Learning Solution Operators For PDEs

⋆ ⋆ ⋆

Algorithms, Analysis and Applications

Andrew Stuart

Computing and Mathematical Sciences
California Institute of Technology

Various Locations in Paris

⋆ ⋆ ⋆

December 14th and 15th, 2023

Table of Contents

Talk Outline

Set-Up

Algorithms

Analysis

Application I: Material-Dependent Constitutive Models

Application II: History-Dependent Constitutive Models

Conclusions

Operator Learning

Supervised Learning

Determine $\Psi^\dagger : \mathcal{U} \to \mathcal{V}$ from samples

$$\{u_n, \Psi^\dagger(u_n)\}_{n=1}^N, \quad u_n \sim \mu.$$

Probability measure μ supported on $\mathcal{U}$.

In standard supervised learning $\mathcal{U} = \mathbb{R}^{d_x}$ and $\mathcal{V} = \mathbb{R}^{d_y}$ (regression) or $\mathcal{V} = \{1, \cdots K\}$ (classification).

Supervised Learning Of Operators

Separable Banach spaces $\mathcal{U}, \mathcal{V}$ of vector-valued functions:

$$\mathcal{U} = \{u : D_u \to \mathbb{R}^{d_i}\}, \quad D_u \subseteq \mathbb{R}^{d_u}$$
$$\mathcal{V} = \{v : D_v \to \mathbb{R}^{d_o}\}, \quad D_v \subseteq \mathbb{R}^{d_v}.$$

Operator Learning

Training

Consider a family of parameterized functions from $\mathcal{U}$ into $\mathcal{V}$:

$$\Psi : \mathcal{U} \times \Theta \mapsto \mathcal{V}.$$

Here $\Theta \subseteq \mathbb{R}^p$ denotes the parameter space.

$$\mu_N = \frac{1}{N}\sum_{n=1}^{N} \delta_{u_n}, \quad \mathrm{RE}(v, w) = \frac{\|v - w\|_{\mathcal{V}}}{\max\{1, \|v\|_{\mathcal{V}}\}},$$

$$\mathcal{R}_N(\theta) = \mathbb{E}^{u\sim\mu_N}\,\mathrm{RE}\big(\Psi^\dagger(u), \Psi(u;\theta)\big),$$

$$\theta^* = \mathrm{argmin}_{\theta\in\Theta}\ \mathcal{R}_N(\theta),$$

Motivation

Uses of Supervised Learning in Banach Space

- Surrogate Modeling;
- Model Discovery.

Example (Porous Medium Flow)

Darcy Law

Mass conservation	$-\nabla \cdot (a\nabla v) = f, \quad x \in D$
Boundary condition	$v = 0, \quad x \in \partial D$

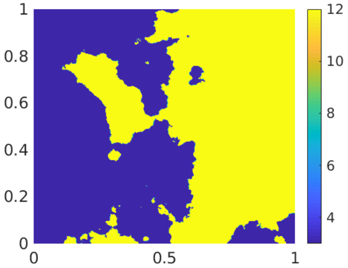
Operator Of Interest

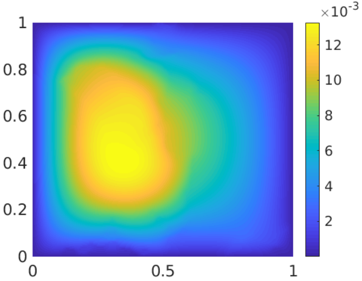
Parametric Dependence $\quad \Psi^{\dagger} : a \mapsto v$

Example (Porous Medium Flow)

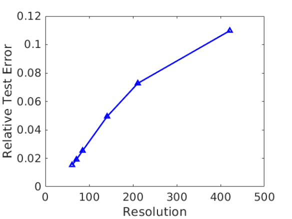
Input-Output

$$\begin{aligned} \text{Input:} \quad & a \in L^{\infty}(D) \quad \textit{(Left)}, \\ \text{Output:} \quad & v \in H_0^1(D). \quad \textit{(Right)}, \end{aligned}$$



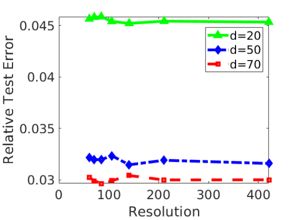


Example (Porous Medium Flow): Discretize Then Learn



Zhu and Zabaras 2018 [23], Khoo et al [10]

Example (Porous Medium Flow): Learn Then Discretize



Bhattacharya et al 2021 [2]

Examples: (Homogenized Constitutive Models)

