Operator Learning: Algorithms, Analysis and Applications

Supervised Learning in Function Space

Andrew Stuart

California Institute of Technology

AFOSR, ARO, DoD, NSF, ONR

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

May 21st 2024

Collaborators

Covered In This Talk

- w/Bhattacharya, Hosseini, Kovachki [1] (PCA-Net)
- w/Li, Kovachki, Azizzadenesheli, Liu, Bhattacharya, Anandkumar [19, 10] (FNO)
- w/Lanthaler, Li [14] (Universal Approximation)
- w/Lanthaler [17] (Complexity of Approximation)
- w/Lanthaler, Trautner [18] (Finite Dimensional Implementation)
- w/Lanthaler, Kovachki [11] (Review)
- Kovachki [12] (Machine Learning and Scientific Computing)

Adjacent To This Talk

- w/De Hoop, Huang, Qian [5] (Cost-Accuracy Trade-off)
- w/De Hoop, Kovachki, Nelsen [6] (Learn Linear Operators)
- w/Nelsen [21] (RFM: Random Features)
- w/Bhattacharya, Liu, Trautner [2] (RNO)
- Lanthaler, Nelsen [16] (Complexity of Random Features)
- Lanthaler [13] (Complexity of PCA)

Overview

Talk Outline

Algorithms on Function Space

Supervised Learning For Functions

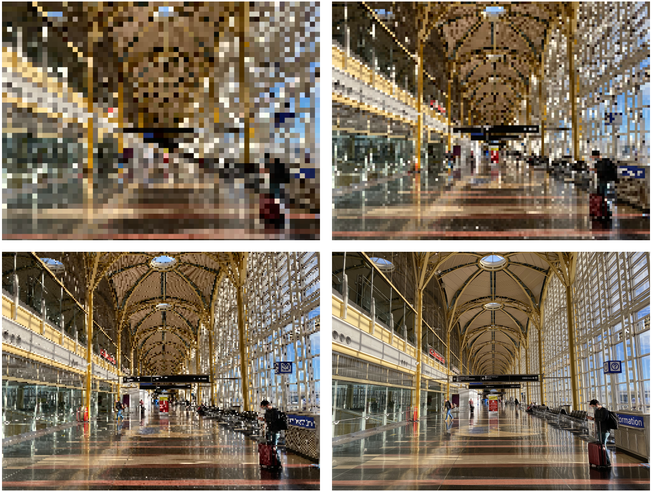
Neural Operators

Theory

Closing

Pixellated/Discretized Images Versus Functions

Thanks to Dima Burov



Ramsay and Silverman 2002 [23] (Statistics)

S 2010, Cotter et al 2013 [24, 4] (Bayesian Inverse Problems)

