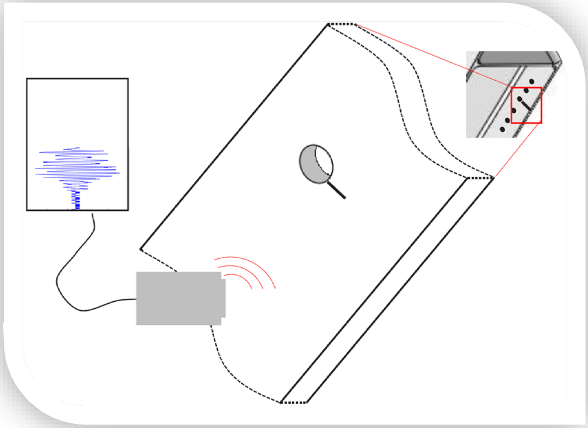


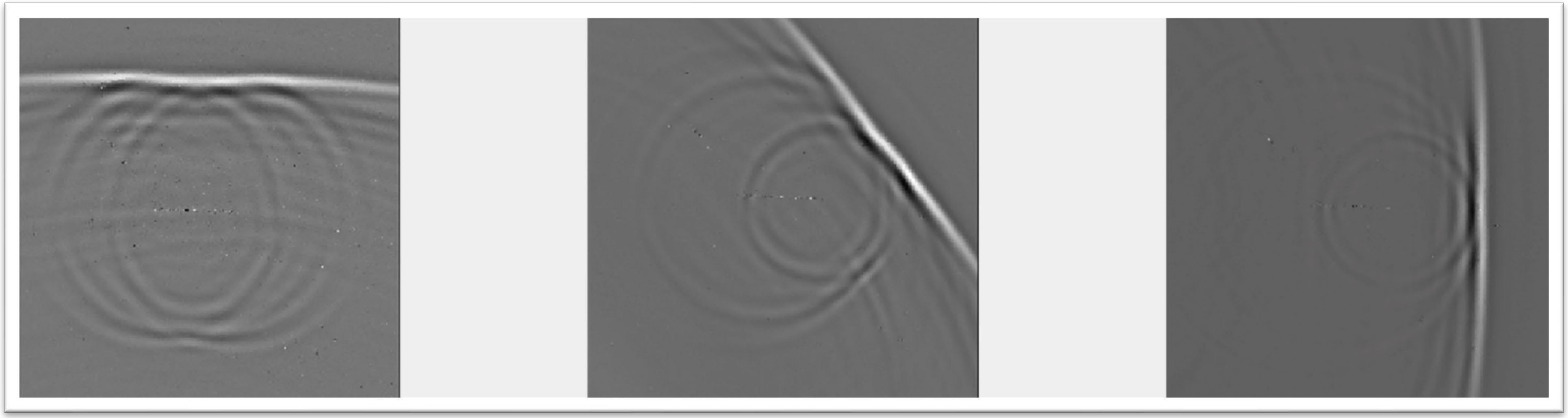


With Physics

Crack characterization in an Aluminum Alloy using PINNs (WPAFB Real Data)

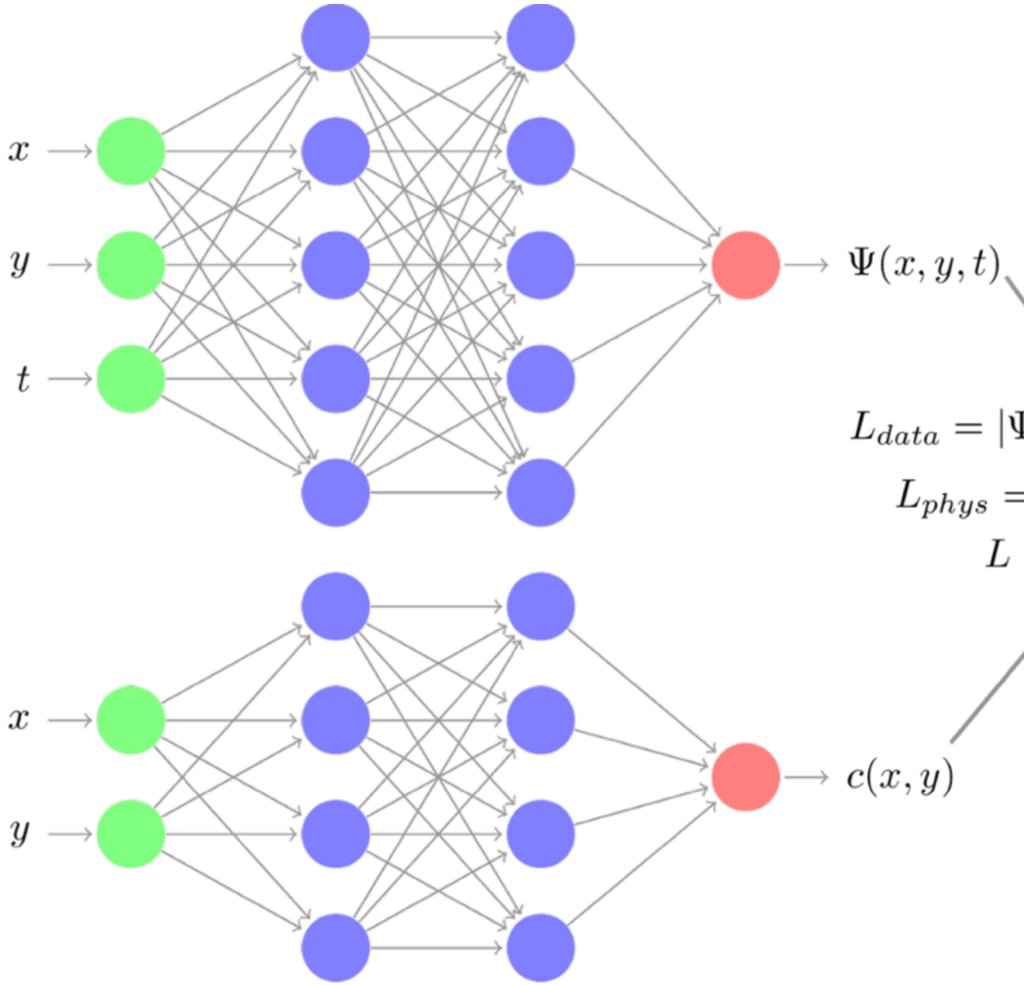
- **Objective:** Identifying and characterizing a surface breaking crack in a 7075-T651 Aluminum alloy metal plate.
- **Dataset:** Particle displacement induced by surface acoustic waves of 5 MHz at three incidence angle 0°, 45°, and 90°.
- **Approach:** Reconstruct the wave field and compute the speed of sound using Inverse PINNs.





K. Shukla, P. Clark Di Leoni, J. Blackshire, D. Sparkman, G. E. Karniadakis,“ Physics-informed Neural Network for Ultrasound Nondestructive Quantification of Surface Breaking Cracks. J Nondestruct Eval 39, 61 (2020). https://doi.org/10.1007/s10921-020-00705-1

