SEE CORRECTION FOR THIS ARTICLE

PNAS BRIEF REPORT APPLIED MATHEMATICS OPEN ACCESS

Check for updates

Elliptic PDE learning is provably data-efficient

Nicolas Boullé[a,1], Diana Halikias[b], and Alex Townsend[b]

Edited by David Donoho, Stanford University, Stanford, CA; received March 8, 2023; accepted July 21, 2023

Partial differential equations (PDE) learning is an emerging field that combines physics and machine learning to recover unknown physical systems from experimental data. While deep learning models traditionally require copious amounts of training data, recent PDE learning techniques achieve spectacular results with limited data availability. Still, these results are empirical. Our work provides theoretical guarantees on the number of input–output training pairs required in PDE learning. Specifically, we exploit randomized numerical linear algebra and PDE theory to derive a provably data-efficient algorithm that recovers solution operators of three-dimensional uniformly elliptic PDEs from input–output data and achieves an exponential convergence rate of the error with respect to the size of the training dataset with an exceptionally high probability of success.

deep learning | inverse problems | sample complexity | neural operators

Many scientific breakthroughs have come from deriving new partial differential equations (PDEs) from first principles to model real-world phenomena and simulating them on a computer to make predictions. However, many crucial problems currently lack an adequate mathematical formulation. It is not clear how to derive PDEs to describe how turbulence sheds off the wing of a hypersonic aircraft, how *Escherichia coli* bacteria swim in unison to form an active fluid, or how atomic particles behave with long-range interactions. Rather than working from first principles, scientists are now looking to derive PDEs from real-world data using deep learning techniques (1).

The success of deep learning in language models, visual object recognition, and drug discovery is well known (2). The emerging field of PDE learning hopes to extend this to discovering new physical laws by supplying deep learning models with experimental or observational data (1, 3). PDE learning commonly seeks to recover features such as symmetries, conservation laws, solution operators, and the parameters of a family of hypothesized PDEs. In most deep-learning applications, a large amount of data is needed, which is often unrealistic in engineering and biology. However, PDE learning can be shockingly data-efficient in practice (4). In particular, surprisingly few data are used to learn the solution operator, which maps the forcing term to the solution of the PDE.

In this paper, we provide a theoretical explanation of this behavior by showing that, for $\epsilon > 0$ sufficiently small, one can recover an ϵ-approximation to the solution operator of a three-dimensional (3D) elliptic PDE with a training dataset of size about $\mathcal{O}(\log^5(1/\epsilon))$. Elliptic PDEs, such as the steady-state heat equation, are ubiquitous in physics and model diffusion phenomena. Solution operators can produce surrogate data for data-intensive machine-learning approaches such as learning reduced order models for design optimization in engineering, uncovering physics in climate models, and PDE recovery (1).

To illustrate the observed data-efficiency of PDE learning, we compare the performance of three techniques (4–6) for recovering the solution operator associated with the two-dimensional (2D) Poisson equation in Fig. 1. We vary the size of the training dataset, consisting of random forcing terms and corresponding solutions obtained by a numerical solver. We then evaluate the accuracy of the predicted solutions on a testing dataset with new forcing terms. The three methods are based on deep learning and differ in their neural network architectures. While the Fourier Neural Operator (5) exploits the fast Fourier transform for computationally efficient training, DeepONet (4) and GreenLearning (6) achieve a faster convergence rate on small training datasets. Here, DeepONet employs a complex network architecture with many parameters. In contrast, GreenLearning leverages prior knowledge that the solution operator is an integral operator and the approximation power of rational neural networks (7). Green's function learning is observed to be the most data-efficient in Fig. 1, as for a fixed training dataset size, it achieves the smallest testing error. All methods plateau due to discretization errors, and the training procedure gets stuck in a local minimum of the loss landscape rather than finding the global minimum. The rapid decay of testing errors prior to the plateau motivates our main result.

Author affiliations: [a]Isaac Newton Institute for Mathematical Sciences, University of Cambridge, Cambridge CB3 0EH, United Kingdom; and [b]Mathematics Department, Cornell University, Ithaca, NY 14853-4201

Author contributions: N.B., D.H., and A.T. designed research; performed research; analyzed data; and wrote the paper.

The authors declare no competing interest.

Copyright © 2023 the Author(s). Published by PNAS. This open access article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

[1] To whom correspondence may be addressed. Email: nb690@cam.ac.uk.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2303904120/-/DCSupplemental.

Published September 18, 2023.

Downloaded from https://www.pnas.org by "CORNELL UNIVERSITY, E-RESOURCES AND SERIALS MANAGEMENT" on May 22, 2024 from IP address 132.236.187.103.

Downloaded from https://www.pnas.org by "CORNELL UNIVERSITY, E-RESOURCES AND SERIALS MANAGEMENT" on May 22, 2024 from IP address 132.236.187.103.

