Scaling Up

Supervised Learning in Function Space

Andrew Stuart

California Institute of Technology

AFOSR, ARO, DoD, NSF, ONR

Slides: `http://stuart.caltech.edu/talks/index.html`

Collaborators

Covered In This Talk

- w/Cotter, Roberts, White [4] (MCMC for functions)
- w/Hairer, Vollmer [8] (Spectral gaps for MCMC for functions)
- w/Bhattacharya, Hosseini, Kovachki [1] (PCA-Net)
- w/Nelsen [17] (RFM: Random features – using FFT)
- w/Li, Kovachki, Azizzadenesheli, Liu, Bhattacharya, Anandkumar [14, 11] (FNO)
- Kovachki [12] (Machine learning and scientific computing)
- w/De Hoop, Huang, Qian [5] (Cost-Accuracy trade-off)

Adjacent To This Talk

- w/De Hoop, Kovachki, Nelsen [6] (Learn linear operators)
- w/Bhattacharya, Liu, Trautner [2] (RNO)

Overview

Talk Outline

The Idea

MCMC For Functions

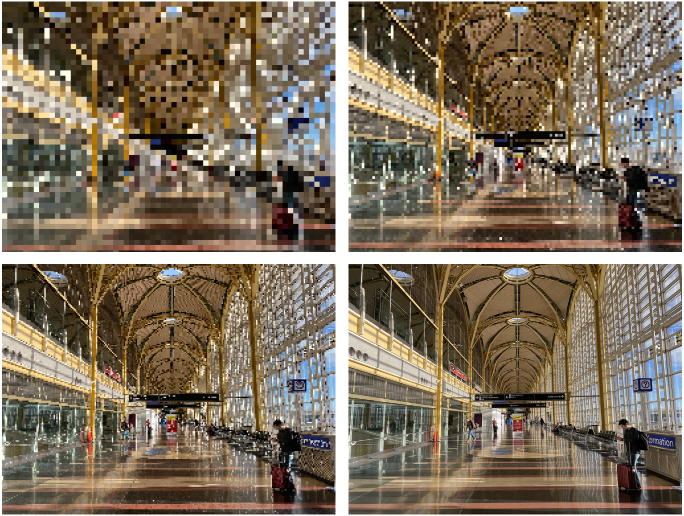
Supervised Learning For Functions

Neural Operators

Numerical Results

Closing

Pixellated Images Versus Functions



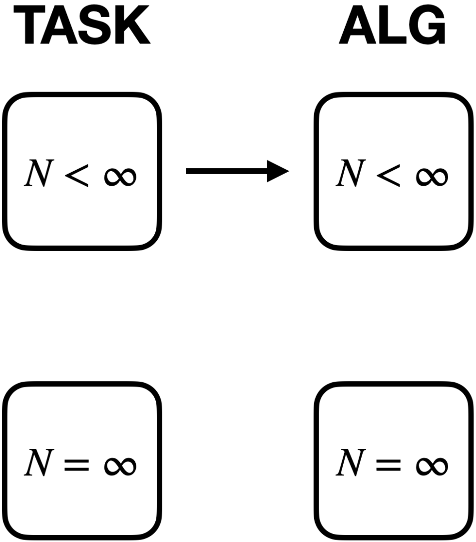
Thanks to Dima Burov

Finite Dimensional Vectors Versus Functions

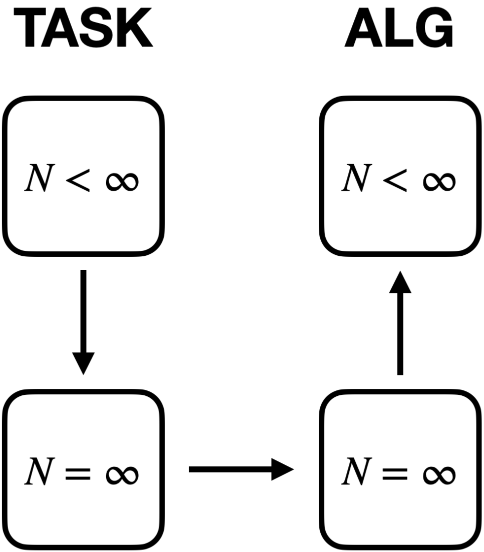
$\mathbb{R}^{3N}$ $N = 64^2$	$\mathbb{R}^{3N}$ $N = 128^2$
$\mathbb{R}^{3N}$ $N = 256^2$	$u : \mathscr{D} \to \mathbb{R}^3$

Thanks to Edo Calvello

Don't



Do



Talk Outline

The Problem

Target Measure

Consider probability measures π, π_0 supported on $\mathcal{U}$ and $\Phi : \mathcal{U} \to \mathbb{R}^+$:

$$\pi_0 = \mathcal{N}(0, C),$$
$$\pi(u) \propto \exp\big(-\Phi(u)\big)\pi_0(u).$$

Goal: to draw approximate samples from π on vector space $\mathbb{R}^N$ approximating $\mathcal{U}$.

