Learning Operators for Forward and Inverse Problems

Nikola B. Kovachki

Computing and Mathematical Sciences
California Institute of Technology

Isaac Newton Institute for Mathematical Sciences
Deep Learning and Inverse Problems
Sep 30th, 2021

Table of Contents

Caltech

The Problem

Setting

Input-Output Map: $\Psi^\dagger : \mathcal{X} \to \mathcal{Y}$, Separable Banach Spaces

Data: $\{x_n, y_n\}_{n=1}^N, \;\; y_n = \Psi^\dagger(x_n),$

$x_n \overset{i.i.d.}{\sim} \mu$ or $\{x_n\} \subset K$ compact

Goal

Parameter Space $\Theta \subseteq \mathbb{R}^p$

Operator Class: $\Psi : \mathcal{X} \times \Theta \to \mathcal{Y}$

Operator Approximation: $\Psi(\cdot;\theta^\star) \approx \Psi^\dagger$

Key Idea

Design Architecture On Banach Space Then Discretize

Example

Elliptic PDE

$$\begin{aligned} -\nabla \cdot (a \nabla u) &= f \quad \text{in } D \\ u &= g \quad \text{in } \partial D \end{aligned}$$

Operator Of Interest

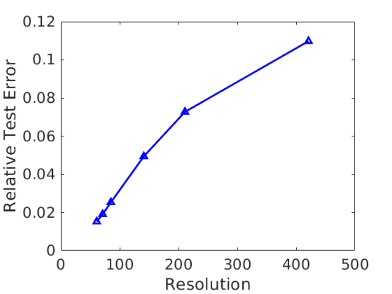
Nonlinear $\Psi^\dagger : a \in L^\infty(D) \mapsto u \in H^1(D)$

Error Metric

$$\|\Psi^\dagger - \Psi\|_{L^p_\mu(\mathcal{X};\mathcal{Y})} \quad \text{or} \quad \sup_{x \in K} \|\Psi^\dagger(x) - \Psi(x)\|_{\mathcal{Y}}$$

Key Idea

Example What Goes Wrong If You Discretize And Then Apply Standard Neural Network Algorithms



[1] Y Zhu and N Zabaras

Design Architecture On Banach Space Then Discretize

Model Reduction Approach

K Bhattacharya, B Hosseini, NB Kovachki and AM Stuart

Model Reduction And Neural Networks For Parametric PDEs

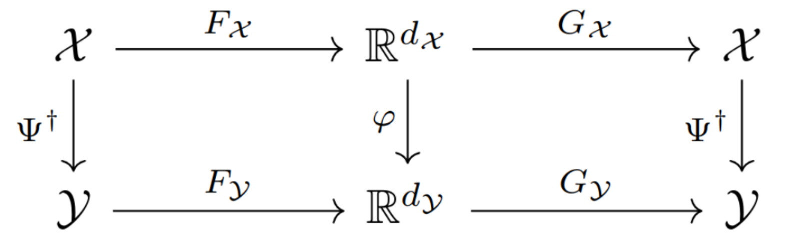
SMAI-JCM 7, 121-157.

Neural Operator: Neural Networks For Maps Between Function Spaces

arXiv:2108.08481.

The Idea

In A Picture



In Equations

$$G_{\mathcal{X}} \circ F_{\mathcal{X}} \approx I_{\mathcal{X}}$$

$$G_{\mathcal{Y}} \circ F_{\mathcal{Y}} \approx I_{\mathcal{Y}}$$

$$G_{\mathcal{Y}} \circ \varphi \circ F_{\mathcal{X}} \approx \Psi^\dagger$$

Existence of the Approximation

Lemma

- $\mathcal{X}$, $\mathcal{Y}$ Banach spaces with the *approximation property* (AP).
- $\Psi^\dagger : \mathcal{X} \to \mathcal{Y}$ continuous.

For any $K \subset \mathcal{X}$ compact and $\epsilon > 0$ there exist bounded linear maps $F_\mathcal{X} : \mathcal{X} \to \mathbb{R}^{d_\mathcal{X}}$, $G_\mathcal{Y} : \mathbb{R}^{d_\mathcal{Y}} \to \mathcal{Y}$, and a continuous map $\varphi \in C(\mathbb{R}^{d_\mathcal{X}}; \mathbb{R}^{d_\mathcal{Y}})$ such that

$$\sup_{x \in K} \|\Psi^\dagger(x) - (\mathcal{G}_\mathcal{Y} \circ \varphi \circ F_\mathcal{X})(x)\|_\mathcal{Y} \leq \epsilon.$$

Lemma

- $\mathcal{X}$ Banach space with AP, $\mathcal{Y}$ separable Hilbert space.
- μ probability measure on $\mathcal{X}$.
- $\Psi^\dagger \in L^p_\mu(\mathcal{X}; \mathcal{Y})$ for $1 \leq p < \infty$.

As before,

$$\|\Psi^\dagger - G_\mathcal{Y} \circ \varphi \circ F_\mathcal{X}\|_{L^p_\mu(\mathcal{X};\mathcal{Y})} \leq \epsilon.$$

PCA-NET

Architecture

$$\Psi_{PCA}(x;\theta)(s) = \sum_{j=1}^{m} \alpha_j(Lx;\theta)\psi_j(s), \quad \forall x \in \mathcal{X} \qquad s \in D.$$

Details

- Lx maps to PCA coefficients under μ.
- $\{\psi_j\}$ are PCA basis functions under $(\Psi^\dagger)^\sharp\mu$.
- $\alpha : \mathbb{R}^d \to \mathbb{R}^m$ finite dimensional neural network.
- Non-intrusive reduced basis method.

Universal Approximation

Theorem

Let $\Psi^\dagger \in L^p_\mu(\mathcal{X};\mathcal{Y})$. For any $\epsilon > 0$, there are dimensions $(d_\mathcal{X}, d_\mathcal{Y})(\epsilon)$, a requisite amount of data $N = N(d_\mathcal{X}, d_\mathcal{Y})$, and a neural network ψ depending on $\epsilon, d_\mathcal{X}, d_\mathcal{Y}$ such that

$$\mathbb{E}_{\{x_j\}\sim\mu}\|\Psi^\dagger - \Psi_{NN}\|_{L^p_\mu(\mathcal{X};\mathcal{Y})} \leq \epsilon$$

where $\Psi_{NN} = G_\mathcal{Y} \circ \psi \circ F_\mathcal{X}$ with $G_\mathcal{Y}$ and $F_\mathcal{X}$ defined by PCA.