Material Properties Dependence

- $\mathcal{U}$: material properties $A(\cdot)$.
- $\mathcal{V}$: stress σ.
- $\sigma = \Psi^\dagger(A)\big(\nabla u + \nabla u^\top\big)$.
- Approximate $\Psi^\dagger \approx \Psi(\cdot;\theta^\star)$

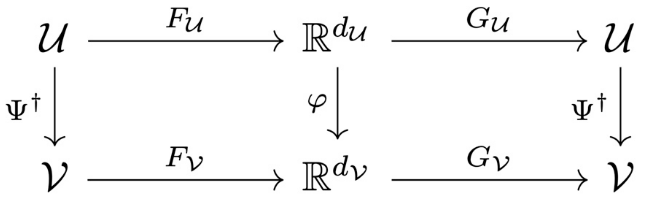
History Dependence

- $\mathcal{U}$: time histories of strain $\{\nabla u\}$.
- $\mathcal{V}$: time histories of stress $\{\sigma\}$.
- $\{\sigma\} = \Psi^\dagger(\{\nabla u\})$.
- Approximate $\Psi^\dagger \approx \Psi(\cdot;\theta^\star)$

Talk Outline

Finding Latent Structure

In A Picture



$$G_{\mathcal{U}} \circ F_{\mathcal{U}} \approx Id$$
$$G_{\mathcal{V}} \circ F_{\mathcal{V}} \approx Id$$
$$G_{\mathcal{V}} \circ \varphi \circ F_{\mathcal{U}} \approx \Psi^{\dagger}$$

PCA-NET

Architecture Bhattacharya, Hosseini, Kovachki and AMS '19 [2]

$$\Psi_{PCA}(u;\theta) = \sum_{j=1}^{m} \alpha_j(Lu;\theta)\psi_j, \quad \forall u \in \mathcal{U} \qquad x \in D_v.$$

Details

- $\{\phi_j\}$ are PCA basis functions under μ.
- $Lu = \{\langle \phi_j, u \rangle\}$ maps to PCA coefficients under μ.
- $\{\psi_j\}$ are PCA basis functions under $(\Psi^\dagger)^\sharp\mu$.
- $\{\alpha_j\}$ are finite dimensional neural networks.

DEEPONET Shallow approximation of functionals (Chen and Chen [6]) made deep

Architecture Lu, Jin, Pang, Zhang, and Karniadakis'19 [20]

$$\Psi_{DEEP}(u;\theta) = \sum_{j=1}^{m} \alpha_j(Lu;\theta_\alpha)\psi_j(\bullet;\theta_\psi), \quad \forall u \in \mathcal{U} \qquad x \in D_v.$$

Details

- Lu maps to linear functionals on $\mathcal{U}$.
- e.g. PCA coefficients under μ; or pointwise $\{u(x_\ell)\}$.
- $\{\alpha_j, \psi_j\}$ are finite dimensional neural networks.
- $\theta = (\theta_\alpha, \theta_\psi)$.

Fourier Neural Operator (FNO) DNN (Goodfellow et al [8]) Extended to Operators

Architecture Li, Kovachki, Azizzadenesheli, Liu, Bhattacharya, AMS and Anandkumar et al '20 [18, 13]

$\mathcal{U} = \mathcal{V}$ Hilbert

$$\Psi_{FNO}(u;\theta) = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \mathcal{L}_2 \circ \mathcal{L}_1 \circ \mathcal{R}(u),\ \forall u \in \mathcal{U},$$

$$\mathcal{L}_l(v)(x;\theta) = \sigma\big(W_l v(x) + b_l + \mathcal{K}(v)(x;\gamma_l)\big),$$

$$\mathcal{K}(v)(x;\gamma) = \sum_{m=1}^{M} \alpha_m\big(\gamma(m)\big)\langle \mathrm{f}_m, v\rangle_{\mathcal{U}^{\otimes d_c}}\mathrm{g}_m(x).$$

Details

- $\mathcal{Q}, \mathcal{R}$ pointwise NNs or linear transformations.
- (W_l, b_l) define pointwise affine transformations.
- $\mathcal{K}$ eg FFT as convolutional integral operator.
- θ collects parameters from previous three bullets.

Talk Outline

Universal Approximation (Latent) over compact sets

Definition

A Banach space has the **approximation property (AP)** if every compact operator is a limit of finite-rank operators.

Theorem 1 Kovacvhki '22 [12, 13]

Assume

- $\mathcal{U}$, $\mathcal{V}$ Banach spaces with the AP.
- $\Psi^\dagger : \mathcal{U} \to \mathcal{V}$ continuous, $K \subset \mathcal{U}$ compact.