MCMC

Construct Markov chain with kernel p so that

$$u_{n+1} \sim p(u_n, \cdot),$$
$$\text{Law}(u_n) \to \pi, \quad \text{as} \quad n \to \infty.$$

Randon Walk Metropolis (RWM) Algorithm

Metropolis et al 1953 [16]

Applies to any π.

Proposal

For some $\delta \in (0, \infty)$:

$$u^{\star}_{n+1} = u_n + \sqrt{2\delta}\xi_n,$$
$$\xi_n \sim \mathcal{N}(0, C).$$

Accept-Reject

$$a(u, v) = \min\Big\{\frac{\pi(v)}{\pi(u)}, 1\Big\},$$

$$u_{n+1} = u^{\star}_{n+1}, \quad \text{with probability} \quad a(u_n, u^{\star}_{n+1}),$$
$$u_{n+1} = u_n, \quad \text{otherwise.}$$

The pCN Algorithm

Cotter et al 2013 [4]

Applies to $\pi \propto \exp(-\Phi(u))\mathcal{N}(0, C)$.

Proposal

For some $\delta \in (0, \frac{1}{2}]$:

$$\begin{aligned} u_{n+1}^{\star} &= (1-2\delta)^{\frac{1}{2}} u_n + \sqrt{2\delta}\xi_n, \\ \xi_n &\sim \mathcal{N}(0, C). \end{aligned}$$

Accept-Reject

$$a(u, v) = \min\Big\{\exp\big(\Phi(u) - \Phi(v)\big), 1\Big\},$$

$$\begin{aligned} u_{n+1} &= u_{n+1}^{\star}, \quad \text{with probability } a(u_n, u_{n+1}^{\star}), \\ u_{n+1} &= u_n, \quad \text{otherwise.} \end{aligned}$$

Comparison of RWM (Don't) and pCN (Do)

Applies to $\pi \propto \exp(-\Phi(u))\mathcal{N}(0, C)$.

Theorem (RWM) Hairer et al '14 [8]

For optimal step-size choice $\delta = \Theta(N^{-\frac{1}{2}})$ spectral gap $\mathfrak{sg}$ satisfies

$$1 - \mathfrak{sg} \leq \mathcal{O}(N^{-\frac{1}{2}}).$$

Thus $\Omega(N^{\frac{1}{2}})$ steps are required to sample.

Theorem (pCN) Hairer et al '14 [8]

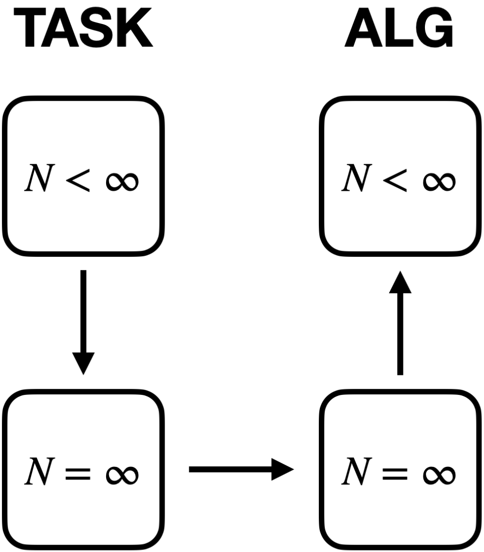
For all $\delta \in (0, \frac{1}{2}]$ spectral gap $\mathfrak{sg}$ satisfies

$$\mathfrak{sg} = \Theta(1).$$

Thus $\Theta(1)$ steps are required to sample.

Talk Outline

Do



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:

$$\mathcal{U} = \{u : D_x \to \mathbb{R}^{d_i}\}, \quad D_x \subseteq \mathbb{R}^{d_x}$$
$$\mathcal{V} = \{v : D_y \to \mathbb{R}^{d_o}\}, \quad D_y \subseteq \mathbb{R}^{d_y}.$$

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.