Finite Dimensional Vectors Versus Functions

Thanks to Edo Calvello

$$\mathbb{R}^{3N}$$

$$N = 64^2$$

$$\mathbb{R}^{3N}$$

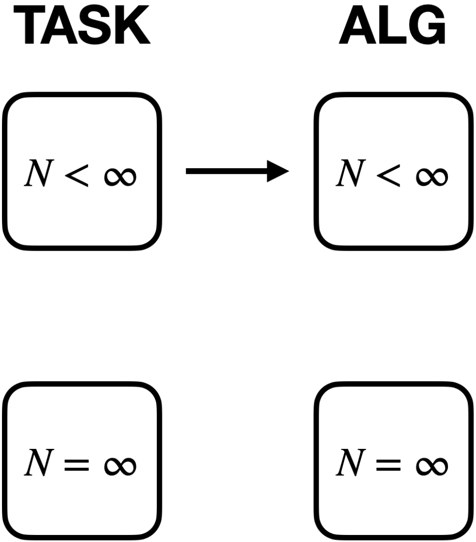
$$N = 128^2$$

$$\mathbb{R}^{3N}$$

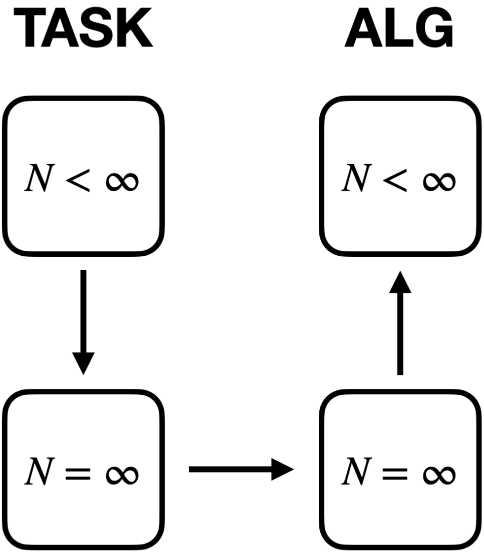
$$N = 256^2$$

$$u : \mathscr{D} \to \mathbb{R}^3$$

Don't



Do



Talk Outline

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:

$$\mathcal{U} = \{u : D \to \mathbb{R}\}, \quad D \subseteq \mathbb{R}^d$$
$$\mathcal{V} = \{v : D \to \mathbb{R}\}.$$

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.

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

Testing

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

Example (Fluid Flow in a Porous Medium)

Darcy Law

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

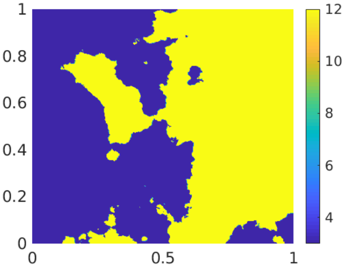
Operator Of Interest

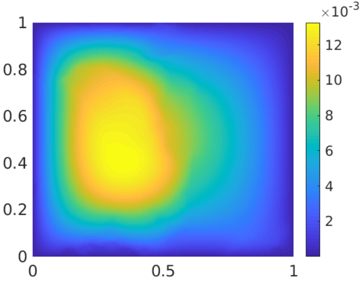
Parametric Dependence $\Psi^{\dagger} : a \mapsto v$

Example (Fluid Flow in a Porous Medium)

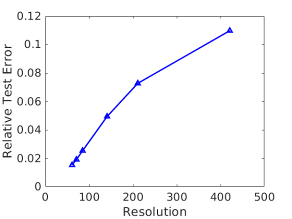
Input-Output

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



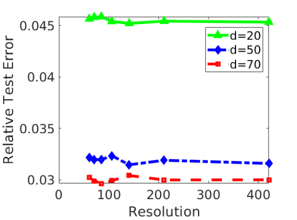


Example: Don't



Zhu and Zabaras 2018 [25]

Example: Do



Bhattacharya et al 2021 [1]

2D Incompressible Navier Stokes Equation

Formulation Chandler and Kerswell [3]

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

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

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

Operator

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

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

Choose $\omega|_{t=0} \sim \mu := \mathcal{N}\left(0, (-\Delta + \tau^2)^{-(\nu+1)}\right)$. $\tau = 3, \nu = 1$.

Forward Problem

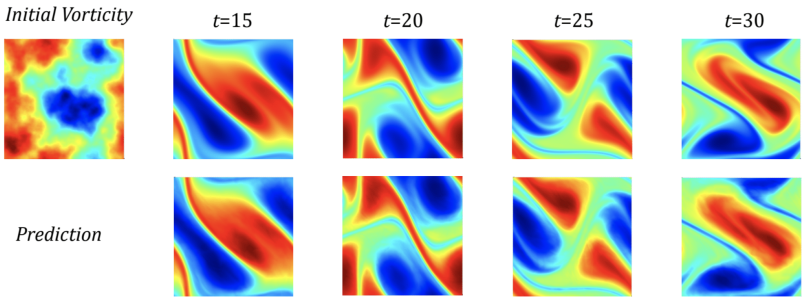


Figure: $\Psi^\dagger/\Psi : \omega|_{t=0} \to \omega|_{t=\tau}$.

