Kernel Methods are Competitive for Operator Learning

Pau Batlle[1], Matthieu Darcy[1], Bamdad Hosseini[2], Houman Owhadi[1]

[1]California Institute of Technology [2]University of Washington

ICIAM

August 22nd, 2023

Table of Contents

The operator learning problem

The operator learning problem (informal version)

Let $\{u_i, v_i\}_{i=1}^N$ be N elements of $\mathcal{U} \times \mathcal{V}$ such that

$$\mathcal{G}^\dagger(u_i) = v_i, \quad \text{for } i = 1, \dots, N\,.$$

The operator learning problem is summarized as :

$$\text{Given the data } \{u_i, v_i\}_{i=1}^N \text{ approximate } \mathcal{G}^\dagger.$$

Throughout this talk

$$\mathcal{U} \text{ is a space of functions } u\,:\,\Omega \to \mathbb{R}$$
$$\mathcal{V} \text{ is a space of functions } v\,:\,D \to \mathbb{R}.$$

In this talk

Past work has focused on Operator Neural Networks[1,2,3] that generalize Neural Networks to functional inputs and outputs. However they have not been benchmarked against simpler methods.

Our contribution

We propose a family of kernel based-methods that are **simple, fast** and **competitive in accuracy.** The methods are natural benchmarks for more complex methods.

[1] Zongyi Li et al. *Fourier Neural Operator for Parametric Partial Differential Equations*. 2020.

[2] Kaushik Bhattacharya et al. *Model Reduction and Neural Networks for Parametric PDEs*. 2021. arXiv: 2005.03180 [math.NA].

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

Table of Contents

The operator learning problem

The operator learning problem

Let $\{u_i, v_i\}_{i=1}^N$ be N elements of $\mathcal{U} \times \mathcal{V}$ such that

$$\mathcal{G}^\dagger(u_i) = v_i, \quad \text{for } i = 1, \ldots, N\,.$$

Let $\phi : \mathcal{U} \to \mathbb{R}^m$ and $\varphi : \mathcal{V} \to \mathbb{R}^n$ be bounded linear operators.

Given the data $\{\phi(u_i), \varphi(v_i)\}_{i=1}^N$ approximate $\mathcal{G}^\dagger$.

A common example is the case where we have pointwise values of the functions:

$$\phi : u \mapsto \big(u(x_1), u(x_2), \ldots, u(x_m)\big)^T \quad \text{and} \quad \varphi : v \mapsto \big(v(y_1), v(y_2), \ldots, v(y_n)\big)^T\,.$$

Diagram summary

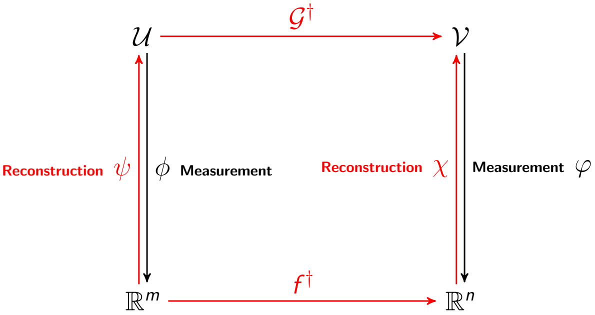
Summary of our method

Given the data $\{\phi(u_i), \varphi(v_i)\}_{i=1}^N$ our method to approximate $\mathcal{G}^\dagger$:

$$\mathcal{G}^\dagger(u_i) = v_i, \quad \text{for } i = 1, \ldots, N\,.$$

can be summarized in two steps:

1. Define the reconstructions ψ and χ as the optimal recovery map.
2. Approximate the function $f^\dagger$ using a kernel method.



Optimal recovery

We will assume that $\mathcal{U}$ and $\mathcal{V}$ are RKHSs arising from kernels Q and K respectively. The reconstruction operators are defined as optimal recovery maps

$$\psi(\phi(u)) := \underset{w\in\mathcal{U}}{\arg\min} \|w\|_Q \quad \text{s.t.} \quad \phi(w) = \phi(u),$$
$$\chi(\varphi(v)) := \underset{w\in\mathcal{V}}{\arg\min} \|w\|_K \quad \text{s.t.} \quad \varphi(w) = \varphi(v),$$

The maps are the minmax optimal recovery of u and v respectively[4]. Optimal recovery maps can be expressed in closed form using standard representer theorems for kernel interpolation:

$$\psi(\phi(u))(x) = Q(x, X)Q(X, X)^{-1}\phi(u) \quad \text{and} \quad \chi(\varphi(v))(y) = K(y, Y)K(Y, Y)^{-1}\varphi(v).$$

[4] Houman Owhadi and Clint Scovel. *Operator-Adapted Wavelets, Fast Solvers, and Numerical Homogenization: From a Game Theoretic Approach to Numerical Approximation and Algorithm Design*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2019.