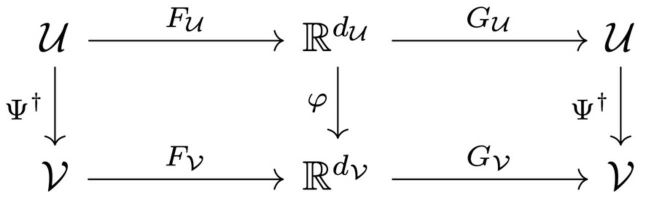
$$\theta^* = \text{argmin}_\theta \ \mathcal{R}_\infty(\theta), \quad \mathcal{R}_\infty(\theta) := \mathbb{E}^{u\sim\mu} \|\Psi^\dagger(u) - \Psi(u;\theta)\|_{\mathcal{V}}^2.$$

Testing

$$\text{error} = \mathbb{E}^{u\sim\mu} \left(\frac{\|\Psi^\dagger(u) - \Psi(u;\theta^\star)\|_{\mathcal{V}}}{\|\Psi^\dagger(u)\|_{\mathcal{V}}} \right).$$

Finding Latent Structure

In A Picture



Example (Fluid Flow in a Porous Medium)

Darcy Law

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

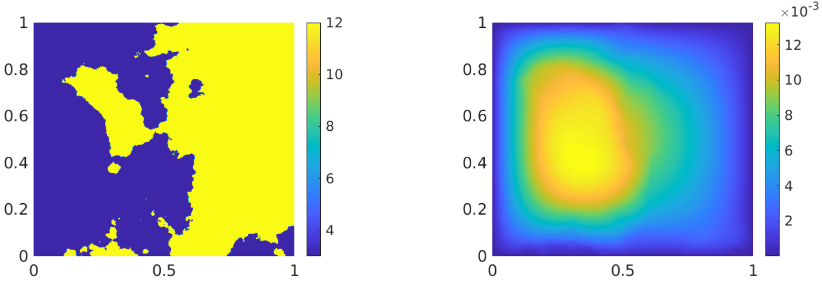
Operator Of Interest

Parametric Dependence $\Psi^\dagger : a \mapsto v$

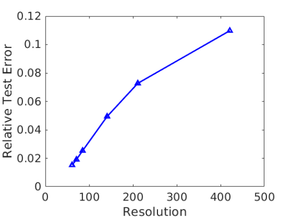
Example (Fluid Flow in a Porous Medium)

Input-Output

$$\text{Input:} \quad a \in L^\infty(D) \quad (\mathit{Left}),$$
$$\text{Output:} \quad v \in H^1(D). \quad (\mathit{Right}),$$

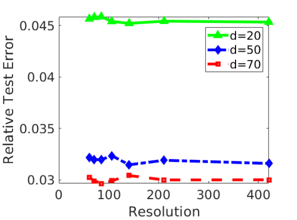


Example: Don't



Zhu and Zabaras 2018 [19]

Example: Do



Bhattacharya et al 2021 [1]

Theoretical Justification – PCA-NET

Theorem Bhattacharya, Hosseini, Kovachki and AMS '19 [1]

Let $\Psi^\dagger \in L^p_\mu(\mathcal{U};\mathcal{V})$. For any $\epsilon > 0$, there are latent dimensions, data volume and network size such that $\Psi_{PCA} = G_\mathcal{V} \circ \varphi \circ F_\mathcal{U}$ satisfies

$$\mathbb{E}^{\text{data}} \|\Psi^\dagger - \Psi_{PCA}\|_{L^p_\mu(\mathcal{U};\mathcal{V})} \leq \epsilon.$$

Talk Outline

RFM Random Features Method (Rahimi and Recht [18]) Extended to Operators

Architecture Nelsen and AMS '21 [17]

$$\Psi_{RFM}(u;\theta)(y) := \sum_{j=1}^{m} \theta_j \psi(u;\gamma_j)(y) \quad \forall u \in \mathcal{U}\ y \in D_y; \quad \gamma_j \, \text{i.i.d.}\,.$$

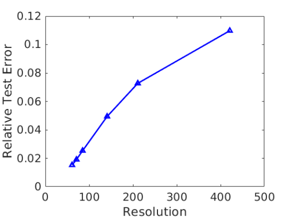
Fourier Space Random Features

- $\mathcal{F}$ denotes Fourier transform.
- γ a Gaussian random field.
- χ Fourier space reshuffle.
- σ an activation function.
- $\psi(u;\gamma) = \sigma\big(\mathcal{F}^{-1}(\chi \mathcal{F}\gamma \mathcal{F} u)\big)$.

