Supervised learning in function spaces

Part II: Deep Operator Networks

https://github.com/PredictiveIntelligenceLab/TRIPODS_Winter_School_2022

Instructors:

- Paris Perdikaris (University of Pennsylvania, pgp@seas.upenn.edu)
- Jacob Seidman (University of Pennsylvania, seidj@sas.upenn.edu)
- Georgios Kissas (University of Pennsylvania, gkissas@seas.upenn.edu)

2022 TRIPODS Winter School & Workshop on Interplay between Artificial Intelligence (AI) and Dynamical Systems
Mathematical Institute for Data Science (MINDS)
Johns Hopkins University

Supervised Operator Learning
Problem Formulation

- Given a dataset of N pairs of input and output functions

$$\{(u^1, s^1), \ldots, (u^N, s^N)\}$$

learn an operator

$$\mathcal{F} : C(\mathcal{X}, \mathbb{R}^{d_u}) \to C(\mathcal{Y}, \mathbb{R}^{d_s})$$

such that

$$\mathcal{F}(u^i) = s^i, \quad \forall i$$

How do we design an architecture for Operators?

$$F(u)(y) = \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right) \sigma(w_k^\top y + \zeta_k)$$

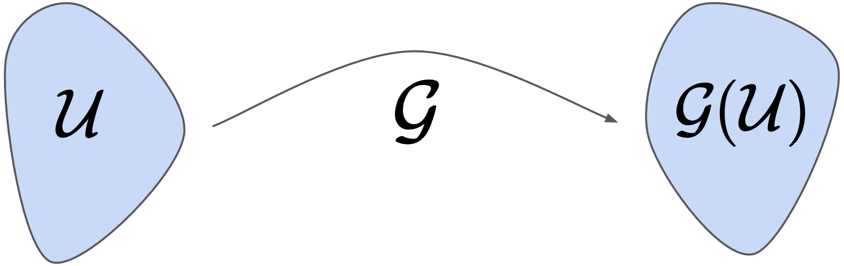
Theorem (Chen & Chen ‘95): If $\mathcal{X} \subset \mathbb{R}^{d_x}, \mathcal{Y} \subset \mathbb{R}^{d_y}, \mathcal{U} \subset C(U, \mathbb{R})$ are all compact, given a continuous $\mathcal{G} : \mathcal{U} \to C(D, \mathbb{R})$, for any $\epsilon > 0,$ there exists F of the above form such that

$$\sup_{u \in \mathcal{U}} \sup_{y \in \mathcal{Y}} \|F(u)(y) - \mathcal{G}(u)(y)\| < \epsilon$$

Chen, Tianping, 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 (1995): 911-917.

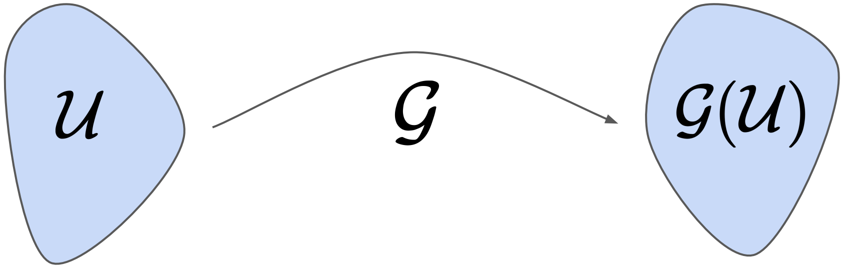
Motivating Chen & Chen Architecture

- Consider a continuous operator $\mathcal{G} : \mathcal{U} \to C(\mathcal{Y}, \mathbb{R})$ with $\mathcal{U} \subset C(\mathcal{X}, \mathbb{R})$ compact



- Given $u \in \mathcal{U}$, we want to be able to evaluate $\mathcal{G}(u)(y)$ for any $y \in \mathcal{Y}$

Thus, our aim is to approximate functions in the set $\mathcal{G}(\mathcal{U})$



Since $\mathcal{U}$ is compact and $\mathcal{G}$ is continuous, $\mathcal{G}(\mathcal{U})$ is also compact

We can find a finite dimensional subspace of $\mathcal{G}(\mathcal{U})$ that is ϵ-close to any $\mathcal{G}(u)$