For any $\epsilon > 0$ $\exists$ bounded linear $F_\mathcal{U} : \mathcal{U} \to \mathbb{R}^{d_\mathcal{U}}$, $G_\mathcal{V} : \mathbb{R}^{d_\mathcal{V}} \to \mathcal{V}$, and a continuous map $\varphi \in C(\mathbb{R}^{d_\mathcal{U}}; \mathbb{R}^{d_\mathcal{V}})$ such that

$$\sup_{u \in K} \|\Psi^\dagger(u) - (G_\mathcal{V} \circ \varphi \circ F_\mathcal{U})(u)\|_\mathcal{V} \leq \epsilon.$$

Universal Approximation (Latent) Bochner integration

Theorem 2 Kovacvhki '22 [12, 13]

Assume

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

For any $\epsilon > 0$ $\exists$ bounded linear $F_\mathcal{U} : \mathcal{U} \to \mathbb{R}^{d_\mathcal{U}}$, $G_\mathcal{V} : \mathbb{R}^{d_\mathcal{V}} \to \mathcal{V}$, and a continuous map $\varphi \in C(\mathbb{R}^{d_\mathcal{U}};\mathbb{R}^{d_\mathcal{V}})$ such that

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

Complexity of Approximation (Latent)

Lanthaler, Mishra and Karniadakis '21 [16] (DeepONet and complexity, NSE)

Lanthaler '23 [14] (PCA-Net and complexity, Darcy and NSE)

Marcati and Schwab '23 [21] (Analyticity of coefficients/solution, Darcy)

Herrmann, Schwab and Zech '23 [9] (Operator holomorphy, Darcy)

Lanthaler and AMS '23 [17] (Complexity estimates, Hamilton-Jacobi)

Universal Approximation (FNO)

Theorem 3 Lanthaler, Li and AMS '23 [15]

Assume

- $\mathcal{U} = C^s(\overline{D}, \mathbb{R}^d), \mathcal{V} = C^{s'}(\overline{D}, \mathbb{R}^{d'})$.
- $\mathcal{U} = W^{s,p}(\overline{D}, \mathbb{R}^d), \mathcal{V} = W^{s',p'}(\overline{D}, \mathbb{R}^{d'})$.
- $\Psi^\dagger : \mathcal{U} \to \mathcal{V}$ continuous, $K \subset \mathcal{U}$ compact.

For any $L, M > 0$ and any $\epsilon > 0$ $\exists$ an FNO $\Psi(\cdot; \theta^\star) : \mathcal{U} \to \mathcal{V}$ such that

$$\sup_{u \in K} \|\Psi^\dagger(u) - \Psi(u; \theta)\|_{\mathcal{V}} \leq \epsilon.$$

Kovachki, Lanthaler and Mishra '21 [11] (Complexity, NSE)

Talk Outline

Multiscale Problem

Canonical Elliptic Multiscale Problem

$$
\begin{aligned}
-\nabla \cdot (A^{\epsilon} \nabla u^{\epsilon}) &= f, \quad x \in \Omega \\
u^{\epsilon} &= 0, \quad x \in \partial\Omega \\
A^{\epsilon}(x) &= A\left(\frac{x}{\epsilon}\right), \quad A \in \mathcal{U} := L^{\infty}(\mathbb{T}^d, \mathbb{R}^{d\times d}).
\end{aligned}
$$

Standing Asssumption on A

$$
\begin{aligned}
PD_{\alpha,\beta} =& \{A \in L^{\infty}(\mathbb{T}^d; \mathbb{R}^{d\times d}) : \\
& \forall (y,\xi) \in \mathbb{T}^d \times \mathbb{R}^d,\ \alpha|\xi|^2 \leq \langle \xi, A(y)\xi \rangle \leq \beta|\xi|^2\}.
\end{aligned}
$$

Operator Learning

Homogenized Elliptic Problem; $u_0 \approx u^\epsilon$

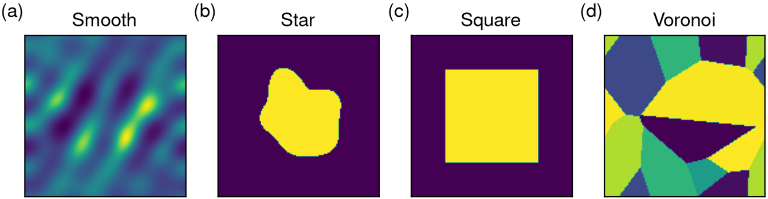
$$
\begin{aligned}
-\nabla \cdot (A_0 \nabla u_0) &= f, \quad x \in \Omega \\
u_0 &= 0, \quad x \in \partial\Omega \\
A_0(x) &= A_0, \quad \text{constant.}
\end{aligned}
$$