Example: Darcy

Elliptic PDE

$$-\nabla \cdot (a \nabla u) = 1, \quad s \in D = (0,1)^2$$
$$u = 0, \quad s \in \partial D.$$

Operator Of Interest

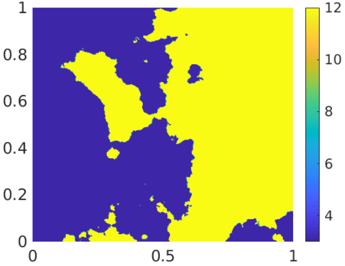
Nonlinear $\quad \Psi^\dagger : a \in L^\infty(D) \mapsto u \in H_0^1(D).$

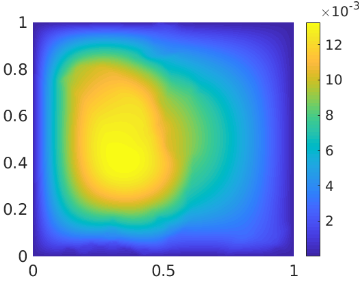
Example: Darcy (piecewise-constant)

Input-Output

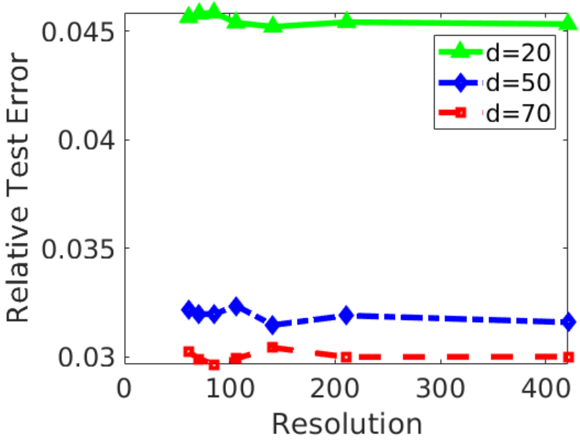
Input: $a \in L^\infty(D)$ (Left),

Output: $u \in H_0^1(D)$. (Right),



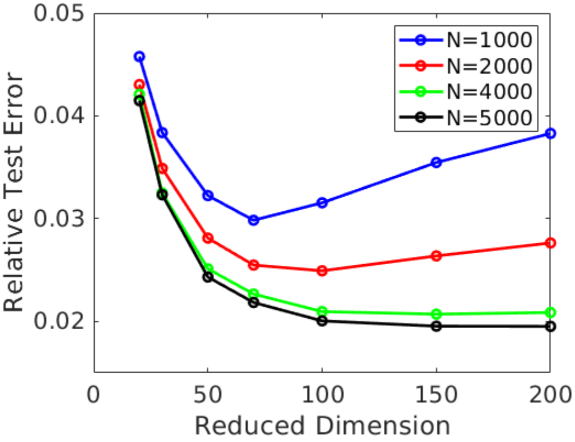


Example: Darcy (piecewise-constant)



Architecture defined on function space

Example: Darcy (piecewise-constant)



Care needed in relating N and d

Application to Crystal Plasticity

BG Liu, NB Kovachki, Z Li, K Azizzadenesheli,
A Anandkumar, AM Stuart and K Bhattacharya
A learning-based multiscale method and its application to inelastic impact problems
arXiv:2102.07256.

Mechanical Model

Nonlinear PDE

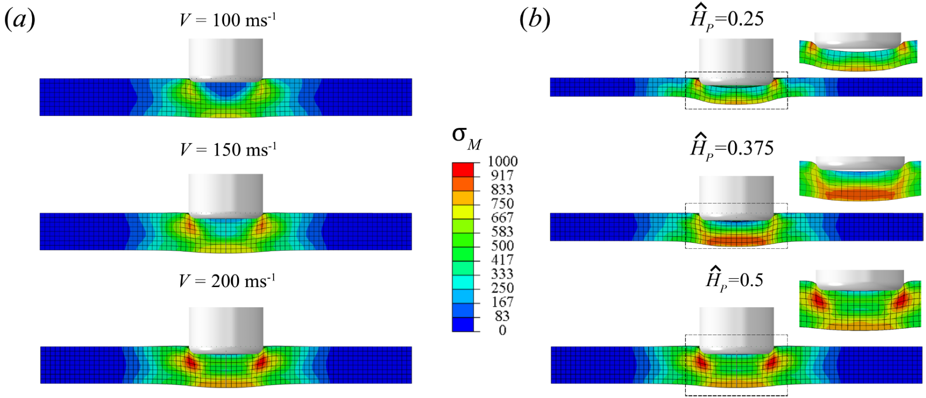
$$
\begin{aligned}
\rho_0(x)\ddot{u}(s,t) &= \nabla_s \cdot P[\nabla u](s,t) + \rho_0(s)b(s,t), && (s,t) \in D \times [0,T] \\
(P[\nabla u]n)(s,t) &= h(s,t), && (s,t) \in \Gamma_a \times [0,T] \\
u(s,t) &= g(s,t) && (s,t) \in \Gamma_b \times [0,T] \\
u(s,0) &= x, && s \in D \\
\dot{u}(s,0) &= u_0(s), && s \in D
\end{aligned}
$$

Operator Of Interest

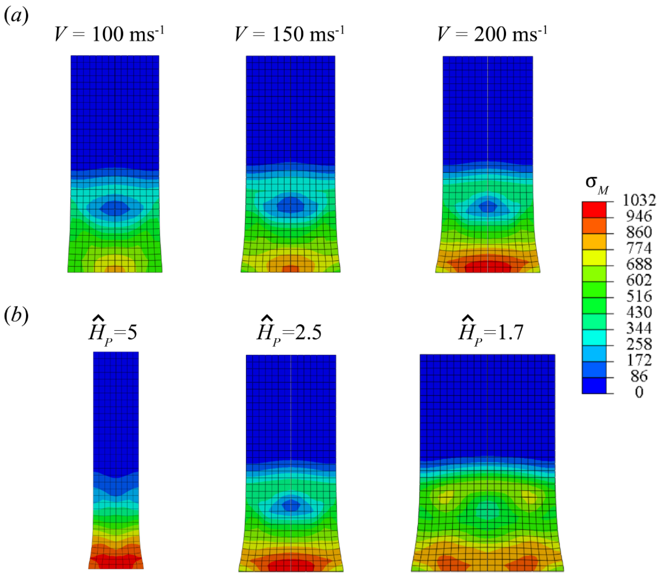
- Homogenization: microscale computations define constitutive law.
- Map strain on each cell boundary to interior stress:

$$\Psi^\dagger : \{[0,T] \to \mathbb{R}^{3\times 3}\} \to \{[0,T] \to \mathbb{R}^{3\times 3}\}$$

Taylor Anvil Test



Blunt Impact



Computational Cost

Micromechancial model	**Blunt impact**	**Taylor Anvil**
Surrogate	8.6×10^{-2}	4.1×10^{-2}
Taylor RVE	9.0×10^{1}	4.2×10^{1}
Periodic RVE	2.5×10^{6}	1.2×10^{6}

Table: Computational cost per time step in seconds on a single CPU.