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

Bayesian Inverse Problem

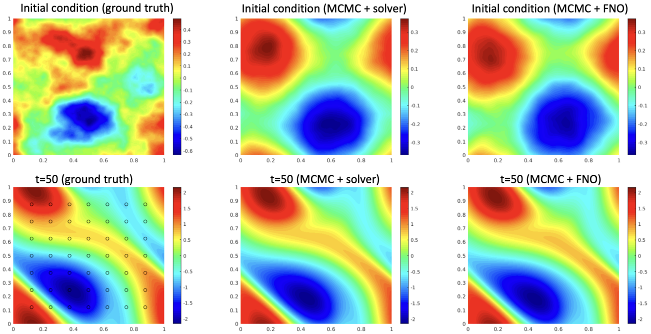


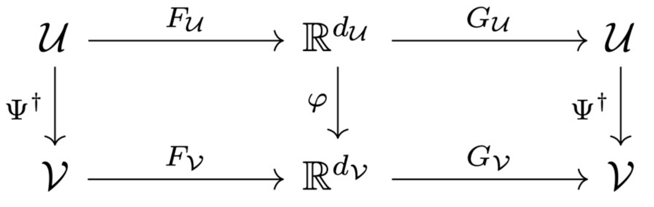
Figure: Posterior mean: using MCMC with Ψ.

State-of-the-art MCMC requires 3×10^4 evaluations of forward operator $\Psi^\dagger/\Psi$. Timings: 12 hours with the pseudo-spectral solver; under 2 minutes using FNO. (But cost of training $\cdots$).

Talk Outline

Finding Latent Structure

In A Picture



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 in input space $\mathcal{U}$.
- $Lu = \{\langle \phi_j, u \rangle\}_j$ maps to PCA coefficients.
- $\{\psi_j\}$ are PCA basis functions in output space $\mathcal{V}$.
- $\{\alpha_j\}$ are finite dimensional neural networks.

DEEPONET

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

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

Details

- Lu maps to PCA coefficients in input space $\mathcal{U}$.
- Lu comprising pointwise $\{u(x_\ell)\}$ is original version.
- $\{\psi_j\}$ are finite dimensional neural networks in output space $\mathcal{V}$.
- $\{\alpha_j\}$ are finite dimensional neural networks.
- $\theta = (\theta_\alpha, \theta_\psi)$.

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

Architecture Li, Kovachki, Azizzadenesheli, Liu, Bhattacharya, AMS and Anandkumar et al '20 [19, 10]

$$\Psi_{FNO}(u;\theta) = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \mathcal{L}_2 \circ \mathcal{L}_1 \circ \mathcal{R}(u),\ \forall u \in \mathcal{U},$$
$$\mathcal{L}_l(v)(x;\theta) = \sigma\big(W_l v(x) + b_l + \mathcal{K}(v)(x;\gamma_l)\big),$$
$$\mathcal{K}(v)(x;\gamma) = \sum_{m=0}^{M} \gamma^{(m)} \langle v, \varphi^{(m)} \rangle \varphi^{(m)}(x).$$

Details

- $\mathcal{R}$ NN pointwise lifts to $\mathcal{U}^c := \{u : D \to \mathbb{R}^{d_c}\}$.
- $\mathcal{Q}$ NN pointwise projects to $\mathcal{V} := \{u : D \to \mathbb{R}\}$.
- (W_l, b_l) define pointwise affine maps.
- $\mathcal{K}$ defines pointwise linear map in transform space.
- θ collects parameters from previous four bullets.