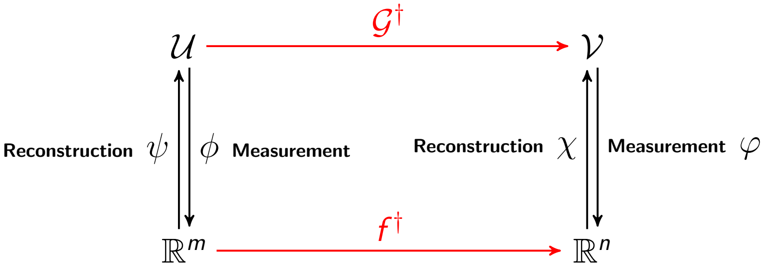
Recovery of $f^\dagger$

Once the reconstruction operators ψ and χ are defined, our best strategy is to reconstruct $f^\dagger$ in the diagram:

$$\bar{f} \approx f^\dagger := \varphi \circ \mathcal{G}^\dagger \circ \psi$$

and to approximate the operator $\mathcal{G}^\dagger$ with the operator

$$\bar{\mathcal{G}} := \chi \circ \bar{f} \circ \phi\,.$$



Recovery of $f^\dagger$

We approximate $f^\dagger : \mathbb{R}^m \to \mathbb{R}^n$ by optimal recovery in a **vector valued** RKHS. Let $\Gamma : \mathbb{R}^m \times \mathbb{R}^m \to \mathcal{L}(\mathbb{R}^n)$ be an **matrix valued kernel**. The (simplest) choice of Γ is the diagonal kernel

$$\Gamma(\boldsymbol{u}, \boldsymbol{u}') = S(\boldsymbol{u}, \boldsymbol{u}')\boldsymbol{I}_{n\times n}$$

where $S(\boldsymbol{u}, \boldsymbol{u}')$ is an arbitrary, real valued kernel. This is equivalent to recovering the vector valued $f^\dagger : \mathbb{R}^m \to \mathbb{R}^n$ **independently component wise**:

$$\bar{f}_j := \underset{h \in \mathcal{H}_S}{\arg\min} \|h\|_S \quad \text{s.t.} \qquad h(\phi(u_i)) = (\varphi(v_i))_j \qquad \text{for} \quad i = 1, \ldots, N.$$

This map can also be expressed in closed form

$$\bar{f} := \Gamma(\cdot, \mathbf{U})\Gamma(\mathbf{U}, \mathbf{U})^{-1}\mathbf{V}\,,$$

where $U_i := \phi(u_i)$ and $V_i := \varphi(v_i)$.

Why such a simple method?

The kernel S can be a standard kernel such as the linear[5], squared exponential or Matérn kernel. This simple choice already offers several advantages:

1. Low cost in training (< 5 seconds on a workstation) and at inference (in the low-medium data regime).
2. Competitive accuracy.
3. Empirically robust to choice of hyper-parameters/kernels.
4. Simple to implement: several libraries solve this problem out of the box.
5. The Gaussian process interpretation provides uncertainty quantification.
6. Convergence guarantees.

---

[5] Equivalent to doing linear regression

Assumptions for convergence guarantees

$$\mathcal{U} = \mathcal{H}_Q \text{ is an RKHS of functions } u : \Omega \to \mathbb{R}$$
$$\mathcal{V} = \mathcal{H}_K \text{ is an RKHS of functions } v : D \to \mathbb{R}.$$

Assumption (Two categories of assumptions)

- *Accuracy of the reconstruction operators ψ and χ: regularity of the domains, regularity of kernels Q and K, space-filling property of the collocation points.*
- *Approximation of $\mathcal{G}^\dagger$: regularity of the operator $\mathcal{G}^\dagger$, regularity of kernel S^n, resolution and space filling property of the data.*