Neural Operator: A Kernel Network Approach

Z Li, NB Kovachki, K Azizzadenesheli,
BG Liu, K Bhattacharya, AM Stuart and A Anandkumar
Neural Operator: Graph Kernel Network for Partial Differential Equations
arXiv:2003.03485
Multipole Graph Neural Operator for Parametric Partial Differential Equations
NeurIPS (2020). arXiv:2006.09535
Fourier Neural Operator for Parametric Partial Differential Equations
ICLR (2021). arXiv:2010.08895
Neural Operator: Neural Networks For Maps Between Function Spaces
arXiv:2108.08481.

NB Kovachki, S Lanthaler, S Mishra
On universal approximation and error bounds for Fourier Neural Operators
arXiv:2107.07562

Linear Approximation Theory

Linear Approximation

Input-Output Map: $\Psi^\dagger : \mathcal{X} \to \mathcal{Y}$, Separable Banach Spaces

Basis: $\text{span}\{\varphi_1, \varphi_2, \dots\} = \mathcal{Y}$

Solution Manifold: $\mathcal{M} = \{\Psi^\dagger(x) : x \in \mathcal{X}\} \subset \mathcal{Y}$

n-term Approximation: $\sum_{j=1}^{n} \alpha_j \varphi_j, \quad \alpha_j \in \mathbb{R}$

Approximation Space: $V_n = \text{span}\{\varphi_1, \dots, \varphi_n\}$

Kolgomorov n-width: $d_n(\mathcal{M})_\mathcal{Y} := \inf_{\dim(V_n) \le n} \sup_{v \in \mathcal{M}} \min_{w \in V_n} \|v - w\|_\mathcal{Y}$

PCA: $\sum_{j=n+1}^{\infty} \lambda_j \le d_n(\mathcal{M})^2_\mathcal{Y}$

Motivation

If $\mathcal{M}$ is not well approximated by a linear space, need non-linear approximation.

Neural Networks in Infinite Dimensions

Classical Neural Networks

$$v_0 = x$$

$$v_{l+1} = \sigma\big(A_l v_l + b_l\big), \quad l = 0, \dots, L-1$$

$$y = A_L v_L + b_L$$

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

In Function Space

$$\{\sigma : \mathbb{R} \to \mathbb{R}\} \mapsto \{\sigma(x)(s) = \sigma(x(s))\} \quad \text{(Nemytskii)}$$

$$\{A_l \in \mathbb{R}^{d_{l+1} \times d_l}\} \mapsto \left\{(A_l x) = \int_D \kappa_l(\cdot, z) x(z)\, dz\right\} \quad \text{(Integral Kernel Operator)}$$

New Architecture

Input: $x : D \subset \mathbb{R}^d \to \mathbb{R}^m$

Output: $y : D' \subset \mathbb{R}^{d'} \to \mathbb{R}^r$

Iteration:

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

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

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

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

Approximation Map

$$(\Psi(x))(s) := y(s)$$

Connection to Transformers

One Layer Update

$$x : D \subset \mathbb{R}^d \to \mathbb{R}^n$$

$$y(s) = \sigma\left(Wx(s) + \int_D \kappa(s,z)x(z)\,dz + b(s)\right)$$

Transformer Structure

$$\kappa(s,z) \mapsto \kappa_x(x(s),x(z)),$$

$$\kappa_x(x(s),x(z)) = g_x(x(s),x(z))V, \qquad V \in \mathbb{R}^{n\times n},\ g_x : \mathbb{R}^{2n} \to \mathbb{R}$$

$$g_x(x(s),x(z)) = \left(\int_D \exp\left(\frac{\langle Qx(r), Kx(z)\rangle}{\sqrt{n}}\right) dr\right)^{-1} \exp\left(\frac{\langle Qx(s), Kx(z)\rangle}{\sqrt{n}}\right)$$

Uniform Approximation over Compacta

Assumption

$D \subset \mathbb{R}^d$ and $D' \subset \mathbb{R}^{d'}$ bounded Lipschitz domains.

- $\mathcal{X} = L^{p_1}(D)$ for $1 \leq p_1 < \infty$.
- $\mathcal{X} = W^{k_1,p_1}(D)$ for $1 \leq p_1 < \infty$.
- $\mathcal{X} = C(D)$.

- $\mathcal{Y} = L^{p_2}(D')$ for $1 \leq p_2 < \infty$.
- $\mathcal{Y} = W^{k_2,p_2}(D')$ for $1 \leq p_2 < \infty$.
- $\mathcal{Y} = C^{k_2}(D')$ for $k_2 \in \mathbb{N}_0$.

$\Psi^\dagger : \mathcal{X} \to \mathcal{Y}$ continuous.

Theorem

For any $K \subset \mathcal{X}$ compact and $\epsilon > 0$, there exists an architecture $\Psi : \mathcal{X} \to \mathcal{Y}$ such that

$$\sup_{x \in K} \|\Psi^\dagger(x) - \Psi(x)\|_{\mathcal{Y}} \leq \epsilon.$$

Bochner Norm Approximation

Assumption

$D \subset \mathbb{R}^d$ and $D' \subset \mathbb{R}^{d'}$ bounded Lipschitz domains.

- $\mathcal{X} = L^{p_1}(D)$ for $1 \leq p_1 < \infty$.
- $\mathcal{X} = W^{k_1,p_1}(D)$ for $1 \leq p_1 < \infty$.
- $\mathcal{X} = C(D)$.
- $\mathcal{Y} = L^2(D')$.
- $\mathcal{Y} = H^{k_2}(D')$.
-

$\Psi^\dagger \in L^{p_3}_\mu(\mathcal{X};\mathcal{Y})$ with μ probability measure on $\mathcal{X}$.