Talk Outline

Algorithms on Function Space

Supervised Learning For Functions

Neural Operators

Theory

Closing

Theory

Kovachki, Lanthaler, AMS '24 [11] (Review, Handbook of Numerical Analysis)

Universal Approximation

Theorem Lanthaler, Li, AMS '23 [14]

- $\mathcal{U} = C^s(D;\mathbb{R})$ and $\mathcal{V} = C^{s'}(D;\mathbb{R})$.
- $\Psi^\dagger : \mathcal{U} \to \mathcal{V}$ continuous, $K \subset \mathcal{U}$ compact.
- $\varphi^{(0)}(x) = 1$ for all $x \in D$.
- $L \geq 1, M \geq 0$.

For any $\epsilon > 0$ $\exists$ d_c sufficiently large and resulting FNO such that

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

Foundational Work:

Lanthaler, Mishra, Karniadakis '21 [15] (DEEPONET)

Kovachki, Lanthaler and Mishra '21 [9] (FNO)

Kovachki '22 [12] (General Theory)

Complexity of Approximation

Theorem Lanthaler, AMS '23 [17]

Assume that $\Psi^\dagger \in C^r(\mathcal{U}, \mathcal{V})$ and $K \subset \mathcal{U}$ compact. Then $\exists \Psi^\dagger$ and $b, c > 0$ such that approximation by Ψ_{FNO} to achieve

$$\sup_{u \in K} \|\Psi^\dagger(u) - \Psi_{FNO}(u)\|_{\mathcal{V}} \leq \epsilon,$$

has complexity which grows like $\exp(c\epsilon^{-b})$.

Exceptions:

Lanthaler, Mishra, Karniadakis '21 [15] (DEEPONET for Darcy, Hyperbolic Conservation Law)

Kovachki, Lanthaler, Mishra '21 [9] (FNO for Darcy, NSE)

Lanthaler '23 [13] (PCA for Darcy)

Lanthaler, AMS '23 [17] (HJ-Net for HJ)

Finite Dimensional Approximation

Theorem Lanthaler, AMS, Trautner '24 [18]

Assume that

- $u \in H^s, s > d/2$.
- $\sigma \in C_b^s$.

Then pseudo-spectral approximation Ψ_{FNO}^N of Ψ_{FNO} on N^d grid points satisfies

$$\|\Psi_{FNO}(u) - \Psi_{FNO}^N(u)\|_{\mathcal{V}} \leq CN^{-s}.$$

Talk Outline

Algorithms on Function Space

Supervised Learning For Functions

Neural Operators

Theory

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;
  - $\cdots$.

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.
Journal of Machine Learning, 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.
SIAM/ASA Journal on Uncertainty Quantification, arXiv:2108.12515, 11(2):480–513, 2023.

References II

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

[8] 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.

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

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

[11] N. B. Kovachki, S. Lanthaler, and A. M. Stuart.
Operator learning: Algorithms and analysis.
arXiv preprint arXiv:2402.15715, 2024.

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

References III

[13] S. Lanthaler.
Operator learning with pca-net: upper and lower complexity bounds.
Journal of Machine Learning Research, 24(318):1–67, 2023.

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

[15] 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.

[16] S. Lanthaler and N. H. Nelsen.
Error bounds for learning with vector-valued random features.
Advances in Neural Information Processing Systems, 36, 2024.

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

[18] S. Lanthaler, A. M. Stuart, and M. Trautner.
Discretization error of fourier neural operators.
arXiv preprint arXiv:2405.02221, 2024.

References IV

[19] 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.

[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] 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.

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

[23] B. Silverman and J. Ramsay.
Applied functional data analysis: methods and case studies.
2002.

[24] A. M. Stuart.
Inverse problems: a bayesian perspective.
Acta numerica, 19:451–559, 2010.

References V

[25] 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.