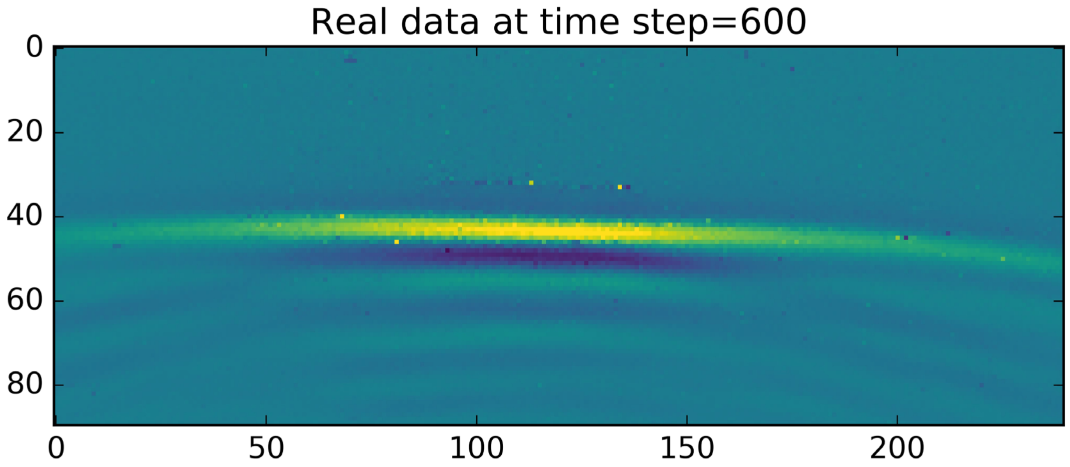
Problem Setup and Method

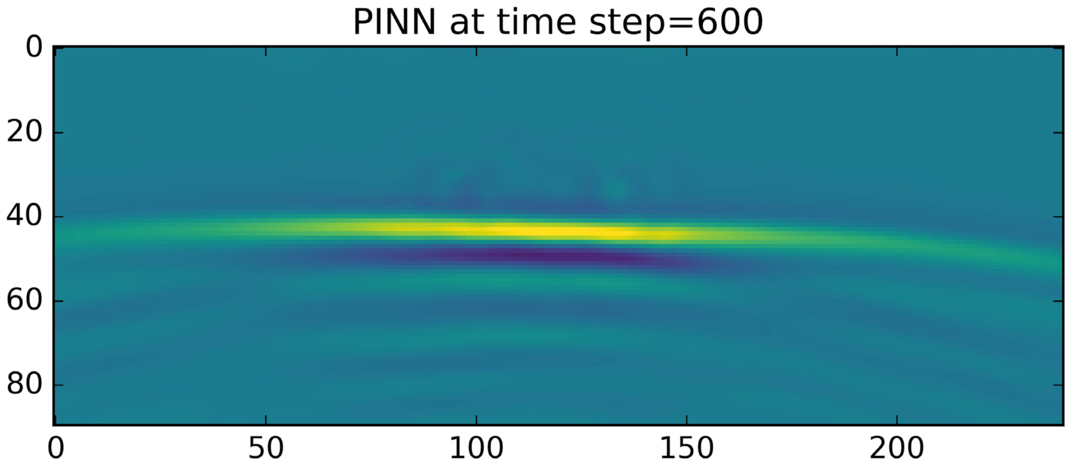


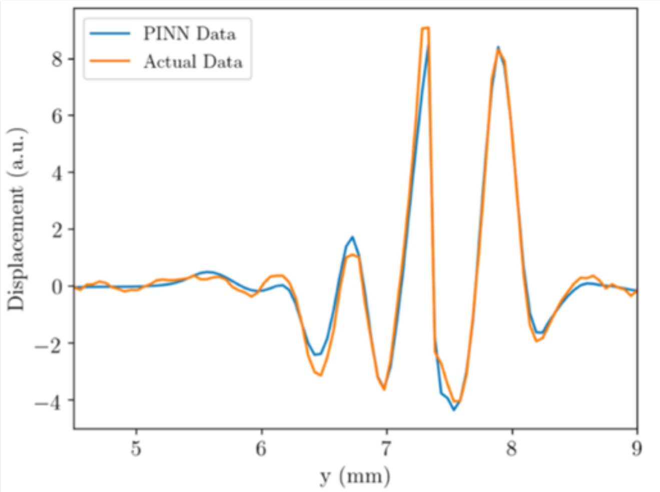
$$L_{data} = |\Psi(x, y, t) - \Psi_{data}(x, y, t)|^2$$
$$L_{phys} = \left| \frac{\partial^2 \Psi}{\partial t^2} - c(x, y)^2 \nabla^2 \Psi \right|^2$$
$$L = \lambda L_{data} + L_{phys}$$

K. Shukla, P. Clark Di Leoni, J. Blackshire, D. Sparkman, G. E. Karniadakis,“ Physics-informed Neural Network for Ultrasound Nondestructive Quantification of Surface Breaking Cracks. J Nondestruct Eval 39, 61 (2020). https://doi.org/10.1007/s10921-020-00705-1

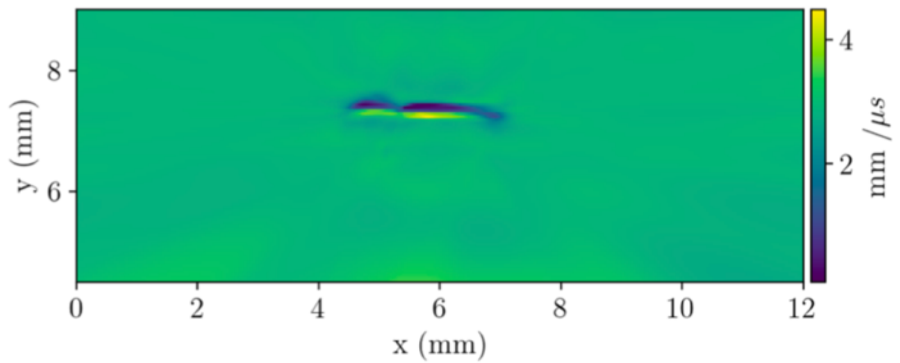
Ultra-Sound Testing of Materials (WPAFB Real Data)







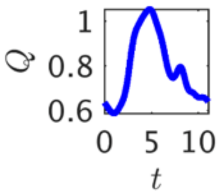
(c) Traces of (a) and (b) at $x = 6$ mm at $t = 12.38$ μs.

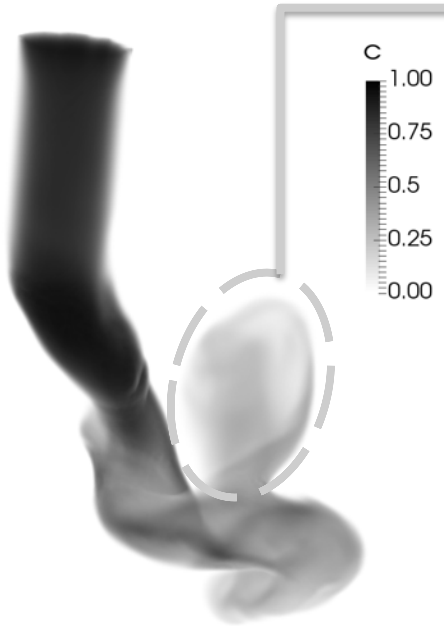


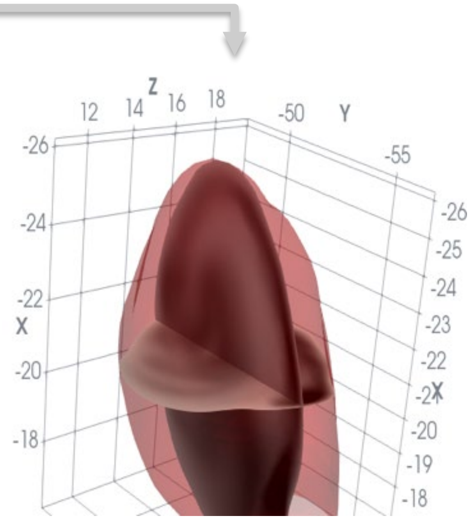
(d) Speed $v(x, y)$ recovered from PINN simulation.

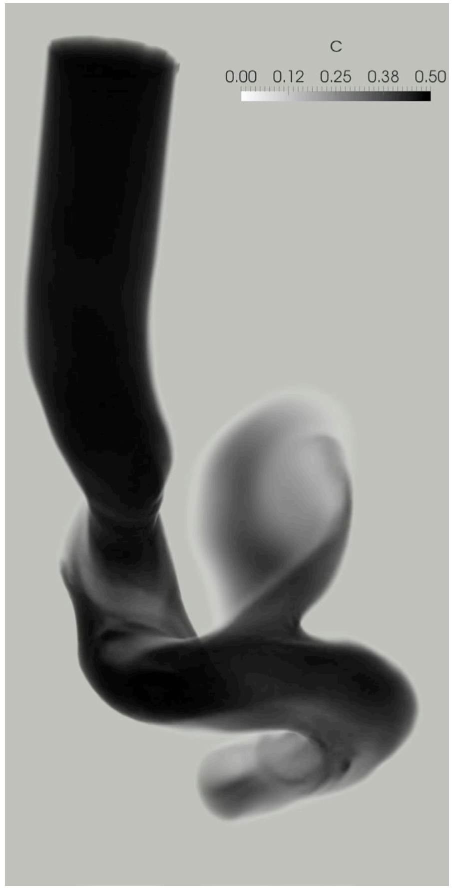
Hidden Fluid Mechanics: Flow dynamics in an intracranial aneurysm

Science. 2020 Feb 28;367(6481):1026-30.