Theorem

For any $\epsilon > 0$, there exists an architecture $\Psi : \mathcal{X} \to \mathcal{Y}$ such that

$$\|\Psi^\dagger - \Psi\|_{L^{p_3}_\mu(\mathcal{X};\mathcal{Y})} \leq \epsilon.$$

Computation

One Layer Update

$$x : D \subset \mathbb{R}^d \to \mathbb{R}^n$$

$$y(s) = \sigma\left(Wx(s) + \int_D \kappa(s,z)x(z)\,dz + b(s)\right)$$

Computing the Integral Kernel

- Restrict integration to balls: $\int_D \to \int_{B_r(s)}$.
- Monte Carlo sampling: $\int_D \approx \frac{|D|}{m}\sum_{j=1}^m$.
- Fast Multiple Method.
- Let $\kappa(s,z) = \kappa(s-z)$, parametrize Fourier components θ:

$$\int_D \kappa(s-z)x(z)\,dz = \mathcal{F}^{-1}(\theta \cdot \mathcal{F}(x)).$$

FNO Universal Approximation

Theorem

- $D \subseteq \mathbb{T}^d$ bounded, Lipschitz domain.
- $E : H^s(D) \to H^s(\mathbb{T}^d)$ linear, periodic extension operator for $s \geq 0$.
- $\Psi^\dagger : H^s(D) \to H^{s'}(D)$ continuous for $s, s' \geq 0$.

For any $K \subset H^s(D)$ compact and $\epsilon > 0$, there exists an architecture $\Psi : H^s(\mathbb{T}^d) \to H^{s'}(\mathbb{T}^d)$ such that

$$\sup_{x \in K} \|\Psi^\dagger(x) - ((\Psi \circ E)(x))|_D\|_{H^{s'}(D)} \leq \epsilon.$$

Steady-State Darcy Flow

PDE

$$-\nabla \cdot (a \nabla u) = f \quad \text{in} \quad \mathbb{T}^d$$

- $s > d/2 + k$ with $k \in \mathbb{N}$ and $f \in \dot{H}^{k-1}(\mathbb{T}^d)$.
- $a \in H^s(\mathbb{T}^d)$ with $a = 1 + \tilde{a}$ such that, for some $\lambda \in (0,1)$,

$$\|a\|_{H^s} \leq \lambda^{-1}, \qquad \|\tilde{a}\|_{L^\infty} \leq 1 - \lambda.$$

- $\Psi^\dagger : a \mapsto u$.

Theorem

For any $\epsilon > 0$, there exists a FNO $\Psi : \mathcal{A}^s_\lambda(\mathbb{T}^d) \to H^1(\mathbb{T}^d)$ such that

$$\sup_{a \in \mathcal{A}^s_\lambda(\mathbb{T}^d)} \|\Psi^\dagger(a) - \Psi(a)\|_{H^1} \leq \epsilon,$$

$$\text{size}(\Psi) \lesssim \epsilon^{-\frac{d}{k}} \log \epsilon^{-1}, \qquad \text{depth}(\Psi) \lesssim \log \epsilon^{-1}.$$

Navier-Stokes/Euler

PDE

$$\partial_t u = -\mathrm{P}(u \cdot \nabla u) + \nu \Delta u \quad \text{in} \quad [0, T] \times \mathbb{T}^d$$

- $u \in C([0, T]; H^s) \cap C^1([0, T]; H^{s-2})$ with $s > d/2 + 2$.
- u_0 divergence-free, mean zero s.t. u exists and is bounded.
- $\Psi^\dagger : u_0 \mapsto u(T, \cdot)$.

Theorem

For any $\epsilon > 0$, there exists a FNO $\Psi : \mathcal{A}^s(\mathbb{T}^d; \mathbb{R}^d) \to L^2(\mathbb{T}^d; \mathbb{R}^d)$ such that

$$\sup_{a \in \mathcal{A}^s(\mathbb{T}^d)} \|\Psi^\dagger(a) - \Psi(a)\|_{L^2} \le \epsilon,$$

$$\text{size}(\Psi) \lesssim \epsilon^{-\left(\frac{1}{2} + \frac{d}{s}\right)} \log \epsilon^{-1}, \qquad \text{depth}(\Psi) \lesssim \epsilon^{-\frac{1}{2}} \log \epsilon^{-1}.$$

Example: Burgers

Burgers' Equation

$$\partial_t u + \partial_z(u^2/2) = \nu \partial_{zz} u, \quad (z,t) \in \mathbb{T}^1 \times (0,1]$$
$$u|_{t=0} = u_0, \qquad z \in \mathbb{T}^1.$$

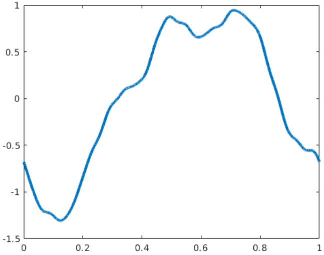
Operator Of Interest

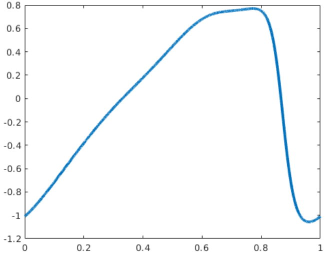
Nonlinear $\Psi^\dagger : u_0 \in L^2(\mathbb{T}^1) \mapsto u|_{t=1} \in H^s(\mathbb{T}^1).$

Example: Burgers

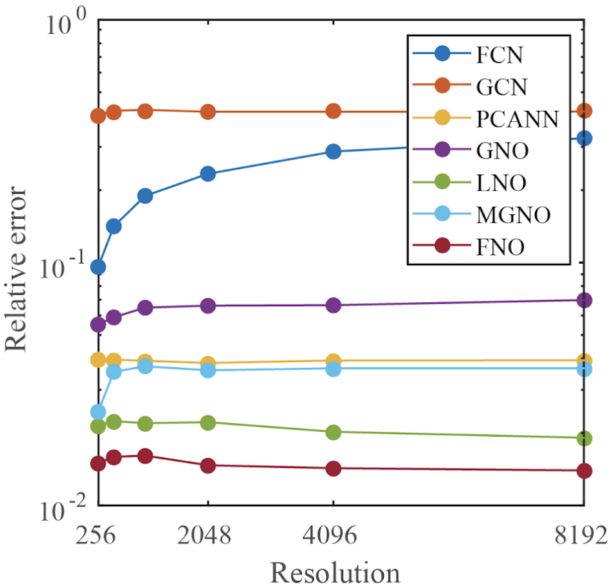
Input-Output

$$\text{Input:} \quad u_0 \in L^2(\mathbb{T}^1) \quad \text{(Left)},$$
$$\text{Output:} \quad u \in H^s(\mathbb{T}^1). \quad \text{(Right)},$$