Talk Outline

Algorithms on Function Space

Supervised Learning For Functions

Neural Operators

Theory

Closing

Cost-Accuracy Trade-Off

Test Error vs. Network Size

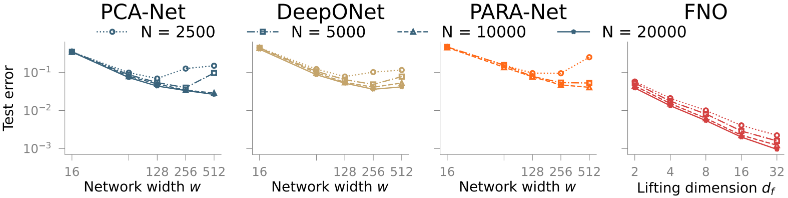


Figure: Test error vs. network size.

Test Error vs. Cost

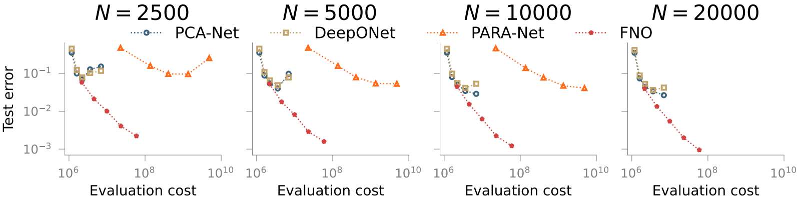


Figure: Test error vs. cost.

Test Error vs. Training Data

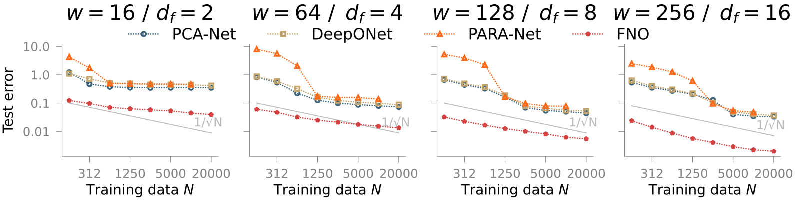


Figure: Test error vs. training data amount N.

Talk Outline

Algorithms on Function Space

Supervised Learning For Functions

Neural Operators

Theory

Closing

Learning Linear Operators

Linear Operators

Setting

$$
\begin{aligned}
\text{Input-Output Spaces:}\quad & \mathcal{U} \subseteq \mathcal{V} = (H, \langle\cdot,\cdot\rangle, \|\cdot\|) \\
\text{Target Linear Operator:}\quad & L^\dagger : \mathcal{D}(L^\dagger) \subseteq H \to H \\
\Pi(du, dv):\quad & v = L^\dagger u + \eta, \quad u \perp\!\!\!\perp \eta \\
& u \sim \mu = \mathcal{N}(0, \mathcal{C}_1), \quad \eta \sim \mathcal{N}(0, \gamma^2\,\mathsf{Id}), \\
\text{Data:}\quad & \{u_n, v_n\}_{n=1}^N \overset{\text{iid}}{\sim} \Pi, \quad N \in \mathbb{N}
\end{aligned}
$$

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 [8]

Talk Outline

Random Features Methods

RFM Random Features Method (Rahimi and Recht [22]) Extended to Operators

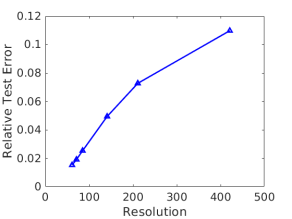
Architecture Nelsen and AMS '21 [21]

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

Fourier Space Random Features

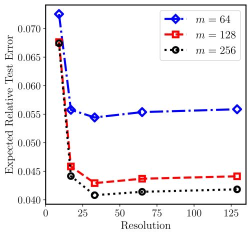
- $\mathcal{F}$ denotes Fourier transform.
- γ a Gaussian random field.
- χ Fourier space reshuffle.
- σ an activation function.
- $\psi(u;\gamma) = \sigma\big(\mathcal{F}^{-1}(\chi \mathcal{F}\gamma \mathcal{F}u)\big)$.