Practical Matters

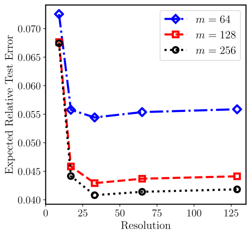
- Quadratic optimization for θ.
- Monte Carlo approximation of GP/Kernel methods.

Example: Don't



Zhu and Zabaras 2018 [19]

Example: Do



Nelsen and S 2021 [17]

FNO DNN (Goodfellow et al [7]) Extended to Operators

Architecture Li, Kovachki, Azizzadenesheli, Liu, Bhattacharya, AMS and Anandkumar et al '20 [14, 11]

$$\Psi_{FNO}(u;\theta) = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \mathcal{L}_2 \circ \mathcal{L}_1 \circ \mathcal{R}(u), \ \forall u \in \mathcal{U},$$
$$\mathcal{L}_l(v)(x;\theta) = \sigma\big(W_l v(x) + b_l + \mathcal{K}(v)(x;\gamma_l)\big).$$

Details

- $\mathcal{Q}, \mathcal{R}$ pointwise NNs or linear transformations.
- (W_l, b_l) define pointwise affine transformations.
- $\mathcal{K}$ convolutional integral operator (FFT on k_{max} modes, parameterized by γ).
- θ collects parameters from previous three bullets.
- Nonlinear Approximation.

Kovachki, Lanthaler and Mishra '21 [10] (beating curse of dimensionality, FNO),

Lanthaler, Mishra and Karniadakis '21 [13] (beating curse of dimensionality, DeepONet),

Kovachki '22 [12] (general neural operator framework)

Universal Approximation

Theorem Kovacvhki '22 [12, 11]

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

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

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

Theorem Kovacvhki '22 [12, 11]

- $\mathcal{U}$ 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 $1 \leq p < \infty$.

Then

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

Lanthaler, Mishra and Karniadakis '21 [13] (DeepONet)

Talk Outline

Bayesian Inverse Problem

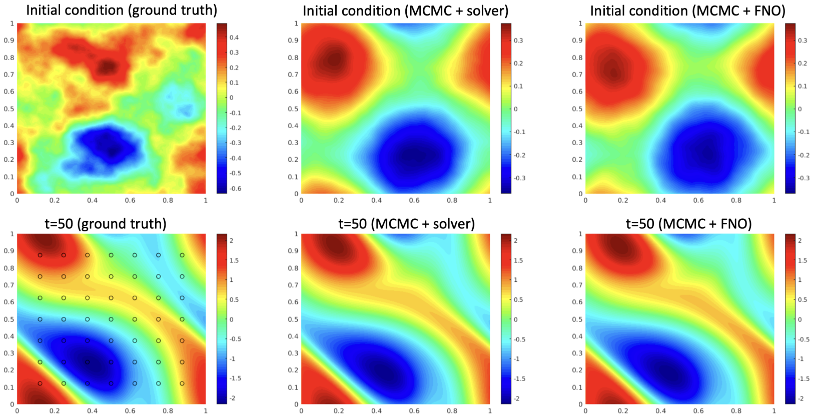


Figure: Posterior mean: using MCMC with Ψ.

For this problem, state-of-the-art MCMC requires 3×10^4 evaluations of the forward operator $\Psi^\dagger/\Psi$. This takes 12 hours with the pseudo-spectral solver; under 2 minutes using FNO.

Test Error vs. Network Size

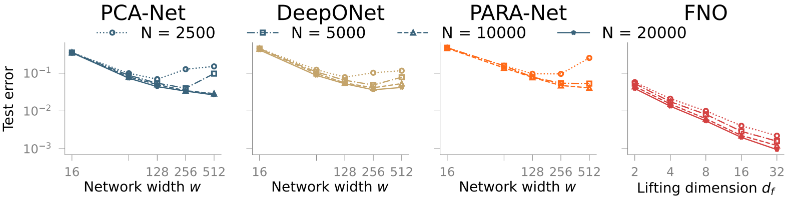


Figure: Test error vs. network size.