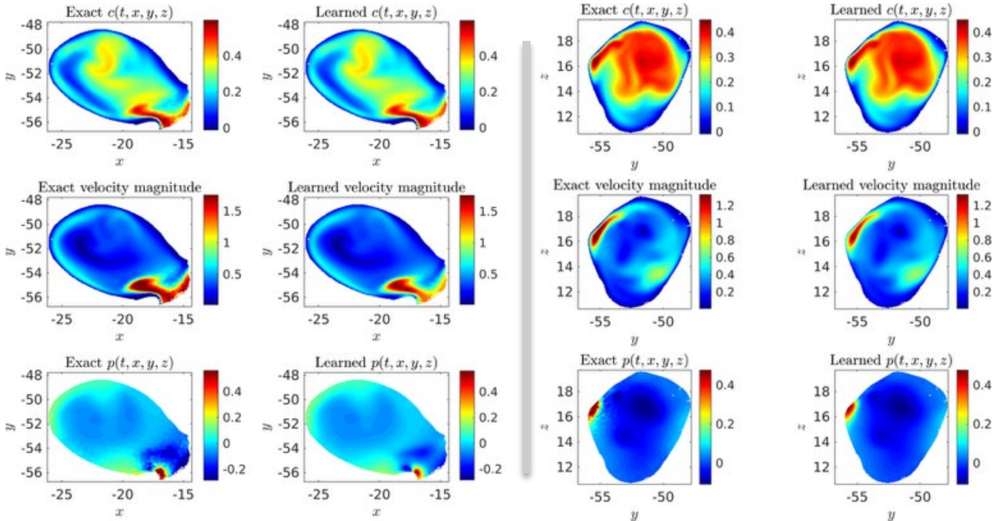


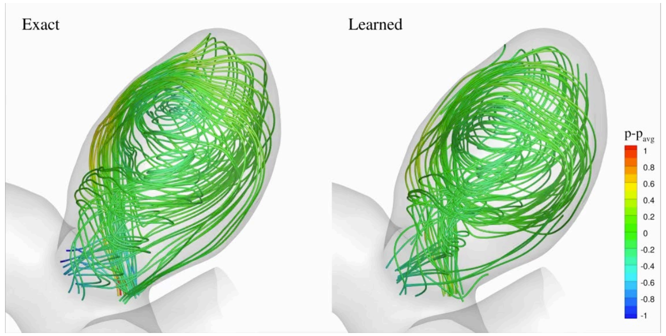




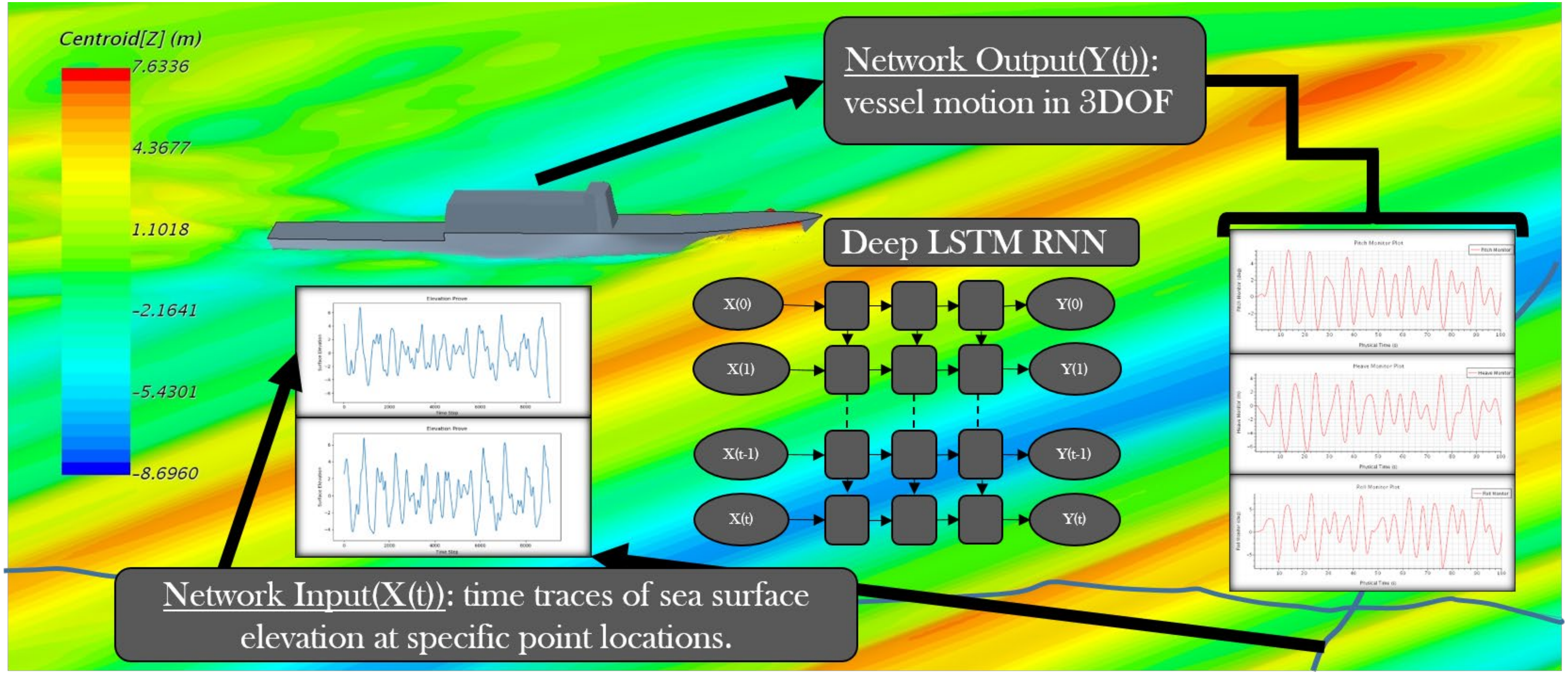
Training Data

Inferred Hidden Physics



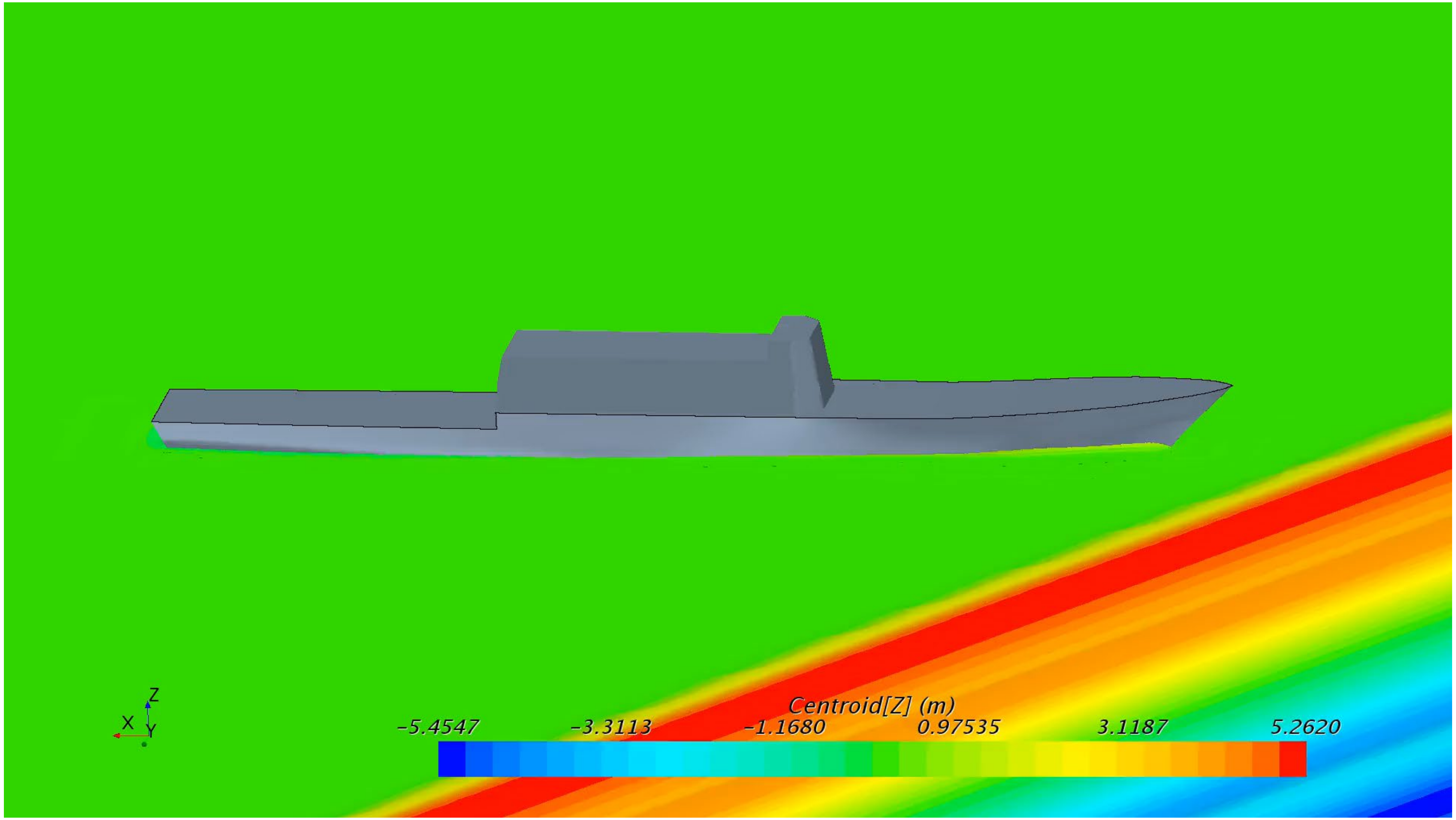


Autonomy: Predicting the motion of battleships in extreme sea states



Jose del Aguila Ferrandis, Chryssostomos Chryssostomidi, Michael Triantafyllou, Karniadakis, G.E. Learning functionals via LSTM

Can Neural Networks Approximate Functionals?



Functions map data, Operators map functions

- Function: $\mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$

  e.g., image classification:  $\mapsto 5$

- Operator: function (∞-dim) $\mapsto$ function (∞-dim)
  e.g., derivative (local): $x(t) \mapsto x'(t)$
  e.g., integral (global): $x(t) \mapsto \int K(s,t)x(s)ds$

  e.g., dynamic system: $x(t) \longrightarrow \boxed{\text{System } T} \longrightarrow y(t)$

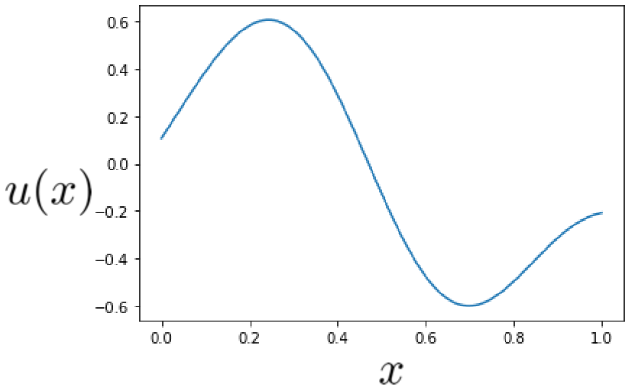
  e.g., biological system
  e.g., social system

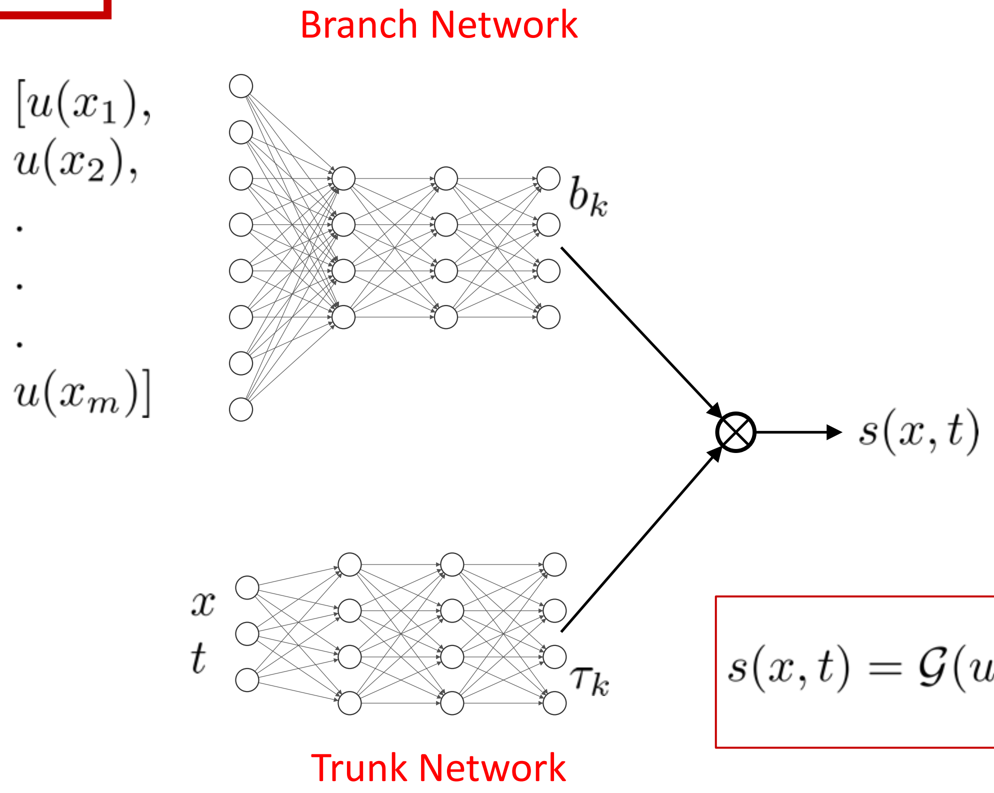