This subspace of functions has a basis of functions $\{t_1(y), \ldots, t_d(y)\}$

So for every $\mathcal{G}(u)$ there are coordinates in this basis $\{b_1(u), \ldots, b_d(u)\}$ such that

$$\Big\| \sum_{k=1}^{d} b_k(u) t_k(y) - \mathcal{G}(u)(y) \Big\|_\infty < \epsilon$$

Proposed Architecture

$$\sum_{k=1}^{d} b_k(u)t_k(y)$$

Approximate with neural network sending

$$\begin{bmatrix} u(x_1) \\ \vdots \\ u(x_m) \end{bmatrix} \longmapsto \begin{bmatrix} b_1(u) \\ \vdots \\ b_d(u) \end{bmatrix}$$

$$b_k(u) = \sum_{i=1}^{n} c_i^k \sigma\left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k\right)$$

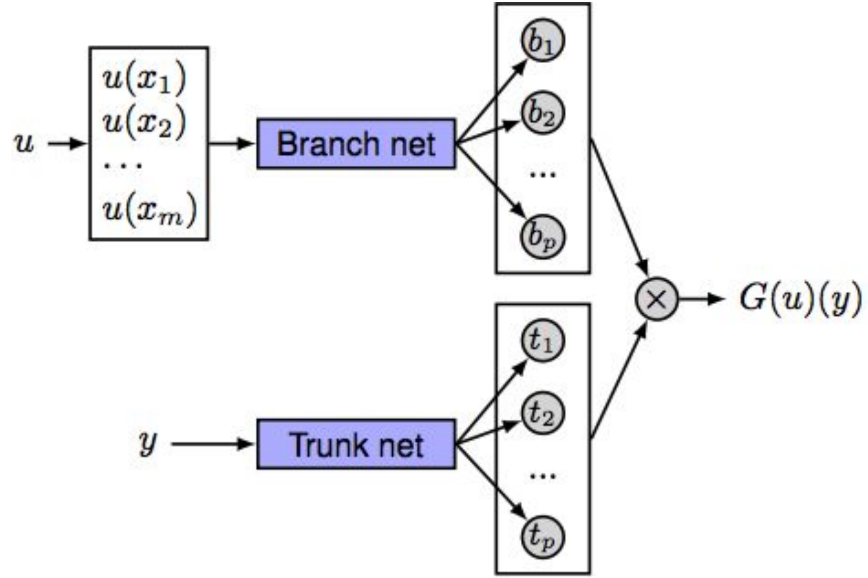
Approximate with neural network sending

$$y \longmapsto \begin{bmatrix} t_1(y) \\ \vdots \\ t_d(y) \end{bmatrix}$$

$$t_k(y) = \sigma(w_k^\top y + \zeta_k)$$

An Initial Approach (Chen and Chen, 1995)

$$F(u)(y) = \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right) \sigma(w_k^\top y + \zeta_k)$$



Chen, Tianping, 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 (1995): 911-917.

An Initial Approach (Chen and Chen, 1995)

$$F(u)(y) = \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right) \sigma(w_k^\top y + \zeta_k)$$

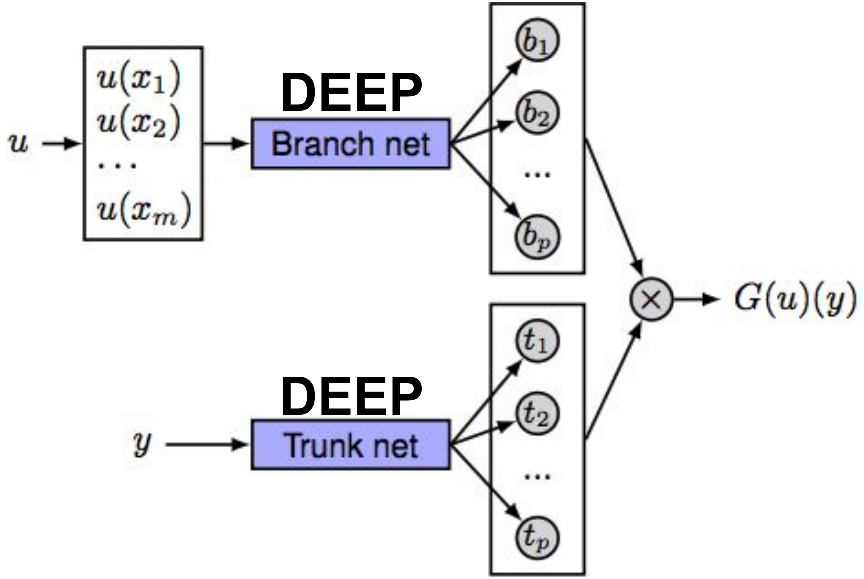
Theorem (Chen & Chen '95): If $\mathcal{X} \subset \mathbb{R}^{d_x}, \mathcal{Y} \subset \mathbb{R}^{d_y}, \mathcal{U} \subset C(U, \mathbb{R}^{d_u})$ are all compact, given a continuous $\mathcal{G} : \mathcal{U} \to C(D, \mathbb{R})$, for any $\epsilon > 0,$ there exists F of the above form such that