Example: Don't



Zhu and Zabaras 2018 [25]

Example: Do



Nelsen and S 2021 [21]

Talk Outline

Algorithms on Function Space

Supervised Learning For Functions

Neural Operators

Theory

Closing

Recurrent Neural Operator

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.

Talk Outline

Algorithms on Function Space

Supervised Learning For Functions

Neural Operators

Theory

Closing

Numerical Experiments

2D Incompressible Navier Stokes Equation

Formulation Chandler and Kerswell [3]

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

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

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

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

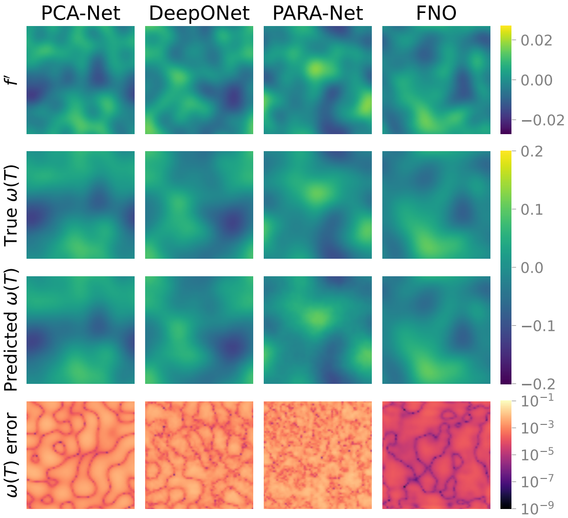


Figure: Learned model predictions for inputs resulting in **median** test errors for networks of size $w = 128$ / $d_f = 16$ trained on $N = 10000$ data.

Navier Stokes Equation

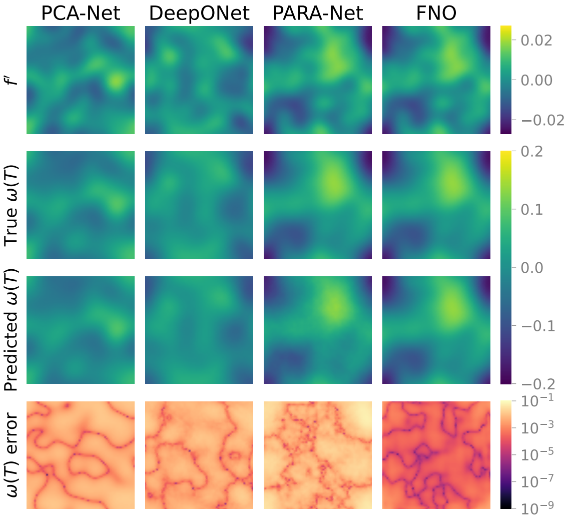


Figure: Learned model predictions for inputs resulting in **worst** test errors for networks of size $w = 128$ / $d_f = 16$ trained on $N = 10000$ data.

Navier-Stokes Equation Output Space

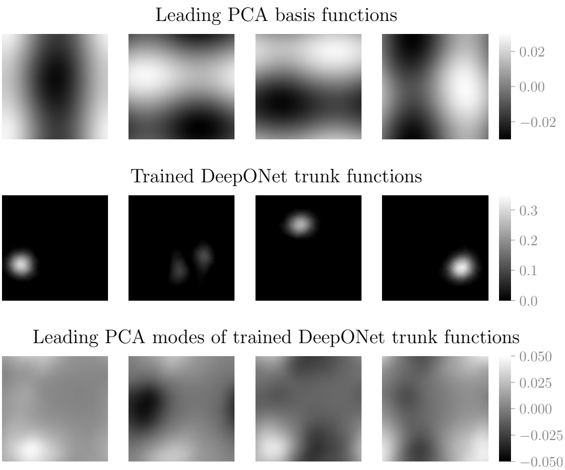


Figure: Comparison of output space bases: PCA-Net and DeepONet.