Constitutive Model Bensoussan, Lions, Papanicolaou '78 [1], Pavliotis and AMS '08 [22][Ch12]

A_0 determined by $\chi \in \mathcal{V} := H^1_{\mathrm{per}}(\mathbb{T}^d, \mathbb{R}^d)$

$$
\begin{aligned}
-\nabla_y \cdot (\nabla_y \chi A) &= \nabla_y \cdot A, \quad y \in \mathbb{T}^d, \\
A_0 &= \int_{\mathbb{T}^d} \left(A(y) + A(y) \nabla \chi(y)^T \right) \, dy.
\end{aligned}
$$

Varying Microstructures



$$K \subset \mathrm{BV}(\mathbb{T}^d; \mathbb{R}^{d\times d}) \cap PD_{\alpha,\beta} \Subset L^2(\mathbb{T}^d; \mathbb{R}^{d\times d}).$$

Universal Approximation (Cell Problem Solution Operator)

Goal: Supervised Learning (FNO)

Bhattacharya, Kovachki, Rajan, AMS, Trautner '23 [3]

- Learn map $A(\cdot) \in \mathcal{U} \mapsto \chi(\cdot) \in \mathcal{V} = H^1_{\mathrm{per}}(\mathbb{T}^d, \mathbb{R}^d)$.
- How to choose $\mathcal{U}$?

Theorem 4 Bhattacharya, Kovachki, Rajan, AMS, Trautner '23 [3]

Define the mapping $\Psi^\dagger : PD_{\alpha,\beta} \to \dot{H}^1(\mathbb{T}^d; \mathbb{R}^d)$ from the solution map $A \mapsto \chi$ given by

$$-\nabla_y \cdot (\nabla_y \chi A) = \nabla_y \cdot A, \quad y \in \mathbb{T}^d.$$

Then, for any $\epsilon > 0$ and $K \subset PD_{\alpha,\beta}$ compact in $L^2(\mathbb{T}^d; \mathbb{R}^{d\times d})$, there exists an FNO $\Psi(\cdot; \theta^*) : K \to \dot{H}^1(\mathbb{T}^d; \mathbb{R}^d)$ such that

$$\sup_{A\in K} \|\Psi^d(A) - \Psi(A; \theta^\star)\|_{\dot{H}^1} < \epsilon.$$

Stability Estimates

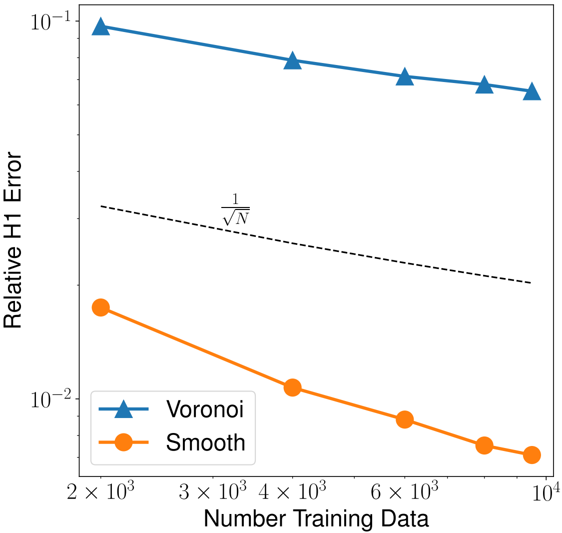
Lemma $A \mapsto \chi$

Assume that $U = PD_{\alpha,\beta} \bigcap \mathcal{U}$.