Test Error vs. Cost

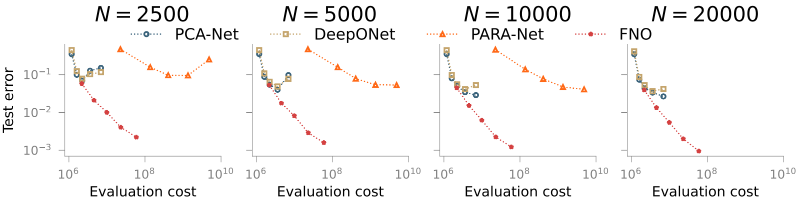


Figure: Test error vs. cost.

Test Error vs. Training Data

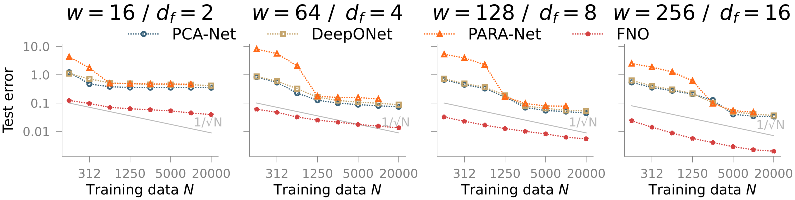


Figure: Test error vs. training data amount N.

Talk Outline

Closing

Conclusions

- Conceptualize in the $N = \infty$ limit:
  - task;
  - algorithm.
- Has led to new MCMC for sampling.
- Has led to new neural networks for operator learning.
- Comparison with standard numerical methods is lacking.
- More approximation theory needed; interaction between:
  - Data volume;
  - Richness/design of parameterization;
  - Finite dimensional discretization;
  - Optimization.
- Other $N = \infty$ limits are important to understand:
  - Autoencoders;
  - Triangular maps;
  - Normalizing flows;
  - Score-based transport;
  - $\cdots$.

Recruiting Postdocs

astuart@caltech.edu

References I

[1] 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.

[2] K. Bhattacharya, B. Liu, A. Stuart, and M. Trautner.
Learning markovian homogenized models in viscoelasticity.
arXiv preprint arXiv:2205.14139, 2022.

[3] G. J. Chandler and R. R. Kerswell.
Invariant recurrent solutions embedded in a turbulent two-dimensional kolmogorov flow.
Journal of Fluid Mechanics, 722:554–595, 2013.

[4] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White.
Mcmc methods for functions: modifying old algorithms to make them faster.
Statistical Science, 28(3):424–446, 2013.

[5] M. De Hoop, D. Z. Huang, E. Qian, and A. M. Stuart.
The cost-accuracy trade-off in operator learning with neural networks.
arXiv preprint arXiv:2203.13181, 2022.

[6] M. V. de Hoop, N. B. Kovachki, N. H. Nelsen, and A. M. Stuart.
Convergence rates for learning linear operators from noisy data.
arXiv preprint arXiv:2108.12515, 2021.

References II

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

[8] M. Hairer, A. M. Stuart, and S. J. Vollmer.
Spectral gaps for a metropolis–hastings algorithm in infinite dimensions.
The Annals of Applied Probability, 24(6):2455–2490, 2014.

[9] B. T. Knapik, A. W. Van Der Vaart, and J. H. van Zanten.
Bayesian inverse problems with gaussian priors.
The Annals of Statistics, 39(5):2626–2657, 2011.

[10] N. Kovachki, S. Lanthaler, and S. Mishra.
On universal approximation and error bounds for fourier neural operators.
arXiv preprint arXiv:2107.07562, 2021.

[11] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar.
Neural operator: Learning maps between function spaces.
arXiv preprint arXiv:2108.08481, 2021.

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

References III

[13] S. Lanthaler, S. Mishra, and G. E. Karniadakis.
Error estimates for deeponets: A deep learning framework in infinite dimensions.
arXiv preprint arXiv:2102.09618, 2021.

[14] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar.
Fourier neural operator for parametric partial differential equations.
ICLR 2021; arXiv:2010.08895, 2020.

[15] 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.

[16] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller.
Equation of state calculations by fast computing machines.
The journal of chemical physics, 21(6):1087–1092, 1953.

[17] N. H. Nelsen and A. M. Stuart.
The random feature model for input-output maps between banach spaces.
SIAM Journal on Scientific Computing, 43(5):A3212–A3243, 2021.

[18] A. Rahimi, B. Recht, et al.
Random features for large-scale kernel machines.
In *NIPS*, volume 3, page 5. Citeseer, 2007.

References IV

[19] 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.