$\Rightarrow$ Can we learn operators via neural networks?
$\Rightarrow$ How?

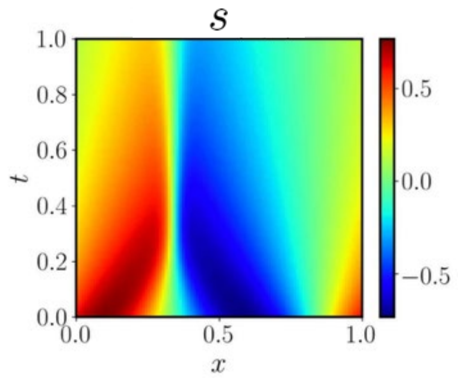
Deep Operator Network (DeepONet)

1D Burger's Equation

$$\frac{\partial s}{\partial t} + s\frac{\partial s}{\partial x} - \nu\frac{\partial^2 s}{\partial x^2} = 0, \quad (x,t) \in (0,1) \times (0,1]$$
$$s(x,0) = u(x), \quad x \in (0,1)$$







$$s(x,t) = \mathcal{G}(u)(x,t) \approx \sum_{k=1}^{p} b_k(v)\tau_k(\xi) + b_0$$

DeepONet learns the underlying operator from the data

Complex Valued DeepONet for 3D Maxwell's Equation

Goal: learn the Maxwell solution operator

$$\mathcal{G} : (\boldsymbol{E}^i, f) \mapsto \boldsymbol{E}(\boldsymbol{x}), \qquad \boldsymbol{x} = (x, y, z) \in \Omega.$$

Time-harmonic Maxwell equations

$$\begin{aligned} -i\omega\epsilon\boldsymbol{E} - \nabla \times \boldsymbol{H} &= 0, \\ -i\omega\mu\boldsymbol{H} + \nabla \times \boldsymbol{E} &= 0, \qquad \boldsymbol{x} \in \Omega, \end{aligned}$$

with

$$\omega = 2\pi f, \qquad \boldsymbol{E} = \boldsymbol{E}^i + \boldsymbol{E}^s.$$