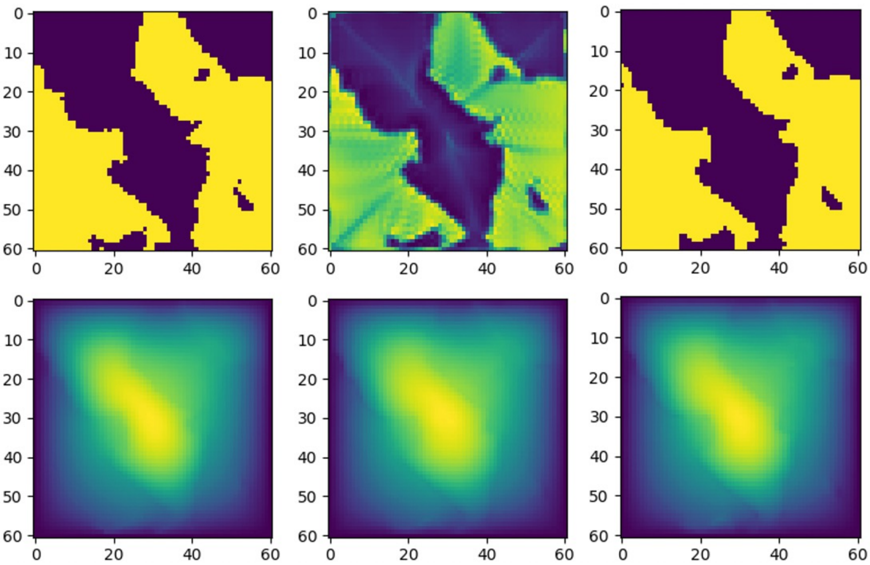




Example: Burgers



Example: Darcy Inverse Problem



Caltech

Application to Turbulent Flow

Z Li, NB Kovachki, K Azizzadenesheli,
BG Liu, K Bhattacharya, AM Stuart and A Anandkumar
Fourier Neural Operator for Parametric Partial Differential Equations
ICLR (2021). arXiv:2010.08895
Markov Neural Operators for Learning Chaotic Systems
arXiv:2106.06898

Fluid Flow on the Torus

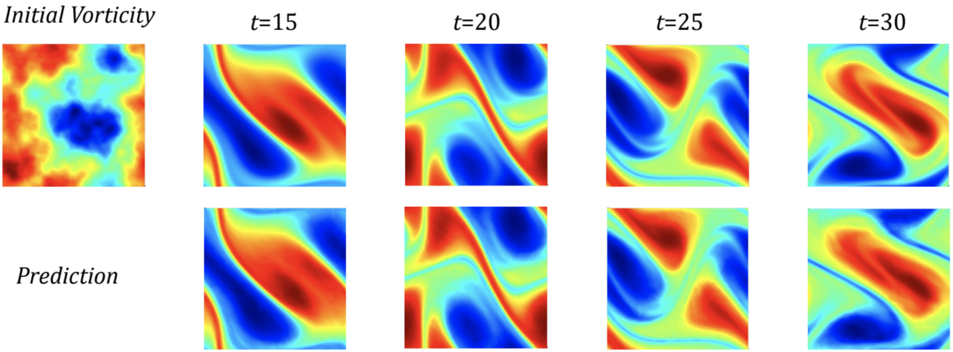
Nonlinear PDE

$$\frac{du}{dt} - \nu \mathrm{P}\Delta u + \mathrm{P}(u \cdot \nabla)u = f, \qquad s \in \mathbb{T}^2, t \in [0, T]$$
$$u(0) = u_0 \qquad s \in \mathbb{T}^2$$

Operator Of Interest

$$\omega = \nabla \times u$$
$$\Psi^\dagger : \omega|_{t=0} \in L^2(\mathbb{T}^2) \mapsto w|_{t=T} \in H^s(\mathbb{T}^2)$$

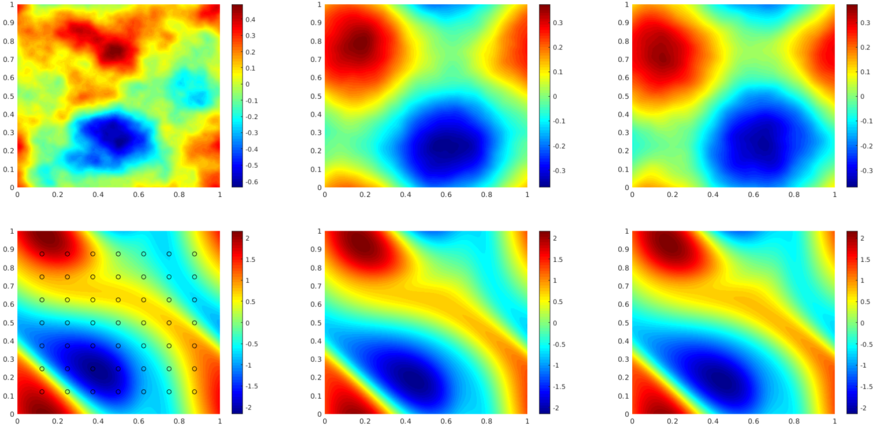
Predicting Trajectories



- $\mathrm{Re} = \mathcal{O}(10^3)$, $N = 10,000$, error $= 3\%$ in H^1.
- Trained on 64×64 grid and evaluated on 256×256.

Caltech

An Inverse Problem



- MCMC: 2 minutes for 50,000 samples with FNO, 30 hours with a pseudo-spectral solver.

Learning Linear Operators from Noisy Data

MV de Hoop, NB Kovachki, NH Nelsen, AM Stuart

Convergence Rates for Learning Linear Operators from Noisy Data

arXiv:2108.12515.