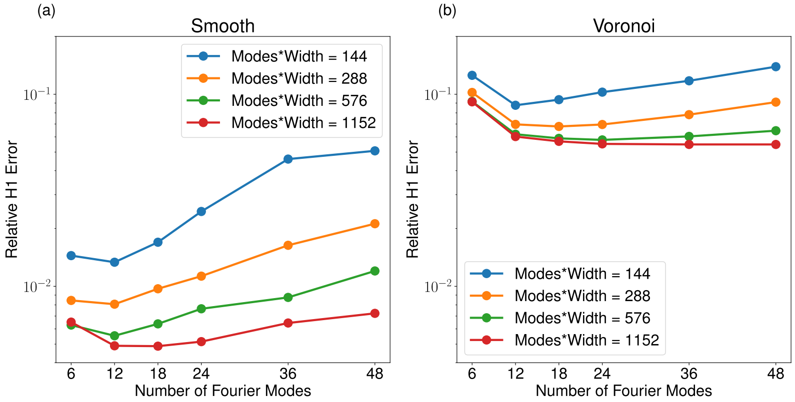
- For $\mathcal{U} = L^2(\mathbb{T}^d; \mathbb{R}^{d \times d})$ the map $U \to \dot{H}^1(\mathbb{T}^d; \mathbb{R}^d)$ is continuous.
- For $\mathcal{U} = L^q(\mathbb{T}^d; \mathbb{R}^{d \times d})$ the map $U \to \dot{H}^1(\mathbb{T}^d; \mathbb{R}^d)$ is Lipschitz, $q \in (2, q')$, for some $q' \in (2, \infty)$.

Bonito, De Vore, Nochetto 2013 [5]

Test Error versus Data Size



Test Error versus Model Size



Talk Outline

Big Picture

Multiscale Problem Pavliotis and AMS '08 [22][Ch12]

Displacement $u^\epsilon(x,t)$, stress $\sigma^\epsilon(x,t)$, $0 < \epsilon \ll 1$. F=MA:

$$\rho\,\partial_t^2 u^\epsilon = \nabla\cdot(\sigma^\epsilon) + f, \quad \sigma^\epsilon = \Psi^\epsilon\left(\{\nabla u^\epsilon\}, \frac{x}{\epsilon}\right).$$

Homogenized Problem Bensoussan, Lions, Papanicolaou [1]

Approximate $u^\epsilon = u_0 + \epsilon u_1 + \epsilon^2 u_2 + \dots$

Determine map Ψ, so that small scales are removed in u_0. F=MA:

$$\rho\,\partial_t^2 u_0 = \nabla\cdot(\sigma) + f, \quad \sigma = \Psi(\{\nabla u_0\}).$$

Operator Learning

- $\mathcal{U}$ histories of strain $\{\nabla u_0\}$; $\mathcal{V}$ histories of stress $\{\sigma\}$.
- Approximate $\Psi \approx \Psi_{NN}$

Viscoelasticity I Francfort and Suquet '86 [7]

Quasi-Static Viscoelasticity Multiscale Problem

$$-\nabla \cdot (\sigma^\epsilon) = f, \quad x \in \Omega$$
$$\sigma^\epsilon = \nu^\epsilon \partial_t \nabla u^\epsilon + E^\epsilon \nabla u^\epsilon$$
$$E^\epsilon(x) = E\left(\frac{x}{\epsilon}\right), \quad \nu^\epsilon(x) = \nu\left(\frac{x}{\epsilon}\right), \quad E, \nu : \mathbb{T}^d \to \mathbb{R}.$$

Laplace Transform, Homogenize, Invert

$$-\nabla \cdot (\overline{\sigma}^\epsilon) = f, \quad x \in \Omega$$
$$\overline{\sigma}^\epsilon = (s\nu^\epsilon + E^\epsilon)\nabla \overline{u}^\epsilon.$$

- Introduces memory.

Viscoelasticity II Bhattacharya, Liu, AMS, Trautner '22 [4]

Theorem (Piecewise-Constant Homogenization: Memory)

In piecewise-constant case and in dimension $d = 1$ homogenized equation for u_0 is Markovian:

$$
\begin{aligned}
-\nabla \cdot (\sigma) &= f, \quad x \in \Omega, \\
\sigma &= \nu' \partial_t \nabla u_0 + E' \nabla u_0 + \langle \mathbb{1}, r \rangle \\
\partial_t r_\ell &= -\alpha_\ell r_\ell + \beta_\ell \nabla u_0, \quad \ell \in \{1, 2, \cdots, L\},
\end{aligned}
$$

for some choice of $E' \in \mathbb{R}_+$, $\nu' \in \mathbb{R}_+$, $\alpha \in \mathbb{R}^L_+$, $\beta \in \mathbb{R}^L$, $L \in \mathbb{Z}_+$.

- Homogenization introduces memory.
- In $d = 1$ (approximate) Markovian structure.
- Dimension $d > 1$?

Viscoelasticity III: Operator Learning

True Solution Map

Let $\Psi : \mathcal{U} \to \mathcal{V}$ be the map such that the homogenized constitutive relation is

$$\sigma = \Psi(\{\nabla u_0\}).$$

Goal: Supervised Learning (RNO-NET)

Learn map $\Psi_{RNO} : \mathcal{U} \to \mathcal{V}$ approximating Ψ with the form

$$\sigma = F\left(\nabla u_0, \partial_t \nabla u_0, r\right)$$
$$\partial_t r = G(r, \nabla u_0), \quad r(0) = 0.$$

- RNO – Recurrent neural operator.
- Dimension of memory variable r has to be learned.