Inputs and output

- Branch input 1: sampled incident field

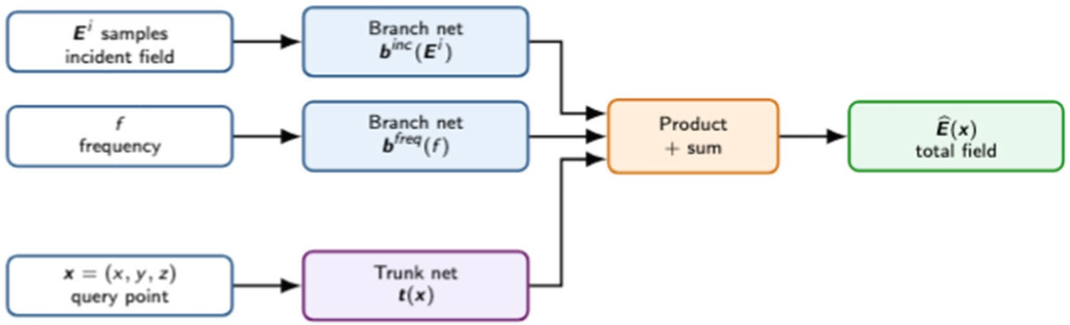
$$\boldsymbol{E}^i = \{\boldsymbol{E}^i(\boldsymbol{x}_j)\}_{j=1}^m.$$

- Branch input 2: frequency f.
- Trunk input: spatial query point $\boldsymbol{x} = (x, y, z)$.
- Output: total electric field $\boldsymbol{E}(\boldsymbol{x})$.

Complex-valued fields

Real and imaginary parts of $\boldsymbol{E}^i$ and $\boldsymbol{E}$ are learned as separate real-valued components.

Multi-input DeepONet setup



DeepONet approximation

$$\boxed{\widehat{\boldsymbol{E}}(\boldsymbol{x}) = \mathcal{G}_\theta(\boldsymbol{E}^i, f)(\boldsymbol{x}) \approx \sum_{\ell=1}^{r} b_\ell^{inc}(\boldsymbol{E}^i) b_\ell^{freq}(f) t_\ell(\boldsymbol{x}) + \boldsymbol{b}_0}$$

- Branch networks encode global problem information.
- Trunk network encodes spatial coordinates.
- The trained operator evaluates $\boldsymbol{E}(\boldsymbol{x})$ at arbitrary query locations.

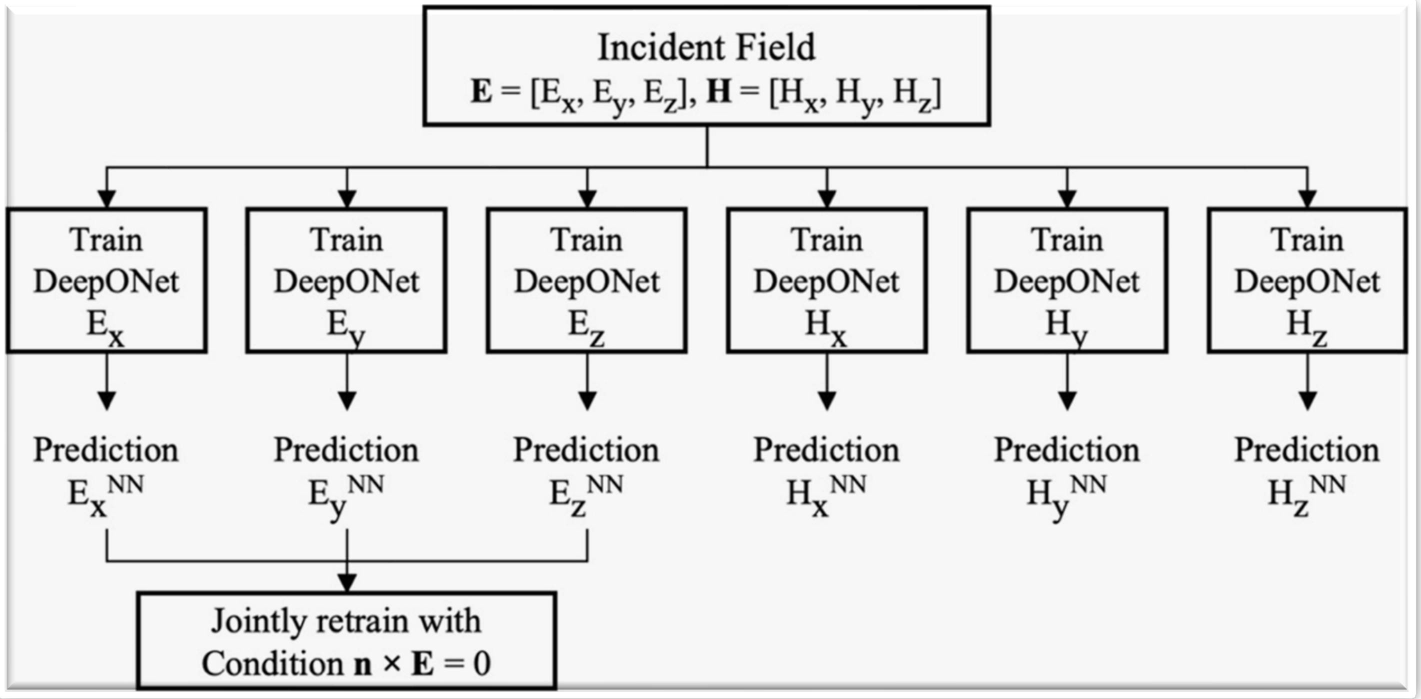
PEC Boundary Condition

$$\boxed{\hat{\boldsymbol{n}} \times \boldsymbol{E}^{NN} = \boldsymbol{0} \qquad \text{on } \partial\Omega_{\mathrm{PEC}}}$$

Idea: the tangential electric field must vanish on a perfect conductor.

- First train DeepONets for each field component.
- Then jointly retrain E_x, E_y, E_z with a PEC penalty.
- Balance data fitting and boundary-condition enforcement.

$$\mathcal{L} = w_1 \left\| \hat{\boldsymbol{n}} \times \boldsymbol{E}^{NN} \right\|_2^2 + w_2 \mathcal{L}_{data}.$$



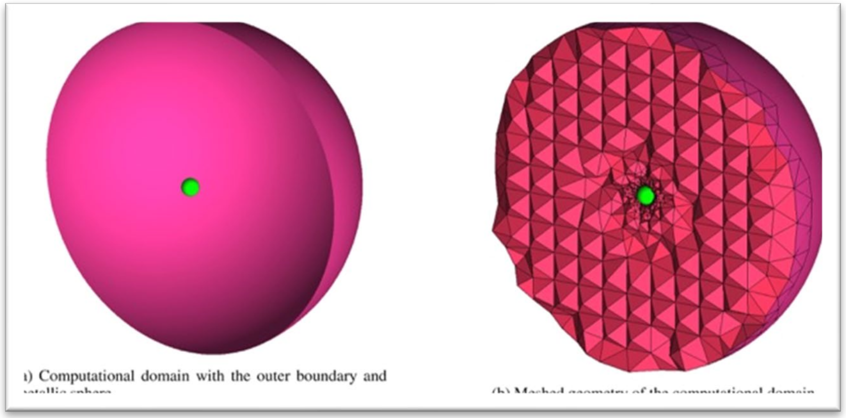
3D Homogeneous Metallic Sphere

Objective: evaluate complex-valued DeepONet on a smooth PEC sphere and compare against real-valued DeepONet.

$$\mathcal{G} : \left(\boldsymbol{E}^i, f\right) \mapsto \boldsymbol{E}(\boldsymbol{x}; f), \qquad \boldsymbol{x} \in \Omega$$