Linear Operators

Setting

$$\text{Input-Output Spaces:} \quad \mathcal{U} \subseteq \mathcal{V} = (H, \langle\cdot,\cdot\rangle, \|\cdot\|)$$

$$\text{Target Linear Operator:} \quad L^\dagger : \mathcal{D}(L^\dagger) \subseteq H \to H$$

$$\Pi(du, dv): \quad v = L^\dagger u + \eta, \quad u \perp\!\!\!\perp \eta$$

$$u \sim \mu = \mathcal{N}(0, \mathcal{C}_1), \quad \eta \sim \mathcal{N}(0, \gamma^2 \,\mathrm{Id}),$$

$$\text{Data:} \quad \{u_n, v_n\}_{n=1}^N \overset{\text{iid}}{\sim} \Pi, \quad N \in \mathbb{N}$$

Approach

Bayesian Formulation: posterior π *on* L *given* $\{u_n, v_n\}_{n=1}^N$

Linear Operators: Convergence Theory

Recall: $x_{jn} \sim \mathcal{N}(0, j^{-2\alpha})$ (data), $\ell_j \sim \mathcal{N}(0, j^{-2\beta})$ (prior), $\ell^\dagger \in \mathcal{H}^s$ (truth)

Theorem (Bayesian Consistency)

$$\mathbb{E}^{\{u_n, v_n\}}\mathbb{E}^{\pi}\|L - L^\dagger\|^2_{L^2_\mu(H;H)} = O\Big(N^{-\left(\frac{\alpha+\beta-1/2}{\alpha+\beta}\right)}\Big) + o\Big(N^{-\left(\frac{\alpha+s}{\alpha+\beta}\right)}\Big) \quad (N \to \infty)$$

Remarks

- Similar lower bounds, with matching rates, in some regimes.
- Similar results with high probability over $\{u_n\}_{n=1}^N \overset{\text{iid}}{\sim} \mu$
- Extensions to error in posterior mean.
- Extensions to test measures $\mu' \not\equiv \mu$.

Analysis builds on Knapik, Van Der Vaart and van Zanten '11 [9]

PCA-NET

Architecture Bhattacharya, Hosseini, Kovachki and AMS '19 [1]

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

Details

- $\{\phi_j\}$ are PCA basis functions under μ.
- $Lu = \{\langle \phi_j, u \rangle\}_j$ 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

Architecture Lu, Jin, Pang, Zhang, and Karniadakis'19 [15]

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

Details

- Lu maps to PCA coefficients under μ.
- Lu comprising pointwise observations $\{u(x_\ell)\}$ is also possible.
- $\{\alpha_j, \psi_j\}$ are finite dimensional neural networks.
- $\theta = (\theta_\alpha, \theta_\psi)$.

RNO (Recurrent Neural Operator) $D = (0, T)$

Architecture Bhattacharya, Liu, AMS, Trautner '22 [2]

$$\Psi_{RNO}(e;\theta)(t) = F\Big(e(t), \frac{de}{dt}(t), r(t); \theta\Big), \quad \forall e \in \mathcal{U} \qquad t \in [0, T],$$

$$\frac{dr}{dt} = G(r, e; \theta), \quad \forall e \in \mathcal{U} \qquad t \in (0, T], \quad r(0) = 0.$$

Details

- Finite dimensional neural networks F, G;
- Two-layer used in this talk.

2D Incompressible Navier Stokes Equation

Formulation Chandler and Kerswell [3]

$$\frac{\partial \omega}{\partial t} + (v \cdot \nabla)\omega - \nu \Delta \omega = f',$$
$$-\Delta \psi = \omega, \qquad \int \psi(x,t)dx = 0,$$
$$v = \Big(\frac{\partial \psi}{\partial x_2}, -\frac{\partial \psi}{\partial x_1}\Big).$$

Operator

Learn the map between $\omega|_{t=0}$ and $\omega|_{t=\tau}$

$$\Psi^\dagger : \omega|_{t=0} \to \omega|_{t=\tau}.$$

Choose $\omega|_{t=0} \sim \mu := \mathcal{N}\big(0, (-\Delta + \tau^2)^{-\delta}\big)$. $\tau = 3, \delta = 2$.