Viscoelasticity IV: Learning the Constitutive Map

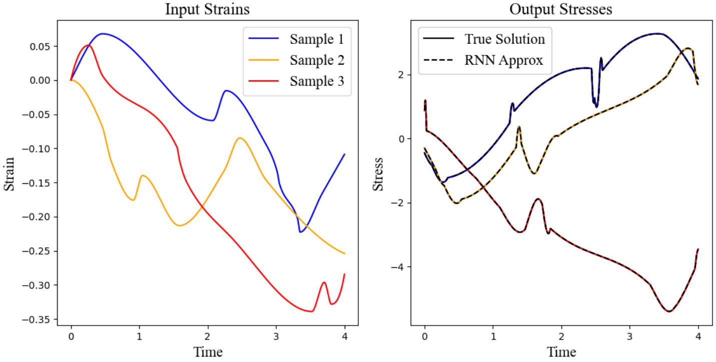


Figure: Viscoelasticity: trained model performs well on test samples

Viscoelasticity V: Choosing the Number of Hidden Variables

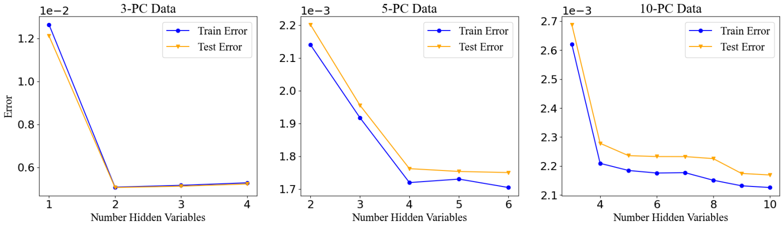


Figure: Absolute L^2 error of RNNs trained with different numbers of hidden variables on different piecewise-constant viscoelastic materials.

Viscoelasticity VI: Time Discretization Invariance

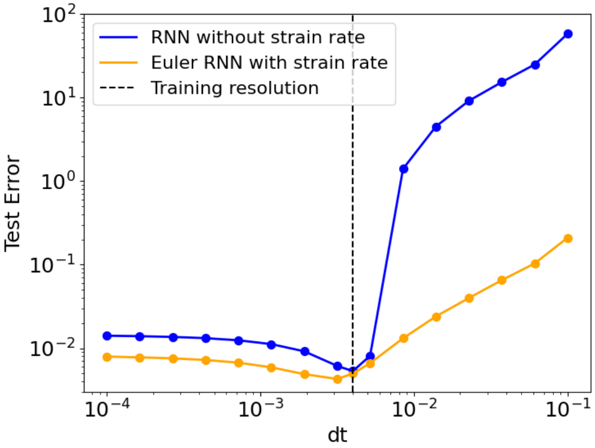


Figure: Viscoelasticity models trained with and without access to the strain rate variable: prefered model exhibits more invariance to time discretization of test trajectories

Talk Outline

Set-Up

Algorithms

Analysis

Application I: Material-Dependent Constitutive Models

Application II: History-Dependent Constitutive Models

Conclusions

Conclusions

1. Algorithms:
   - define on function space;
   - then learn;
   - leads to models which transfer between discretizations.
2. Analysis:
   - universal approximation theory well-developed;
   - complexity (cost verus error) incompletely understood;
   - what solution is found via optimization?
3. Applications:
   - cheap surrogates;
   - scientific discovery;
   - constitutive laws.

References I

[1] A. Bensoussan, J.-L. Lions, and G. Papanicolaou.
Asymptotic Analysis for Periodic Structures, volume 374.
American Mathematical Soc., 2011.

[2] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart.
Model reduction and neural networks for parametric pdes.
The SMAI journal of computational mathematics, 7:121–157, 2021.

[3] K. Bhattacharya, N. Kovachki, A. Rajan, A. M. Stuart, and M. Trautner.
Learning homogenization for elliptic operators.
arXiv:2306.12006, 2023.

[4] K. Bhattacharya, B. Liu, A. M. Stuart, and M. Trautner.
Learning Markovian homogenized models in viscoelasticity.
Multiscale Modeling & Simulation, 21:641–679, 2023.
arXiv:2205.14139.

[5] A. Bonito, R. A. DeVore, and R. H. Nochetto.
Adaptive finite element methods for elliptic problems with discontinuous coefficients.
SIAM Journal on Numerical Analysis, 51(6):3106–3134, 2013.