Theorem (Condensed version of Main Theorem)

$$\lim_{n,m\to\infty} \lim_{N\to\infty} \sup_{u\in B_R(\mathcal{H}_Q)} \|\mathcal{G}^\dagger(u) - \chi^m \circ \bar{f}_N^{m,n} \circ \phi^n(u)\|_{H^{t'}(D)} \to 0\,.$$

Measurement invariance

Mesh invariance is a key property for operator learning methods: this translates to being able to predict the output of a test input function $\tilde{u}$ with a new $\tilde{\phi}(\tilde{u})$. We can do this by using the optimal recovery map $\tilde{\psi}$ that is defined from $\tilde{\phi}$. This gives a new function $h^\dagger$ which is approximated by

$$\bar{h} := \tilde{\varphi} \circ \chi \circ \bar{f} \circ \phi \circ \tilde{\psi} \equiv \tilde{\varphi} \circ \bar{\mathcal{G}} \circ \tilde{\psi}.$$

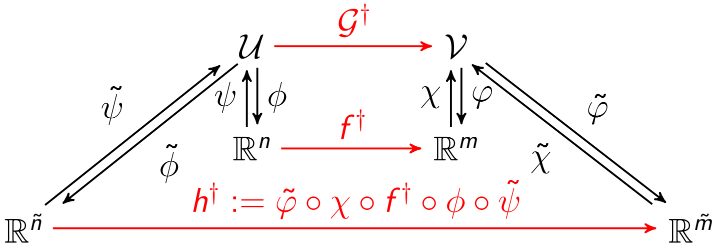


Figure: Mesh invariance of the method.

Table of Contents

Complexity-accuracy tradeoff

We evaluate our operator learning methods through the **cost-accuracy tradeoff**:

- The **accuracy** is measured by the relative L^2 loss:

$$\mathcal{R}_N(\mathcal{G}) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\left\|\mathcal{G}(u_i) - \mathcal{G}^\dagger(u_i)\right\|_{L^2}}{\|\mathcal{G}^\dagger(u_i)\|_{L^2}}\right]$$

- The **complexity** depends on the **training cost** (qualitative metrics) and the **inference cost** (measured in floating point operations - FLOPs).

We compare the test performance of our method with different choices of the kernel S using the examples from two comparison papers[6],[7].

[6] Maarten V. de Hoop et al. *The Cost-Accuracy Trade-Off In Operator Learning With Neural Networks*. 2022.

[7] Lu Lu et al. "A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data". In: *Computer Methods in Applied Mechanics and Engineering* 393 (2022), p. 114778. ISSN: 0045-7825.

Summary of results: accuracy

	Low-data regime			High-data regime			
	Burger's	**Darcy problem**	**Advection I**	**Advection II**	**Hemholtz**	**Structural Mechanics**	**Navier Stokes**
DeepONet	2.15%	2.91%	0.66%	15.24%	5.88%	5.20%	3.63%
POD-DeepONet	1.94%	2.32%	0.04%	n/a	n/a	n/a	n/a
FNO	1.93%	2.41%	0.22%	13.49%	1.86%	4.76%	0.26%
PCA-Net	n/a	n/a	n/a	12.53%	2.13%	4.67%	2.65%
PARA-Net	n/a	n/a	n/a	16.64%	12.54%	4.55%	4.09%
Linear	36.24%	6.74%	$2.15 \times 10^{-13}\%$	11.28%	10.59%	27.11%	5.41%
Best of Matérn/RQ	2.15%	2.75%	$2.75 \times 10^{-3}\%$	11.44%	1.00%	5.18%	0.12%

Table: Summary of numerical results: we report the L^2 relative test error of our numerical experiments and compare the kernel approach with variations of DeepONet , FNO, PCA-Net and PARA-Net.

Inverse problem for Darcy's flow

Let $D = (0,1)^2$ and consider the two-dimensional Darcy flow problem[8]:

$$
\begin{aligned}
-\nabla \cdot (u(x)\nabla v(x)) &= f, \quad x \in D \\
u(x) &= 0, \quad \partial D
\end{aligned}
$$

In this case, we are interested in learning the mapping from the permeability field u to the solution v (here f is considered fixed):

$$\mathcal{G}^\dagger : u(x) \mapsto v(x).$$

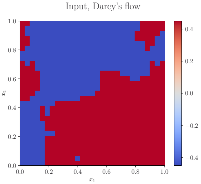
The coefficient u is sampled by $u = \psi(\mu)$ where $\mu = \mathcal{GP}(0, (-\Delta + 9I)^{-2})$ is a Gaussian random field and ψ is binary function.