Forward Problem

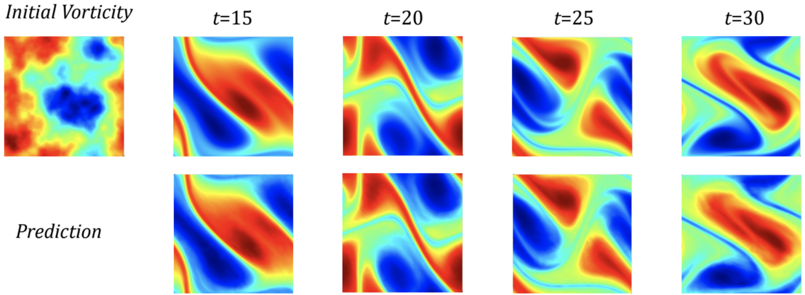


Figure: $\Psi^\dagger / \Psi : \omega|_{t=0} \to \omega|_{t=\tau}$.

The FNO prediction Ψ matches the true solution operator $\Psi^\dagger$.

2D Incompressible Navier Stokes Equation

Formulation Chandler and Kerswell [3]

$$\frac{\partial \omega}{\partial t} + (v \cdot \nabla)\omega - \nu \Delta \omega = f',$$

$$-\Delta \psi = \omega, \qquad \int \psi(x,t)dx = 0,$$

$$v = \Big(\frac{\partial \psi}{\partial x_2}, -\frac{\partial \psi}{\partial x_1}\Big).$$

Operator

Learn the map between forcing f' and the vorticity at time T:

$$\Psi^\dagger : f' \to \omega|_{t=T}.$$

Choose $f' \sim \mu := \mathcal{N}\big(0, (-\Delta + \tau^2)^{-\delta}\big)$. $\tau = 3, \delta = 4$.

Navier Stokes Equation

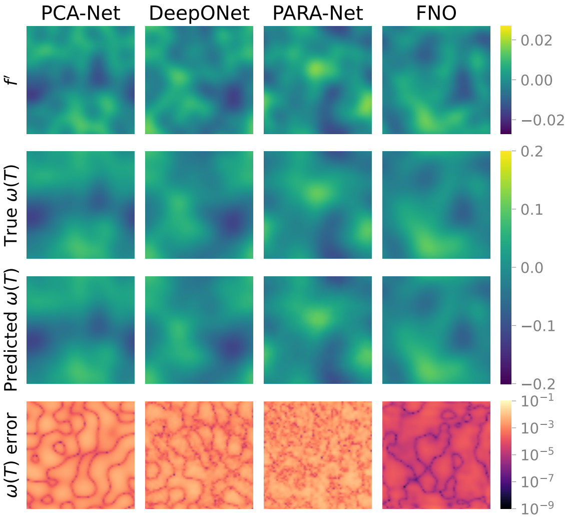


Figure: Learned model predictions for inputs resulting in **median** test errors for networks of size $w = 128$ / $d_f = 16$ trained on $N = 10000$ data.

Navier Stokes Equation

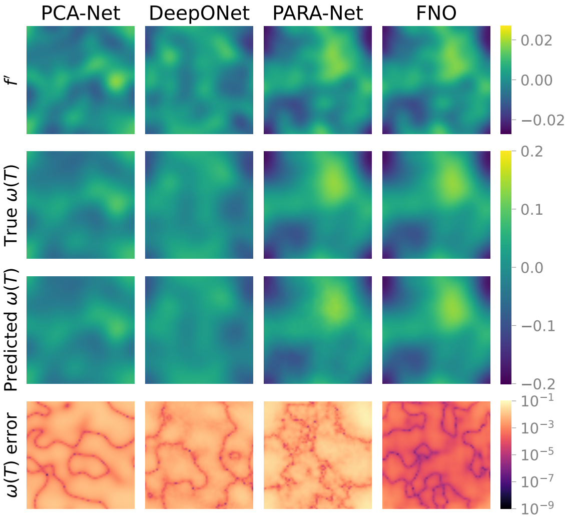


Figure: Learned model predictions for inputs resulting in **worst** test errors for networks of size $w = 128$ / $d_f = 16$ trained on $N = 10000$ data.

Navier-Stokes Equation Output Space

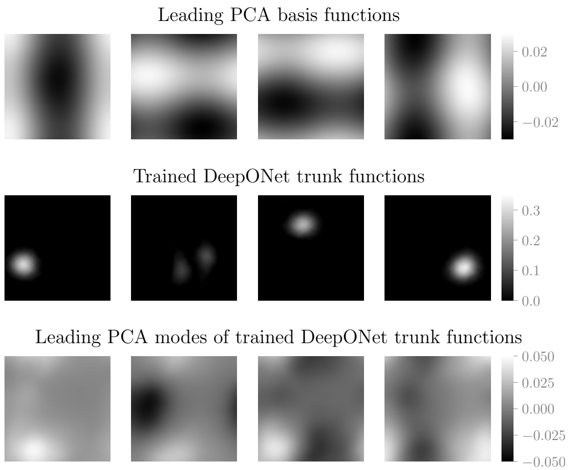


Figure: Comparison of output space bases: PCA-Net and DeepONet.