$$\sup_{u \in \mathcal{U}} \sup_{y \in \mathcal{Y}} \|F(u)(y) - \mathcal{G}(u)(y)\| < \epsilon$$

Chen, Tianping, 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 (1995): 911-917.

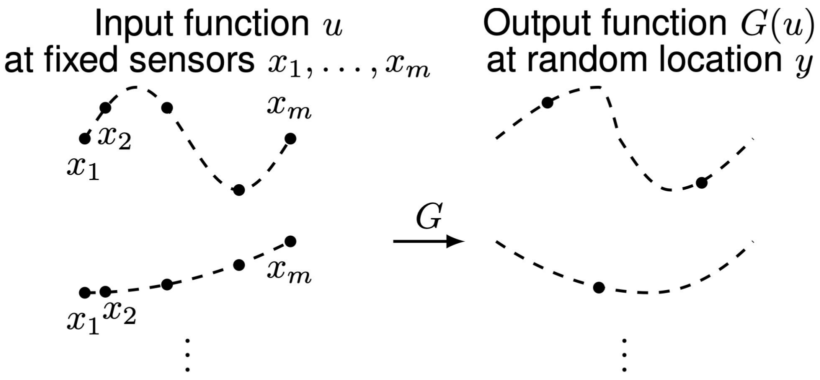
DeepONet: A Modern Extension

Everything is a ***deep*** network!



Lu, Lu, et al. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators." *Nature Machine Intelligence* 3.3 (2021): 218-229.

Training a DeepOnet



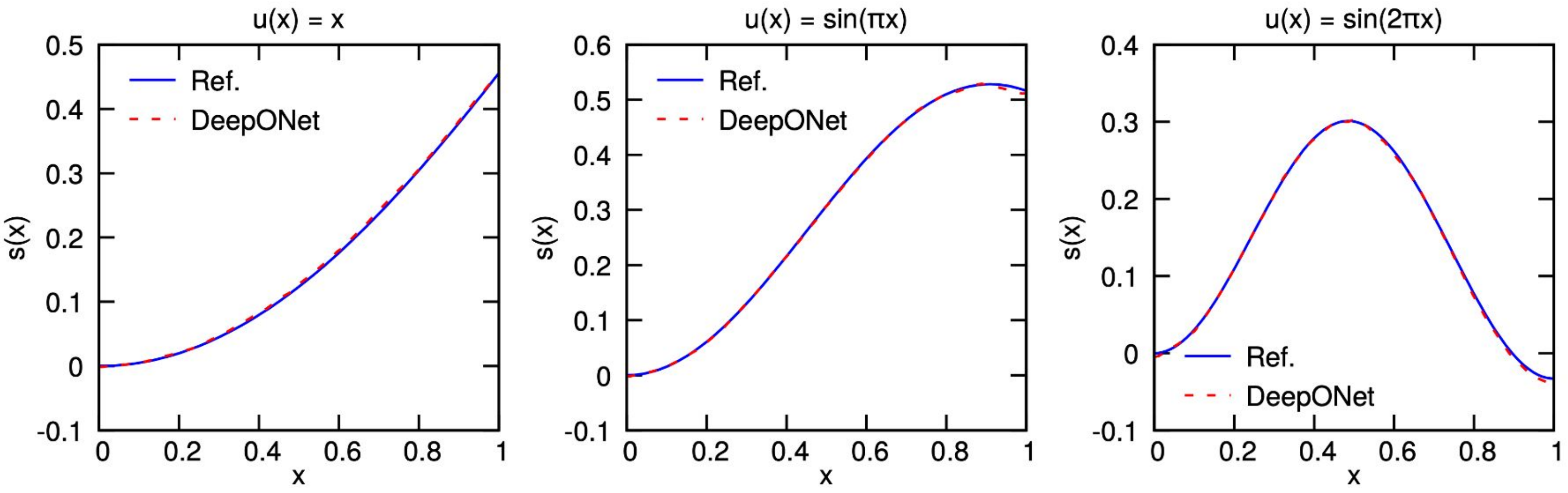
$$[\boldsymbol{u}, \boldsymbol{y}, G(\boldsymbol{u})(\boldsymbol{y})] = \left[\begin{bmatrix} \vdots \\ \boldsymbol{u}^{(i)}(\boldsymbol{x}_1), \boldsymbol{u}^{(i)}(\boldsymbol{x}_2), \cdots, \boldsymbol{u}^{(i)}(\boldsymbol{x}_m) \\ \boldsymbol{u}^{(i)}(\boldsymbol{x}_1), \boldsymbol{u}^{(i)}(\boldsymbol{x}_2), \cdots, \boldsymbol{u}^{(i)}(\boldsymbol{x}_m) \\ \vdots \\ \boldsymbol{u}^{(i)}(\boldsymbol{x}_1), \boldsymbol{u}^{(i)}(\boldsymbol{x}_2), \cdots, \boldsymbol{u}^{(i)}(\boldsymbol{x}_m) \\ \vdots \end{bmatrix}, \begin{bmatrix} \vdots \\ \boldsymbol{y}_1^{(i)} \\ \boldsymbol{y}_2^{(i)} \\ \vdots \\ \boldsymbol{y}_P^{(i)} \\ \vdots \end{bmatrix}, \begin{bmatrix} \vdots \\ G(\boldsymbol{u}^{(i)})(\boldsymbol{y}_1^{(i)}) \\ G(\boldsymbol{u}^{(i)})(\boldsymbol{y}_2^{(i)}) \\ \vdots \\ G(\boldsymbol{u}^{(i)})(\boldsymbol{y}_P^{(i)}) \\ \vdots \end{bmatrix} \right]$$