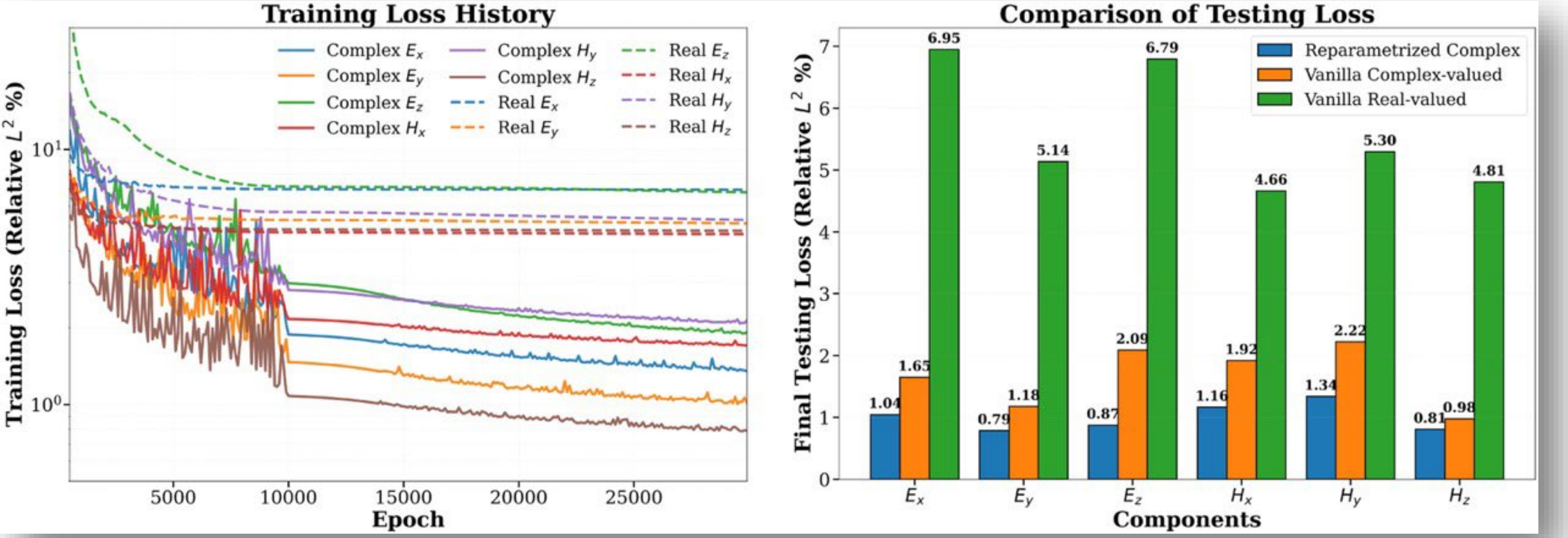
- Metallic sphere of radius $r = 1$ m; outer boundary placed at 3λ.
- Training/testing frequencies sampled from $[0.05, 0.6]$ GHz.
- Data generated using UWVF Maxwell solver.
- Complex-valued DeepONet preserves real–imaginary coupling.
- Testing error reduced by about 5–7$\times$ compared with real-valued DeepONet.
- Inference time: 0.10 s on GPU versus $\sim$ 15 min using UWVF on 192 cores.

Complex-valued DeepONet gives faster and more accurate surrogate predictions.

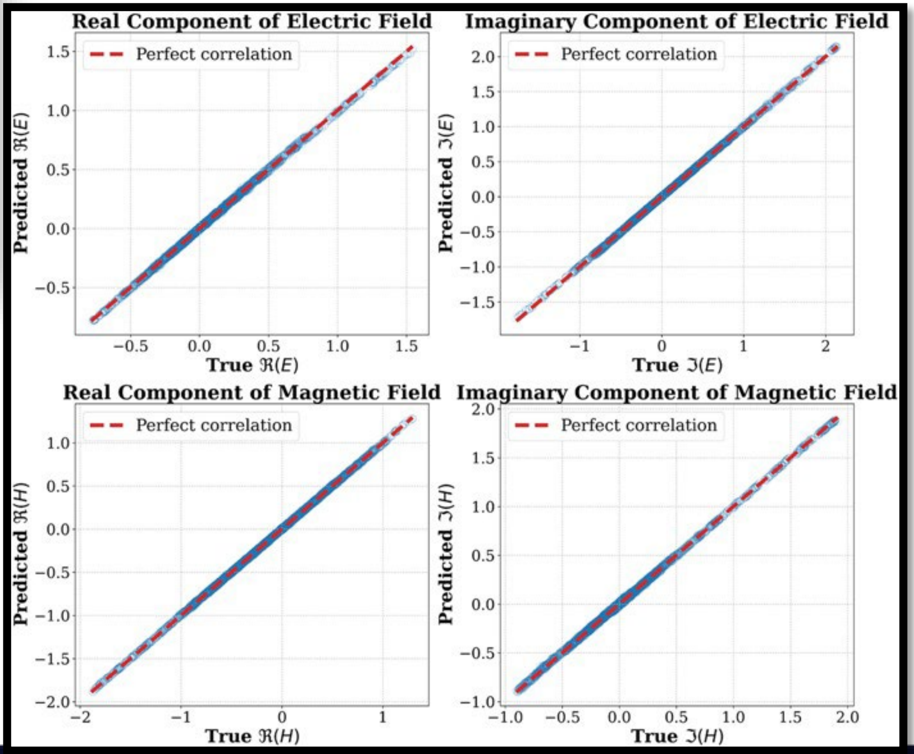


Metrics and Correlations

Training and Testing



Correlation

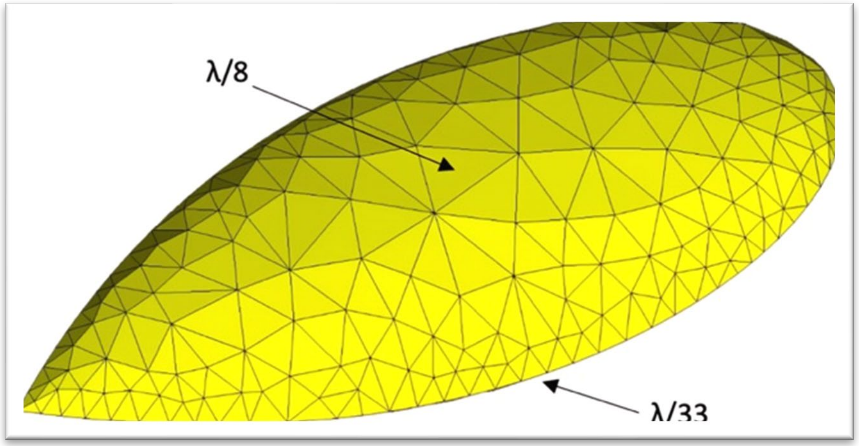


3D Almond Geometry with Sharp Corners

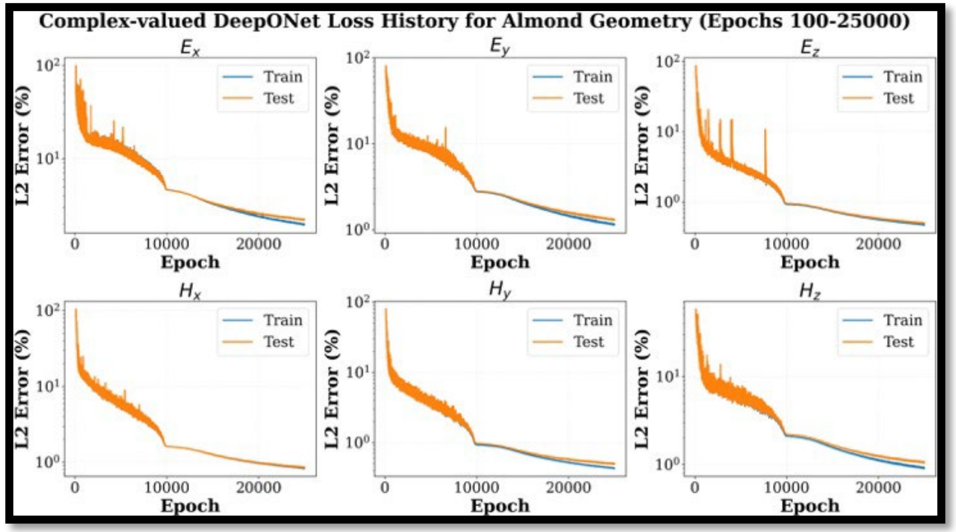
Objective: test complex-valued DeepONet on a singularity-prone metallic target.