[6] T. Chen and H. Chen.
Approximations of continuous functionals by neural networks with application to dynamic systems.
IEEE Transactions on Neural networks, 4(6):910–918, 1993.

[7] G. A. Francfort and P. M. Suquet.
Homogenization and mechanical dissipation in thermoviscoelasticity.
Archive for Rational Mechanics and Analysis, 96(3):265–293, 1986.

[8] I. Goodfellow, Y. Bengio, and A. Courville.
Deep learning.
MIT press, 2016.

References II

[9] L. Herrmann, C. Schwab, and J. Zech.
Neural and gpc operator surrogates: construction and expression rate bounds.
arXiv preprint arXiv:2207.04950, 2022.

[10] Y. Khoo, J. Lu, and L. Ying.
Solving parametric pde problems with artificial neural networks.
European Journal of Applied Mathematics, 32(3):421–435, 2021.

[11] N. Kovachki, S. Lanthaler, and S. Mishra.
On universal approximation and error bounds for Fourier neural operators.
The Journal of Machine Learning Research, 22(1):13237–13312, 2021.

[12] N. B. Kovachki.
Machine Learning and Scientific Computing.
PhD thesis, California Institute of Technology, 2022.

[13] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, and A. Anandkumar.
Neural operator: Learning maps between function spaces with applications to pdes.
J. Mach. Learn. Res., arXiv:2108.08481, 24(89):1–97, 2023.

[14] S. Lanthaler.
Operator learning with pca-net: upper and lower complexity bounds.
arXiv preprint arXiv:2303.16317, 2023.

[15] S. Lanthaler, Z. Li, and A. M. Stuart.
The nonlocal neural operator: Universal approximation.
arXiv:2304.13221, 2023.

[16] S. Lanthaler, S. Mishra, and G. E. Karniadakis.
Error estimates for deeponets: a deep learning framework in infinite dimensions.
Transactions of Mathematics and Its Applications, 6(1):tnac001, 2022.

References III

[17] S. Lanthaler and A. M. Stuart.
The curse of dimensionality in operator learning.
arXiv preprint arXiv:2306.15924, 2023.

[18] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar.
Fourier neural operator for parametric partial differential equations.
ICLR, arXiv:2010.08895, 2021.

[19] B. Liu, N. Kovachki, Z. Li, K. Azizzadenesheli, A. Anandkumar, A. M. Stuart, and K. Bhattacharya.
A learning-based multiscale method and its application to inelastic impact problems.
Journal of the Mechanics and Physics of Solids, 158:104668, 2022.

[20] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis.
Learning nonlinear operators via deeponet based on the universal approximation theorem of operators.
Nature Machine Intelligence, 3(3):218–229, 2021.

[21] C. Marcati and C. Schwab.
Exponential convergence of deep operator networks for elliptic partial differential equations.
SIAM Journal on Numerical Analysis, 61(3):1513–1545, 2023.

[22] G. Pavliotis and A. M. Stuart.
Multiscale Methods: Averaging and Homogenization.
Springer Science & Business Media, 2008.

[23] Y. Zhu and N. Zabaras.
Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification.
Journal of Computational Physics, 366:415–447, 2018.

Plasticity I Liu, Kovachki, Li, Azizzadenesheli, Anandkumar, AMS, Bhattacharya '22 [19]

Plasticity Multiscale Problem

$$
\begin{aligned}
\rho\,\partial_t^2 u^\epsilon &= \nabla \cdot \sigma^\epsilon + f, \qquad x \in \Omega \\
\partial_t \xi^\epsilon &= K(\xi^\epsilon, \nabla u^\epsilon), \qquad x \in \Omega \\
\sigma^\epsilon &= \Psi^\epsilon\left(\nabla u^\epsilon, \xi^\epsilon, \frac{x}{\epsilon}\right)
\end{aligned}
$$

Plasticity II

Homogenized Plasticity Problem

$$\rho\,\partial_t^2 u_0 = \nabla \cdot \sigma_0 + f, \qquad x \in \Omega$$
$$\sigma_0 = \Psi\Big(\{\nabla u_0\}\Big)$$

Plasticity III: Operator Learning

Goal: Supervised Learning (PCA-NET)

Learn map $\Psi_{PCA} : \mathcal{U} \to \mathcal{V}$ approximating Ψ.
In particular causality must be learned.