$$\mathcal{L}_{\text{operator}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left| G_\theta\left(\boldsymbol{u}^{(i)}\right)\left(\boldsymbol{y}^{(i)}\right) - \boldsymbol{s}^{(i)}\left(\boldsymbol{y}^{(i)}\right) \right|^2$$

A pedagogical example

$$\frac{ds(x)}{dx} = u(x), \quad x \in [0,1]$$
$$s(0) = 0$$

$$G : u(x) \longrightarrow s(x) = s(0) + \int_0^x u(t)dt, \quad x \in [0,1]$$



*Courtesy of
G.E. Karniadakis
(Brown)

Applications highlights

Simulation of Electro-Convection

$$\frac{1}{Sc}\left[\frac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right]=-\nabla p+\nabla^2\mathbf{u}+\mathbf{f}_e,$$

$$\nabla\cdot\mathbf{u}=0,$$

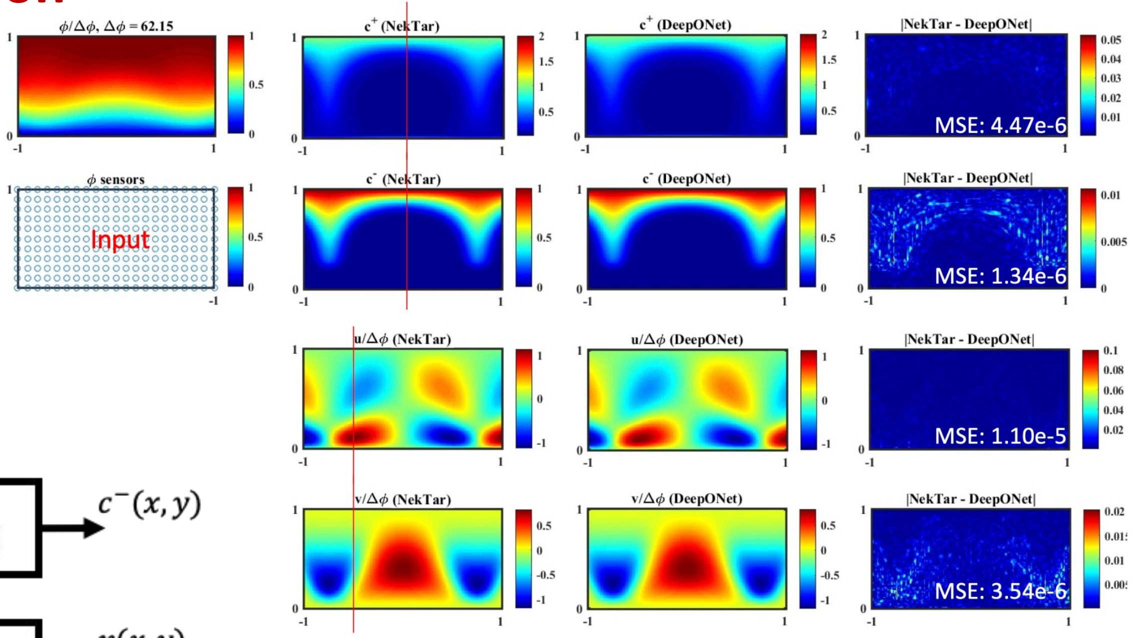
$$-2\epsilon^2\nabla^2\phi=\rho_e,$$

$$\frac{\partial c^{\pm}}{\partial t}=-\nabla\cdot\mathbf{j}^{\pm},$$

$$\mathbf{j}^{\pm}=c^{\pm}\mathbf{u}-\nabla c^{\pm}\mp c^{\pm}\nabla\phi.$$

$$\phi(x,y) \longrightarrow \boxed{G_1} \longrightarrow c^{+}(x,y) \qquad \phi(x,y) \longrightarrow \boxed{G_2} \longrightarrow c^{-}(x,y)$$

$$\phi(x,y) \longrightarrow \boxed{G_3} \longrightarrow u(x,y) \qquad \phi(x,y) \longrightarrow \boxed{G_4} \longrightarrow v(x,y)$$