Caltech

The Problem

Setting

Noisy Linear Map: $L^\dagger : \mathcal{X} \to \mathcal{Y},\ \ y = L^\dagger x + \eta$

Assumptions: $\pi(dx, dy) :\ x \sim \mu = N(0, C),\ \eta \sim N(0, \Gamma),\ x \perp \eta$

Data: $\{x_n, y_n\}_{n=1}^N \overset{i.i.d.}{\sim} \pi$

Risk

Expected Risk: $e_\infty(L) = \mathbb{E}_{\{x,y\}\sim\pi} \frac{1}{2}\|Lx\|_{\mathcal{Y}}^2 - \langle y, Lx\rangle_{\mathcal{Y}}$

Empirical Risk: $e_N(L) = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{2}\|Lx_n\|_{\mathcal{Y}}^2 - \langle y_n, Lx_n\rangle_{\mathcal{Y}}\right] + \|L\|_{\text{CM}}^2$

Optimizers: $\widehat{L} = \inf_L e_\infty(L), \quad \widehat{L}^{(N)} = \inf_L e_N(L)$

Statistical Approximation

Theorem

Excess Risk: $e_\infty(\widehat{L}^{(N)}) - e_\infty(\widehat{L}) = \|\widehat{L}^{(N)} - \widehat{L}\|^2_{L^2_\mu(\mathcal{X},\mathcal{Y})}$

Theorem

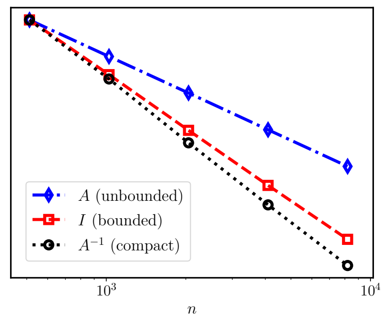
$$
\begin{aligned}
\text{BIP:}\quad & Y = R_X L + E, \quad X \sim N(0, C)^{\otimes N}, E \sim N(0, \Gamma)^{\otimes N} \\
\text{Posterior:}\quad & L|Y, X \sim \nu^{Y,X} \\
\text{Expectation:}\quad & \mathbb{E} = \mathbb{E}^{\{x_n, y_n\} \sim \pi^{\otimes N}} \mathbb{E}^{\nu^{Y,X}} \\
\text{Error:}\quad & C_1 N^{-\alpha} \le \mathbb{E}\|L - \widehat{L}\|^2_{L^2_\mu(\mathcal{X},\mathcal{Y})} \le C_2 N^{-\alpha}, \quad \forall N \ge N_c \\
\text{Error:}\quad & C_1 N^{-\alpha} \le \|\widehat{L}^{(N)} - \widehat{L}\|^2_{L^2_\mu(\mathcal{X},\mathcal{Y})} \le C_2 N^{-\alpha}, \quad \forall N \ge N_c, \text{ w.h.p.}
\end{aligned}
$$

Numerical Illustration

Learning Compact, Bounded and Unbounded Operators

$$A := -\Delta, \quad D(A) = H^2(I) \cap H_0^1(I), \quad I = (0,1).$$

$$L = A,\ \mathrm{Id},\ A^{-1}.$$



Conclusions

1. Neural networks: empirical success in function approximation.
2. Typically:
   - regression: $\mathbb{R}^m \mapsto \mathbb{R}^n$;
   - classification: $\mathbb{R}^m \mapsto \{1, \ldots, K\}$.
3. We consider: $\mathcal{X} \mapsto \mathcal{Y}$, $\mathcal{X}, \mathcal{Y}$ function spaces.
4. Key Idea: Conceive of architecture then discretize.
5. Purely data-driven ("equation-free").
6. Applications: PDEs (model available), Cyber-physical systems, Imaging, Time-series (no model available).
7. Less data needed to learn the forward than the inverse operator.
8. Future work: theory for data needed to achieve given error, posterior consistency for inverse problems.

Caltech

References

[1] M Raissi, P Perdikaris, GE Karniadakis
Physics-informed neural networks...
J. Comp. Phys. 2019.

[2] Weinan E and B Yu
The Deep Ritz Method...
Communications in Mathematics and Statistics 2018.

[3a] Y Zhu and N Zabaras
Bayesian Deep Convolutional Encoder-Decoder Networks...
J. Comp. Phys. 2018.

[3b] Y Khoo, J Lu, L Ying
Solving parametric PDE problems with artificial neural networks
arXiv:1707.03351,

[4] L Zwald, O Bousquet, G Blanchard
Statistical properties of kernel principal component analysis...
International Conference on Computational Learning Theory, Springer, 2004.

References

[5] D Yarotsky
Error bounds for approximations with deep ReLU networks
Neural Networks 2017.

[6] G Kutyniok, P Petersen M Raslan and R Schneider
A theoretical analysis of deep neural networks and parametric PDEs
arXiv:1904.00377

[7] C Schwab, J Zech
Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ
Analysis and Applications 2019

[7] A Chkifa, A Cohen, R DeVore and C Schwab
Sparse adaptive Taylor approximation algorithms for parametric and stochastic elliptic PDEs
ESAIM: Mathematical Modeling and Numerical Analysis 2013

[8] RA DeVore
The Theoretical Foundation of Reduced Basis Methods
Model Reduction and Approximation, SIAM, 2014

Example: Poisson

Elliptic PDE

$$
\begin{aligned}
-\Delta u = f, &\quad z \in D = (0,1)^2 \\
u = 0, &\quad z \in \partial D.
\end{aligned}
$$

Operator Of Interest

Linear (Forcing) $\Psi^\dagger : f \in L^2(D) \mapsto u \in H_0^1(D)$

Lemma

Let $f = \sum_{j=1}^\infty \xi_j \phi_j$ and suppose $(\|\phi_j\|_{L^\infty})_{j\geq 1} \in \ell^p$ for some $p \in (0,1)$. Then

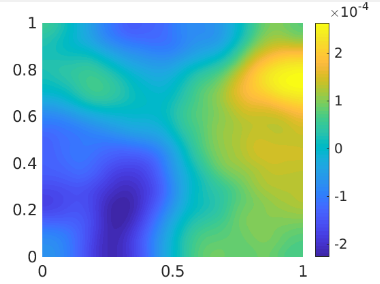
$$
\lim_{K\to\infty} \sup_{\xi} \|\Psi^\dagger(\xi) - \sum_{j=1}^{K} \xi_j \eta_j\|_{H_0^1} = 0
$$

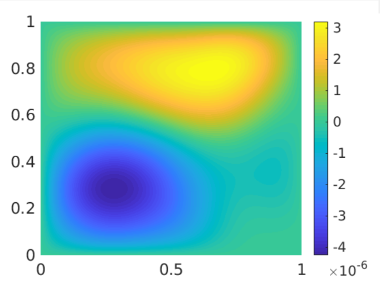
by viewing $\Psi^\dagger : \ell^\infty \to H_0^1$, where $-\Delta \eta_j = \phi_j$, $\eta_j|_{\partial D} = 0$ for each $j \in \mathbb{N}$.