- Almond-shaped PEC target with sharp edges and corners.
- Geometry length: 0.252 m; frequency: 1.19 GHz.
- Mesh resolution: $\lambda/8$ on surface and $\lambda/33$ near sharp edges.
- Incident field evaluated at 11,154 sensor locations.
- Dataset: 61 incident angles, split equally for training and testing.
- Six complex-valued DeepONets predict

$$E_x, E_y, E_z, H_x, H_y, H_z.$$



Testing errors remain low: 0.22%–1.24%, with larger errors near sharp edges.



Radar Cross Section (RCS)

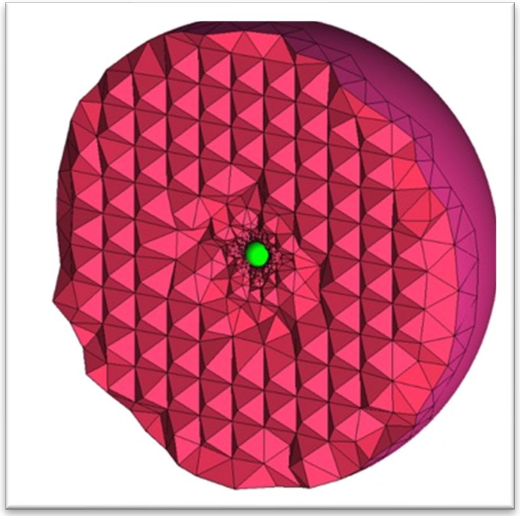
Goal: use DeepONet-predicted fields to compute an integrated electromagnetic quantity.

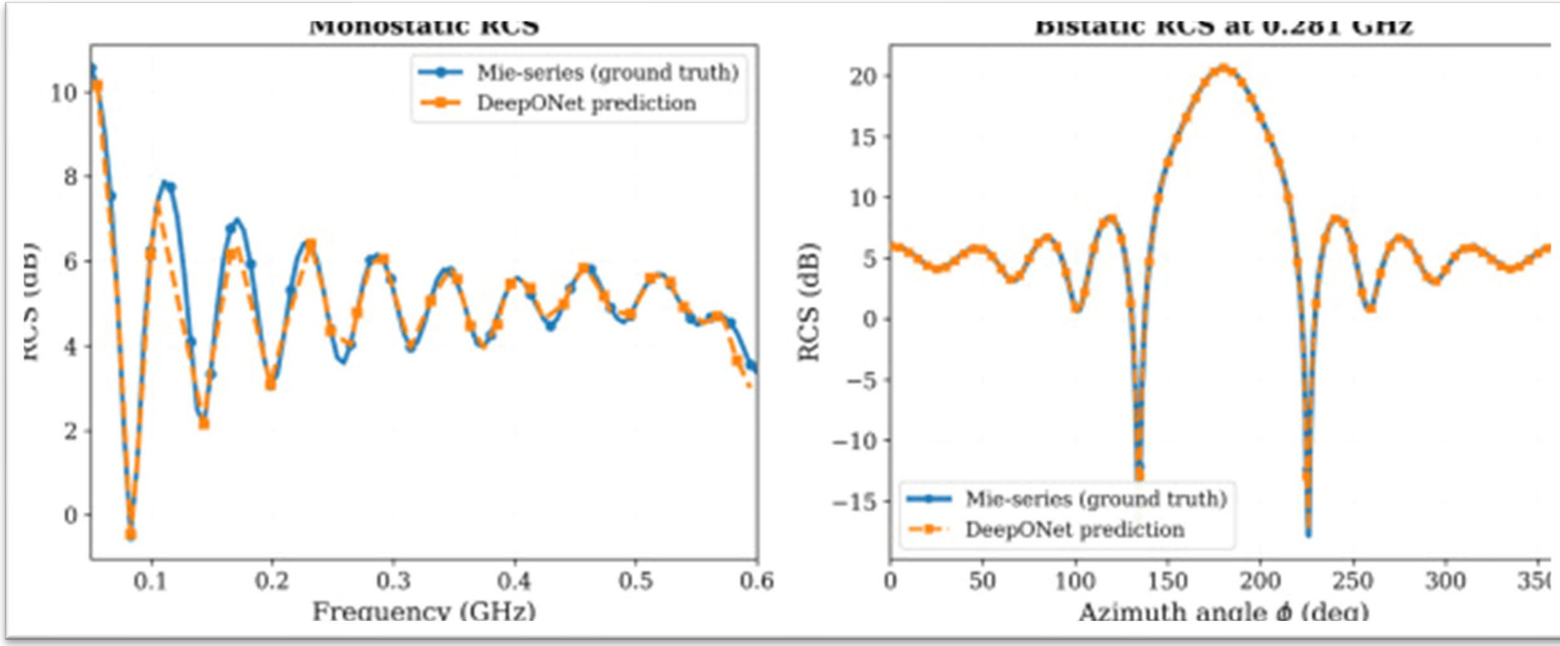
$$\boldsymbol{E}(r,\theta,\phi) \approx \boldsymbol{E}_{\infty}(\theta,\phi)\frac{e^{i\omega\sqrt{\mu\epsilon}\,r}}{r}, \qquad r \to \infty$$

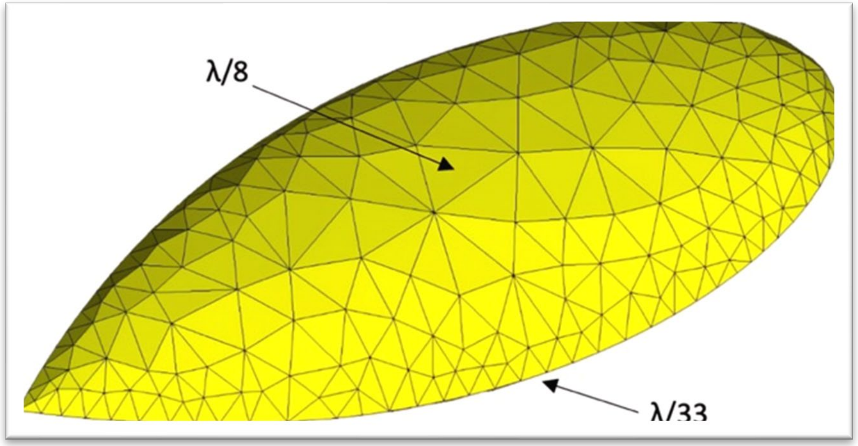
$$\boxed{\sigma(\theta,\phi) = \lim_{r\to\infty} 4\pi r^2 \frac{|\boldsymbol{E}^s(r,\theta,\phi)|^2}{|\boldsymbol{E}^i|^2}}$$

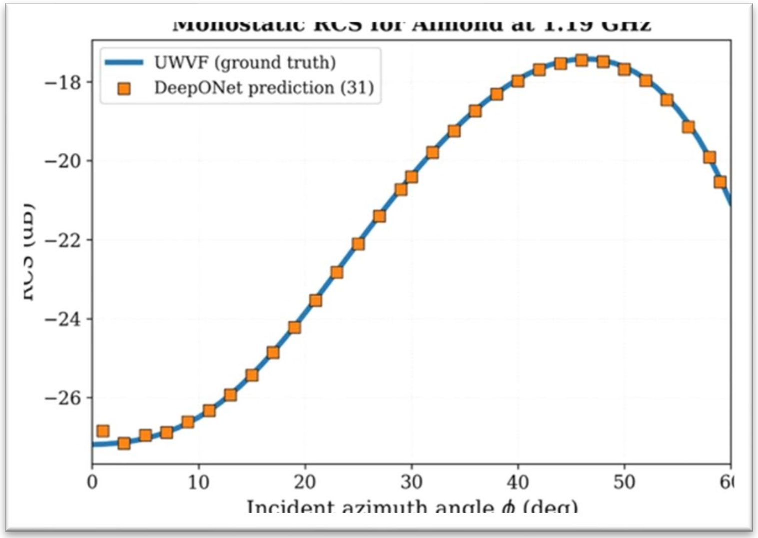
- RCS measures how strongly a target scatters incident electromagnetic energy.
- It is computed from the far-field scattered wave.
- Monostatic RCS: transmitter and receiver are in the same direction.
- Bistatic RCS: transmitter and receiver are in different directions.
- Good RCS agreement confirms that field prediction errors do not destroy the derived physical quantity.