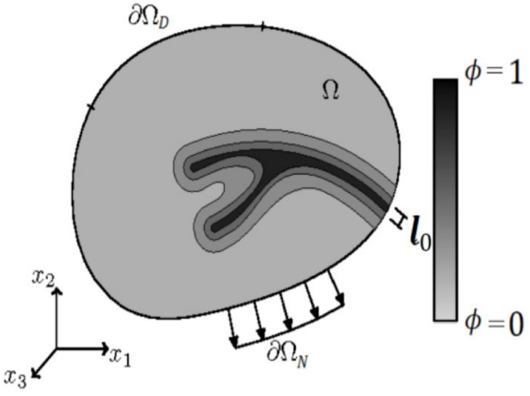
Speed-up, 10,000X: DeepOnet – 0.1s; CFD – hours

Cai S, Wang Z, Lu L, Zaki TA, Karniadakis GE. DeepM&Mnet: Inferring the electroconvection multi-physics fields based on operator approximation by neural networks. Journal of Computational Physics. 2021

*Courtesy of
G.E. Karniadakis
(Brown)

Applications highlights

Phase field modeling of fracture



$$H(x,t) = \max_{s\in[0,t]} \psi^+\left(\epsilon(x,s)\right)$$

Tensile strain energy functional

$\phi \rightarrow 1$ as $\psi^+(\epsilon) \rightarrow \infty$

Initial strain history is defined in terms of distance from the crack to initialize the crack on the domain.

Governing equations

$$-\nabla\cdot g(\phi)\boldsymbol{\sigma} = \boldsymbol{f} \text{ on } \Omega$$

Degradation function

Critical energy release rate

$$\frac{G_c}{l_0}\phi - G_c l_0 \nabla^2\phi = -g'(\phi)H(x,t) \text{ on } \Omega$$

History function

Boundary Conditions

$$g(\phi)\boldsymbol{\sigma}\cdot\boldsymbol{n} = \boldsymbol{t} \text{ on } \partial\Omega_N$$
$$\boldsymbol{u} = \bar{\boldsymbol{u}} \text{ on } \partial\Omega_D$$