Example: Poisson

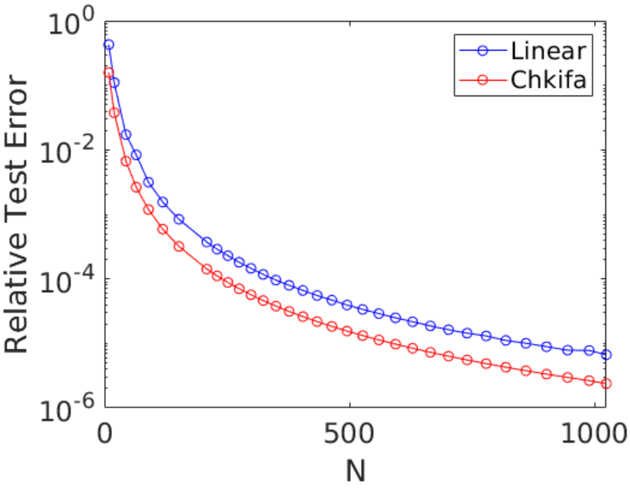
Input-Output

$$\text{Input:} \quad f \in L^2(D) \quad \text{(Left)},$$
$$\text{Output:} \quad u \in H_0^1(D). \quad \text{(Right)},$$





Example: Poisson



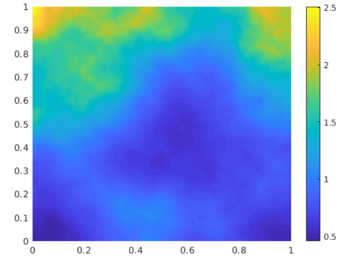
[8] A Chkifa, A Cohen, R DeVore and C Schwab

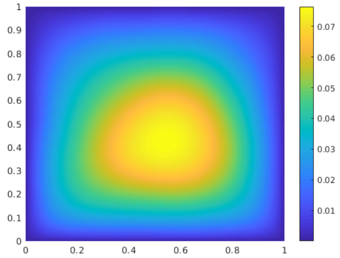
Example: Darcy (log-normal)

Input-Output

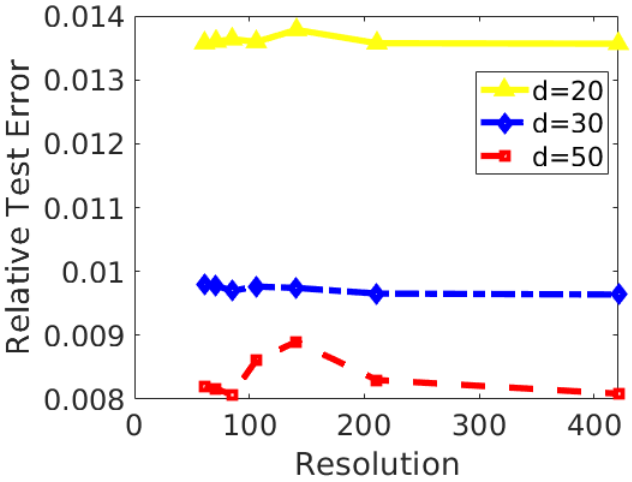
$$\text{Input:} \quad a \in L^2(D) \quad \text{(Left)},$$

$$\text{Output:} \quad u \in H_0^1(D). \quad \text{(Right)},$$

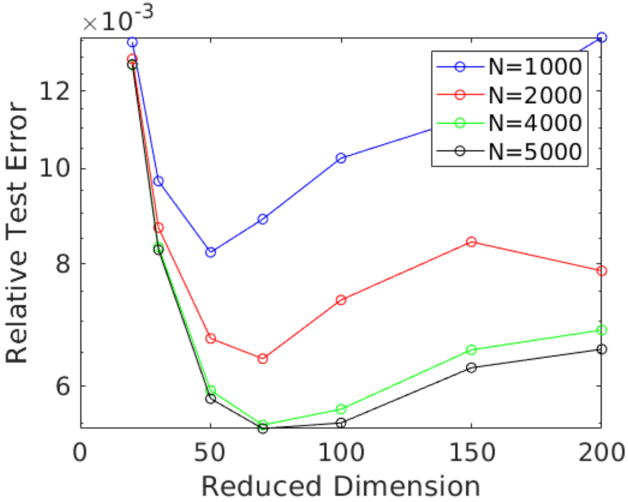




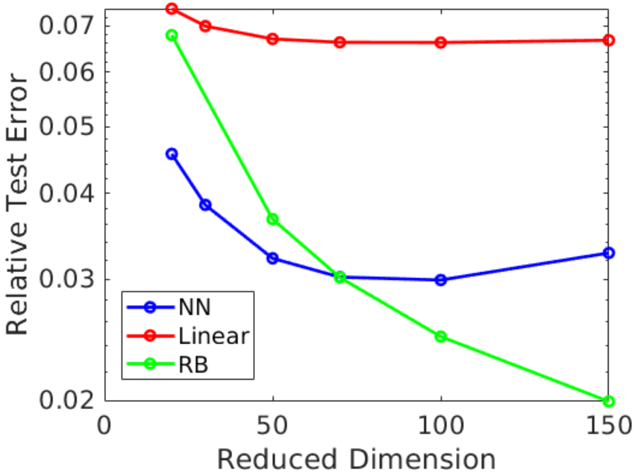
Example: Darcy (log-normal)



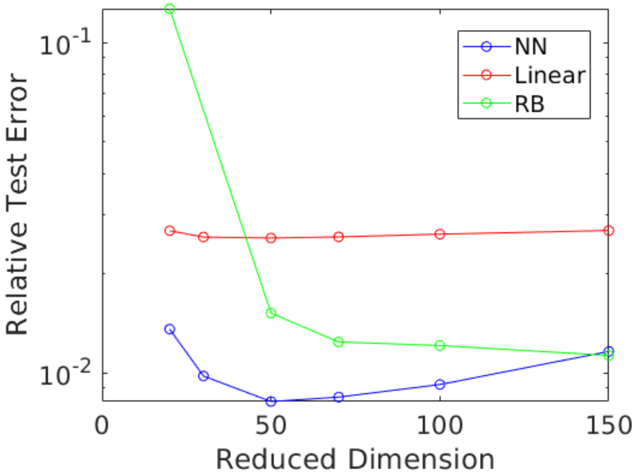
Example: Darcy (log-normal)