Advection Equation

Formulation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \qquad x \in [0,1),$$
$$u(0) = u_0$$

Operator

Learn the map between the initial condition u_0 and the solution at time 0.5, $u|_{t=0.5}$:

$$\Psi^\dagger : u_0 \to u|_{t=0.5}$$

where $u_0 = -1 + 2\mathbb{1}_{\{\widetilde{u_0} \geq 0\}}$ and $\widetilde{u_0} \sim \mathcal{N}(0, (-\Delta + \tau^2)^{-d})$.

Advection Equation

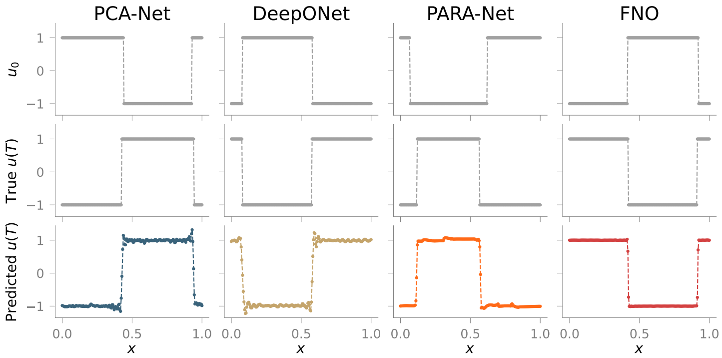


Figure: Learned solution predictions for inputs resulting in **median** test errors for networks of size $w = 128$ / $d_f = 16$ trained on $N = 10000$ data.

Advection Equation

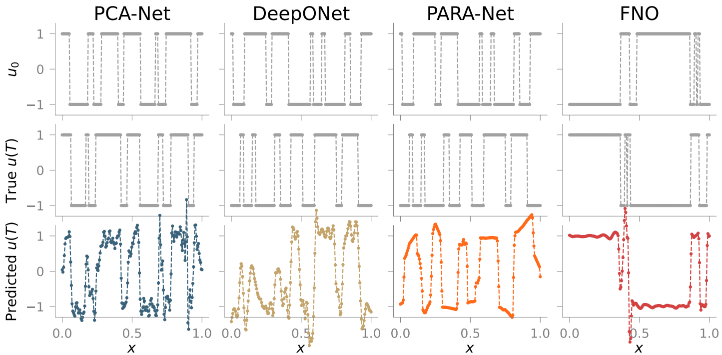


Figure: Learned solution predictions for inputs resulting in **worst** test errors for networks of size $w = 128$ / $d_f = 16$ trained on $N = 10000$ data.

Test Error vs. Cost

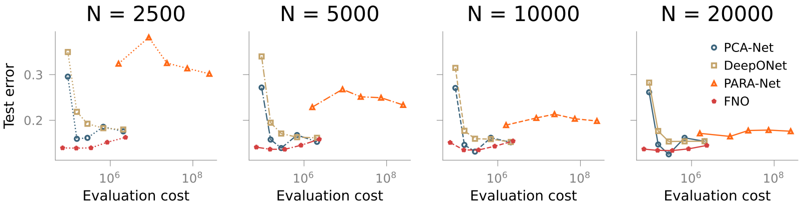


Figure: Test error vs. cost.

Advection Equation Output Space

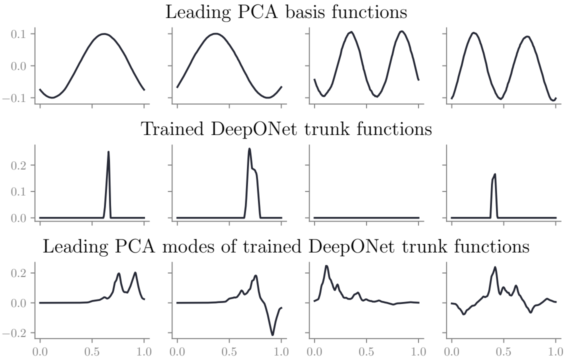


Figure: Comparison of output space bases: PCA-Net and DeepONet.