$$\nabla\phi\cdot\boldsymbol{n} = 0 \text{ on } \partial\Omega$$

Weak form

$$\boldsymbol{\Psi_e} = \int g(\phi)\psi_0\, d\Omega$$

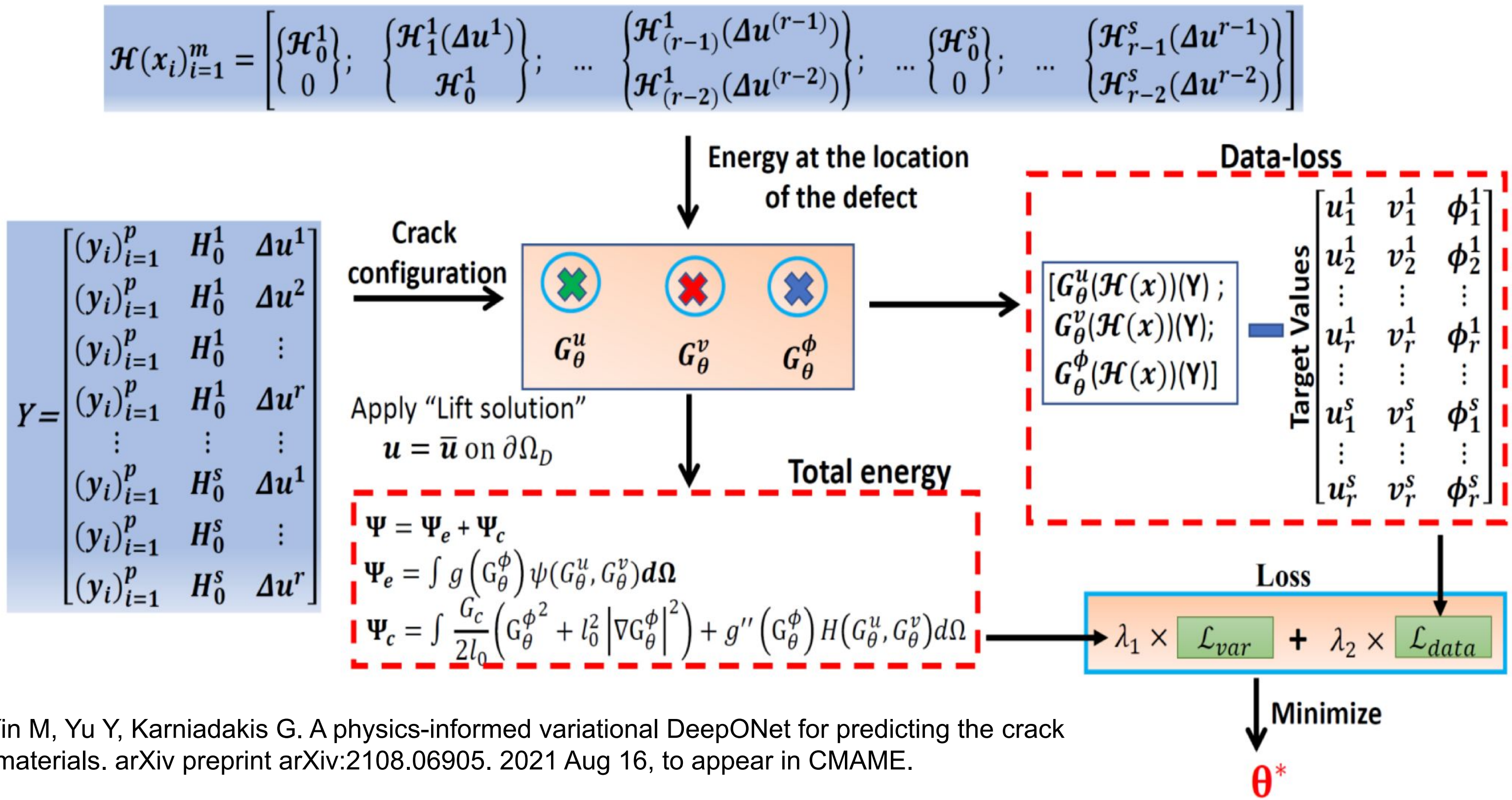
$$\boldsymbol{u} = \bar{\boldsymbol{u}} \text{ on } \partial\Omega_D$$

$$\boldsymbol{\Psi_c} = \int \frac{G_c}{2l_0}\left(\phi^2 + l_0^2|\nabla\phi|^2\right) + g''(\phi)H(x,t)d\Omega$$

*Courtesy of G.E. Karniadakis (Brown)