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

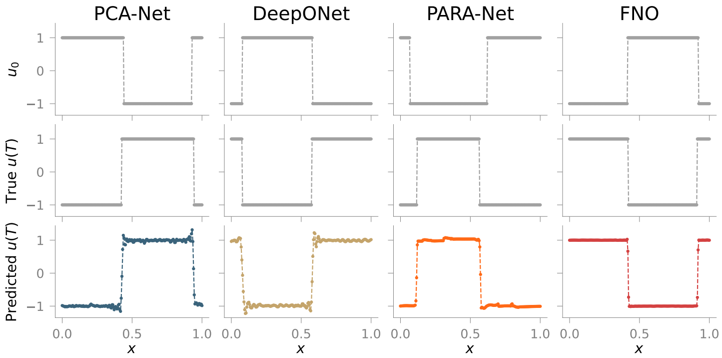


Figure: Learned solution predictions for inputs resulting in **median** test errors for networks of size $w = 128$ / $d_f = 16$ trained on $N = 10000$ data.

Advection Equation

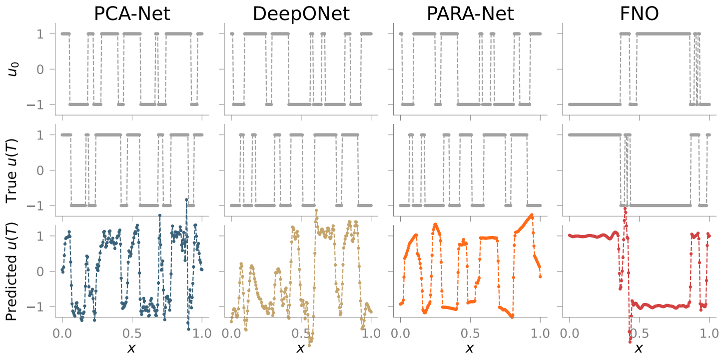


Figure: Learned solution predictions for inputs resulting in **worst** test errors for networks of size $w = 128$ / $d_f = 16$ trained on $N = 10000$ data.

Test Error vs. Cost

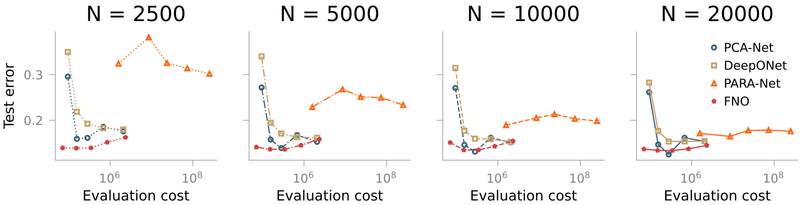


Figure: Test error vs. cost.

Advection Equation Output Space

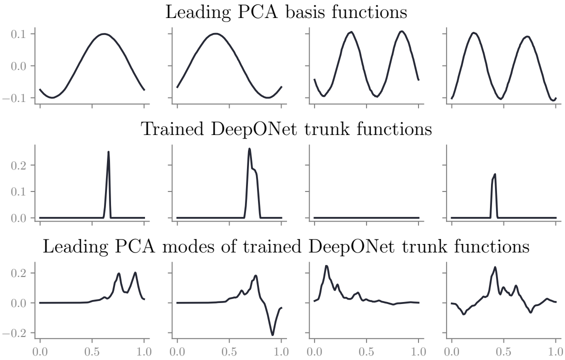


Figure: Comparison of output space bases: PCA-Net and DeepONet.

sectionUniversal Approximation

Talk Outline

Universal Approximation

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

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

Encoder-Decoder Approach

Theorem Kovacvhki '22 [12, 10]

- $\mathcal{U}$, $\mathcal{V}$ Banach spaces with the *approximation property* (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

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

Lanthaler, Mishra and Karniadakis '21 [15] (DeepONet)