Goal: Supervised Learning (RNO-NET)

Learn map $\Psi_{RNO} : \mathcal{U} \to \mathcal{V}$ approximating Ψ with the form

$$\sigma = F\left(\nabla u_0, \partial_t \nabla u_0, r\right)$$
$$\partial_t r = G(r, \nabla u_0), \quad r(0) = 0.$$

Plasticity IV: Learning the Constitutive Map

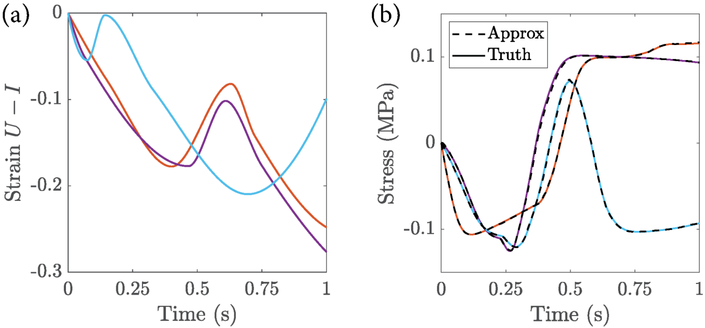


Figure: Viscoplasticity: Trained model performs well on test samples. Left: Input strains. Right: Output truth and approximation.

Plasticity V:
Choosing the Number of Hidden Variables

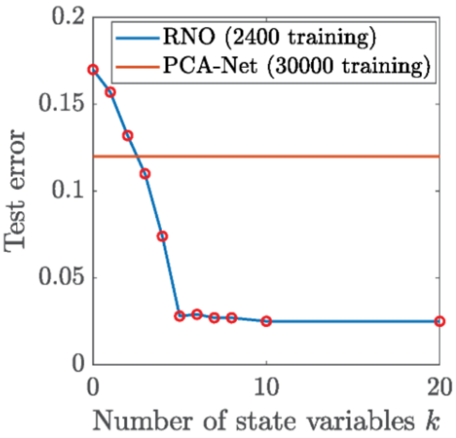


Figure: Viscoplasticity: 3D polycrystal (different hidden variable counts)

Plasticity VI: Time Discretization Invariance

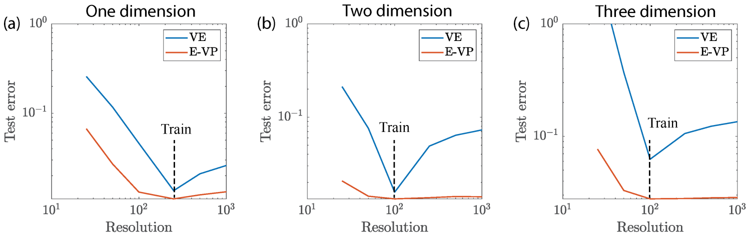
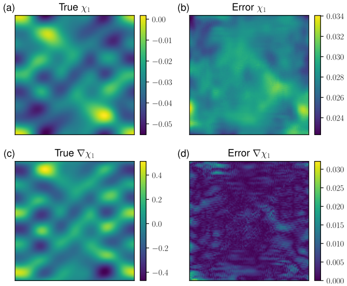


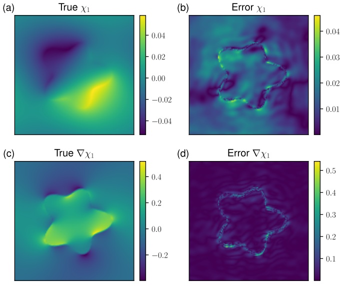
Figure: Models with viscoplastic (VP) and elasto-viscoplastic (E-VP) architecture trained on data from an E-VP material: prefered model exhibits more time discretization invariance

Learning Error for Varying Microstructures (I)

Smooth Microstructure

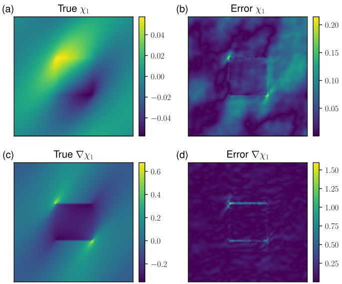


Star Inclusion Microstructure

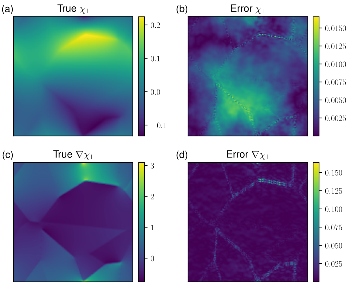


Learning Error for Varying Microstructures (II)

Square Inclusion Microstructure



Voronoi Microstructure



Learning Error for Voronoi Microstructure