Applications highlights

Phase field modeling of fracture

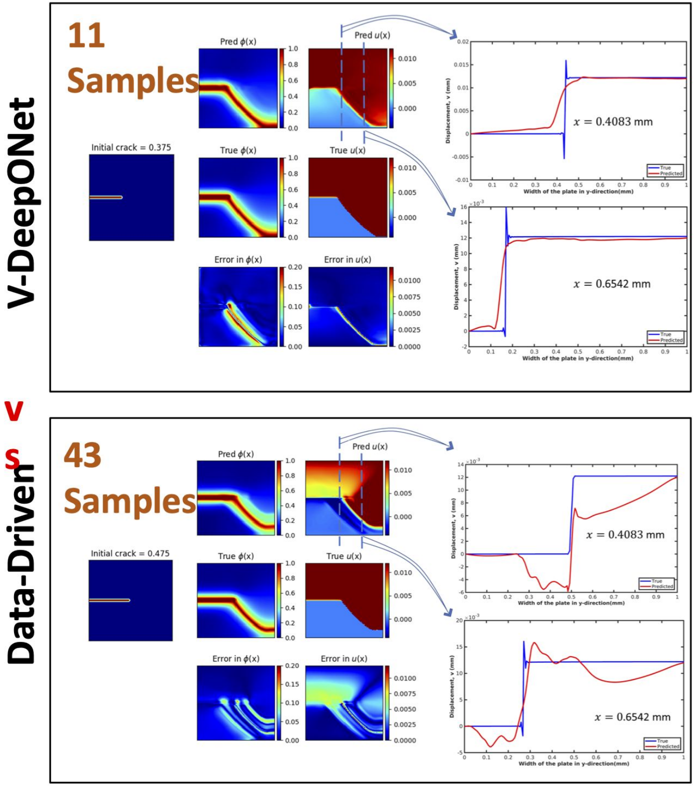


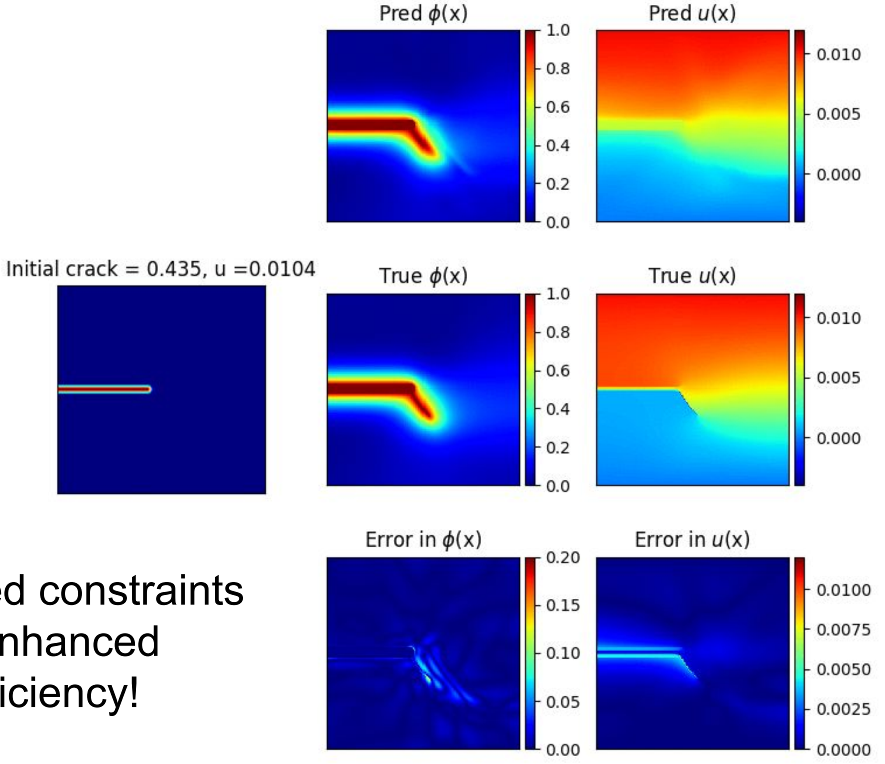
Goswami S, Yin M, Yu Y, Karniadakis G. A physics-informed variational DeepONet for predicting the crack path in brittle materials. arXiv preprint arXiv:2108.06905. 2021 Aug 16, to appear in CMAME.

*Courtesy of
G.E. Karniadakis
(Brown)

Applications highlights

Phase field modeling of fracture





Physics-based constraints lead to enhanced data-efficiency!

Some drawbacks

- Number of “sensor points” $\{x_i\}_{i=1}^{p}$ fixed
- Sensor locations themselves also fixed
- Would have to retrain a new branch and trunk network if either of these change

NEXT UP

- DeepONet in JAX
  - A more in-depth introduction to JAX and an example implementation of the DeepONet method.

THEN

- Introduction to Neural Operators and an implementation of the Fourier Neural operators in JAX.