Example: Darcy (piecewise-constant)



Example: Darcy (log-normal)



Example: Darcy (mesh transfer)

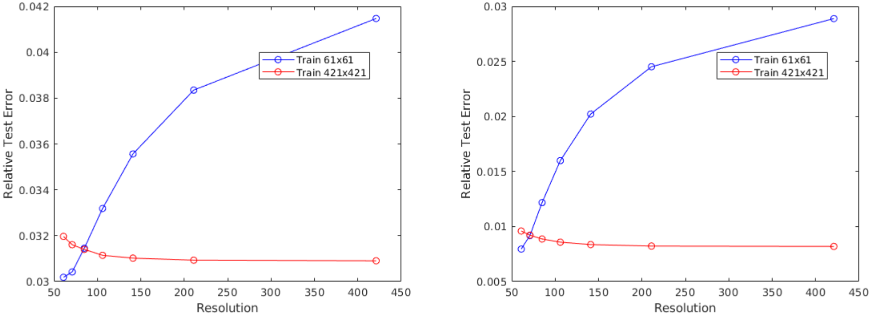
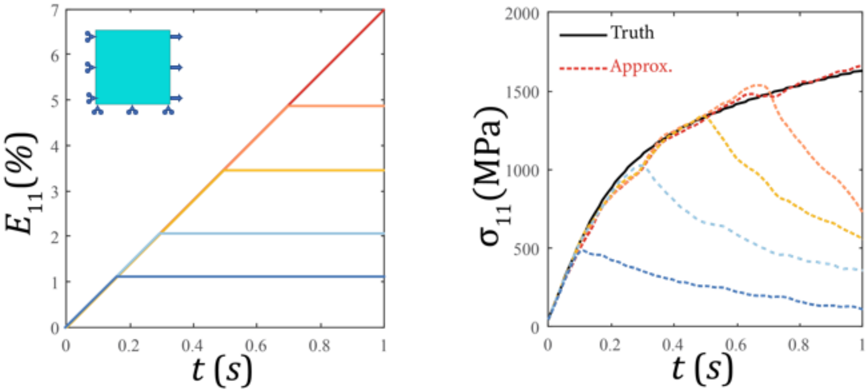
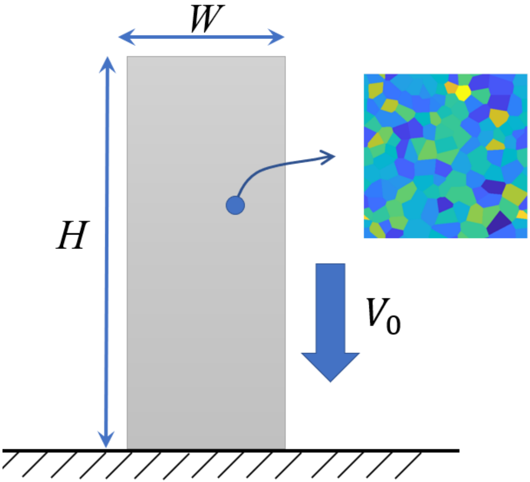


Figure: (Left) Piecewise-constant. (Right) Log-normal.

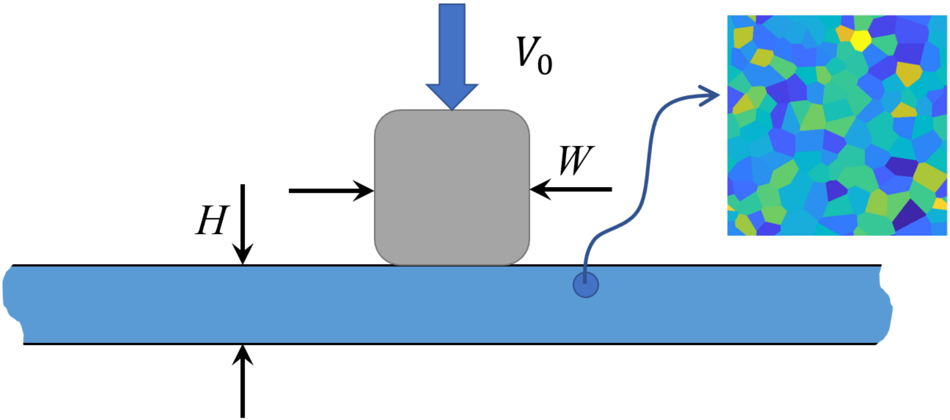
Causality



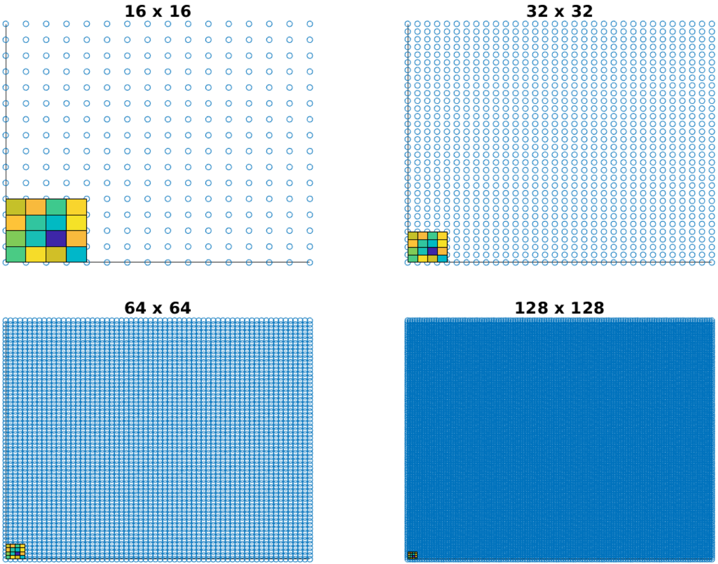
Taylor Anvil Test



Blunt Impact



Conv Nets are Parametrized for Grids



Example: Poisson

Elliptic PDE

$$
\begin{aligned}
-\Delta u &= f, \quad z \in D = (0,1) \\
u &= 0, \quad z \in \partial D.
\end{aligned}
$$

Operator Of Interest

Linear (Forcing) $\Psi^\dagger : f \in L^2(D) \mapsto u \in H_0^1(D)$

Architecture

$$
u(s) = \int_D \kappa(s, z; \theta) f(z)\, dz
$$

Example: Poisson