[8] Lu et al., "A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data".

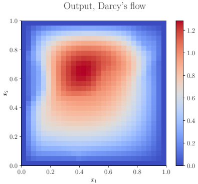
Inverse problem for Darcy's flow

Method	Accuracy
DeepONet	2.91 %
FNO	2.41 %
POD-DeepONet	2.32 %
Linear Regression	6.74 %
GP (Matérn kernel)	2.75%

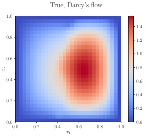
Table: L^2 relative error on the Darcy problem.



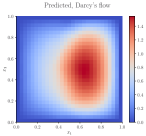
(a) Input



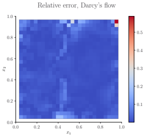
(b) Output



(c) True



(d) Predicted



(e) Relative Error

Navier-Stokes

In the periodic domain $\mathcal{D} = [0, 2\pi]^2$, the vorticity-stream $(\omega - \psi)$ formulation of the incompressible Navier-Stokes equations[9] is

$$\begin{aligned} \frac{\partial w}{\partial t} + (v \cdot \nabla)\omega - \nu \Delta \omega &= f \\ \omega &= -\Delta \psi \\ \int_D \psi &= 0 \\ v &= \left(\frac{\partial \psi}{\partial x_2}, -\frac{\partial \psi}{\partial x_1} \right) \end{aligned}$$

The map of interest is the map from the forcing term f to the vorticity field w at a given time $t = T$:

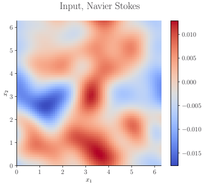
$$\mathcal{G} : f \mapsto w(\cdot, T)$$

[9] Hoop et al., *The Cost-Accuracy Trade-Off In Operator Learning With Neural Networks*.

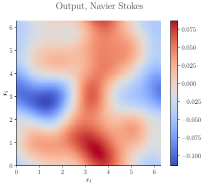
Navier-Stokes

Method	Accuracy
DeepONet	3.63 %
FNO	0.26 %
PCA-Net	2.32 %
Linear Regression	5.41 %
GP (Matérn kernel)	0.12%

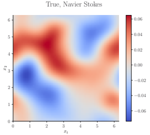
Table: L^2 relative error on Navier-Stokes.



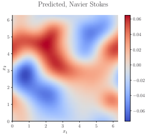
(a) Input



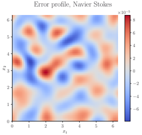
(b) Output



(c) True



(d) Predicted



(e) Error

Inference complexity: high data regime

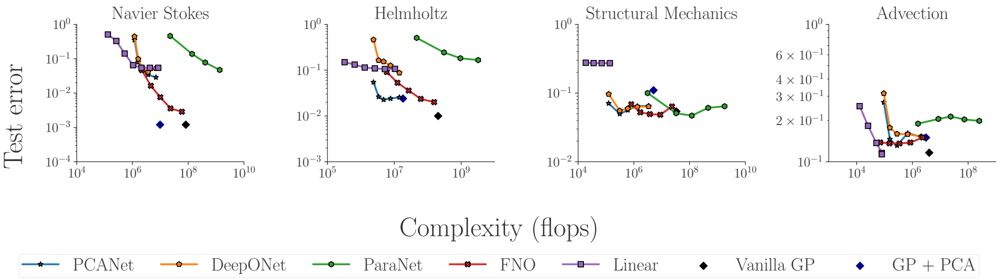


Figure: Linear model refers to the linear kernel, vanilla GP is our implementation with the nonlinear kernels and minimal preprocessing, GP+PCA corresponds to preprocessing through PCA both the input and the output to reduce complexity.

Data taken from Hoop et al., *The Cost-Accuracy Trade-Off In Operator Learning With Neural Networks*.

Conclusion

Our key contributions are:

- A simple, low-cost, and competitive kernel method for operator learning that is a good baseline for many tasks.
- Theoretical guarantees for these methods.
- A general framework for doing operator learning with kernel methods.

Paper to appear in Journal of Computational Physics Pau Batlle et al. *Kernel Methods are Competitive for Operator Learning*. 2023. arXiv: 2304.13202