RCS vs Geometries



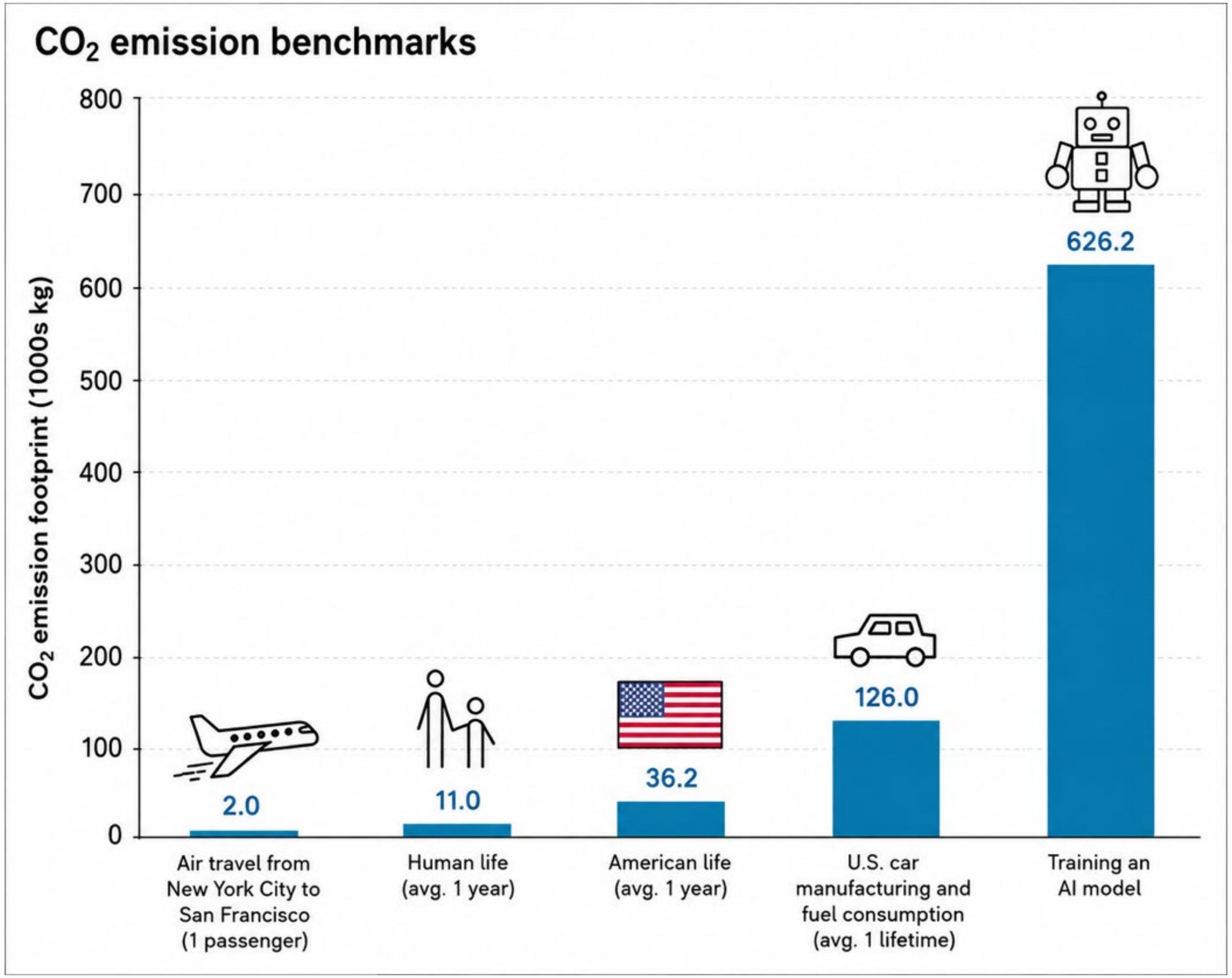






AI Training Emissions in Perspective

- Training large AI models can have a substantial carbon footprint.
- The reported emissions exceed several familiar transportation and lifestyle benchmarks.
- Emissions depend on model size, hardware, training time, and the electricity source.
- Repeated training and hyperparameter searches can further increase energy use.
- Efficient algorithms, optimized hardware, and renewable energy are essential for sustainable AI.

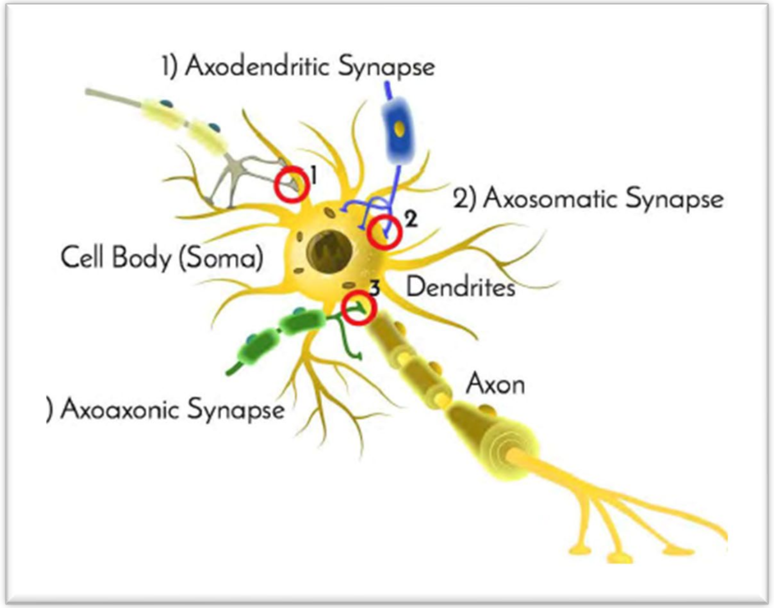


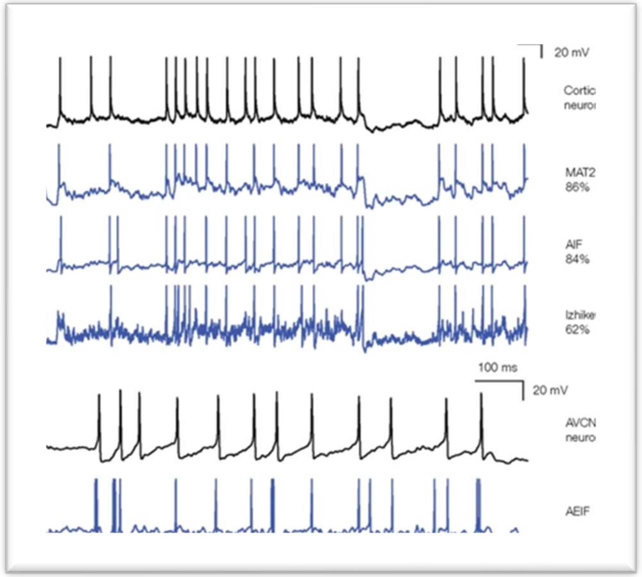
Human Neurons

- We want to make PINNs and DeepONets 1000x faster

Training AlphaGo: 1920 CPUs and 280 GPUs – about 1MegaWatt
Lee Sedol: 1 human brain – about 20 Watts

- Biological plausibility





Conclusions

- Physics-informed neural networks (PINNs) provide a flexible framework for solving both forward and inverse problems, particularly in high-dimensional and data-sparse regimes.

- Physics-informed neural operators can serve as fast and accurate surrogate models for conventional numerical solvers, enabling efficient evaluation across multiple problem configurations. Optimal architecture for any problem should be driven by the Physics

- No single architecture is universally optimal: network design, constraints, and training strategies should be guided by the governing physics, data structure, and quantities of interest.

Physics should drive the model design.

Acknowledgement

- Prof. George Em Karniadakis, Brown University
- Qile Jiang, Phd Candidate, Brown University

Thank you!