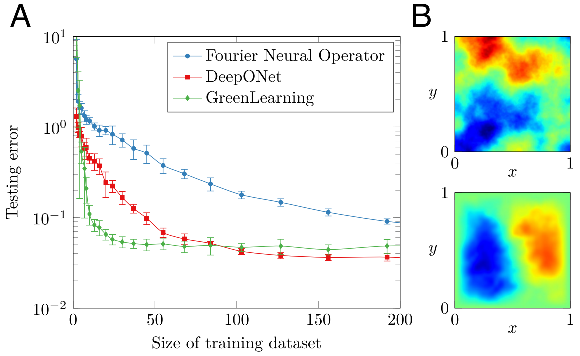

Fig. 1. Elliptic PDE learning methods can be data-efficient. (*A*) Performance of three deep learning techniques in approximating the solution operator of the 2D Poisson equation with zero Dirichlet boundary condition on the domain $[0,1]^2$. On small datasets, DeepONet and GreenLearning attain exponential decay of the testing error, while Fourier Neural Operator (FNO) attains algebraic decay. (*B*) A forcing term (*Top*) and corresponding predicted solution (*Bottom*) to the 2D Poisson equation by a FNO.

There is a lack of understanding of the efficiency of PDE learning methods with limited training data (4). This work provides theoretical insights by constructing a provably data-efficient algorithm that can achieve exponential convergence for learning solution operators of elliptic PDEs.

Consider an unknown uniformly elliptic PDE in three dimensions, defined on a bounded domain $\Omega \subset \mathbb{R}^3$ with Lipschitz smooth boundary, with variable coefficients of the form:

$$\mathcal{L}u = -\nabla \cdot (A(x)\nabla u) = f, \quad x \in \Omega, \quad u|_{\partial\Omega} = 0, \qquad [1]$$

where the coefficient matrix A has bounded coefficient functions and is symmetric positive definite for all $x \in \Omega$. The weak assumptions on Ω and $A(x)$ allow for corner singularities and low regularity of the coefficients. The training data consist of pairs of random forcing terms $f_1, \ldots, f_N$ and corresponding solutions $u_1, \ldots, u_N$ such that $\mathcal{L}u_j = f_j$ for $1 \le j \le N$. Deep learning techniques use these data to predict solutions to Eq. **1** at new forcing terms by recovering the action of the solution operator $\mathscr{F}$, which is given by

$$\mathscr{F}(f) = \int_\Omega G(x,y) f(y)\,\mathrm{d}y, \qquad [2]$$

where G is the associated Green's function. For example, we visualize in Fig. 2*A* the Green's function associated with the 1D Poisson equation. The random forcing terms in the training dataset are sampled from a Gaussian process (GP), i.e., they follow a multivariate Gaussian distribution when sampled on a grid, and the covariance kernel determines the correlation between the function's entries and its smoothness.

Recent work (11) proves that for any $\epsilon > 0$ and 3D elliptic PDEs, a large number of input–output training pairs of size about $\mathcal{O}(\epsilon^{-6})$ is sufficient to recover an ϵ-approximation $\tilde{\mathscr{F}}$ to $\mathscr{F}$ such that

$$\|\mathscr{F} - \tilde{\mathscr{F}}\|_2 \le \epsilon \|\mathscr{F}\|_{\mathrm{HS}},$$

where $\|\cdot\|_2$ is the solution operator norm and $\|\cdot\|_{\mathrm{HS}}$ is the Hilbert–Schmidt norm, while (12) exploits sparse Gaussian elimination techniques and piecewise polynomial inputs to recover $\mathscr{F}$ with $\mathcal{O}(\log(1/\epsilon)^4)$ pairs. Once the ϵ-approximation to $\mathscr{F}$ has been constructed, $\tilde{\mathscr{F}}$ can be used to study the stability and regularity of solutions of the PDE, for example, to see whether small perturbations of the input function lead to small changes in the output solution or whether the solution has certain smoothness or decay properties for all forcing terms. Moreover, $\tilde{\mathscr{F}}$ can be used in numerical methods for approximating the solution of the PDE. By discretizing the input function and applying $\tilde{\mathscr{F}}$ as a surrogate for $\mathscr{F}$, one can obtain a numerical solution of the PDE that approximates the true solution. The integral kernel associated with the Hilbert–Schmidt operator $\tilde{\mathscr{F}}$ is also of interest, as it is an approximation to the Green's function, which can be exploited to recover linear conservation laws, symmetries, boundary effects, and dominant modes (6).

It is an open problem whether one can achieve exponential convergence while recovering the hierarchical structure of the solution operator (8, 13), as then the technique could potentially extend beyond elliptic solution operators. Our main result solves this open problem by exploiting the hierarchical structure of G (9) and randomized linear algebra techniques (10, 13). We derive a randomized algorithm that provably succeeds with exceptionally high probability and needs a training dataset size of only $\mathcal{O}(\log(1/\epsilon)^5[\log(\log(1/\epsilon)) + \log(1/\Gamma_\epsilon)]^4)$ input–output pairs.

Theorem 1. *Let $\epsilon > 0$ be sufficiently small and $\mathscr{F}$ be the solution operator associated with a 3D uniformly elliptic PDE of the form in Eq.* ***1****. There exists a randomized algorithm that constructs an ϵ-approximation $\tilde{\mathscr{F}}$ to $\mathscr{F}$ such that*

$$\|\mathscr{F} - \tilde{\mathscr{F}}\|_2 \le \epsilon \|\mathscr{F}\|_{\mathrm{HS}},$$

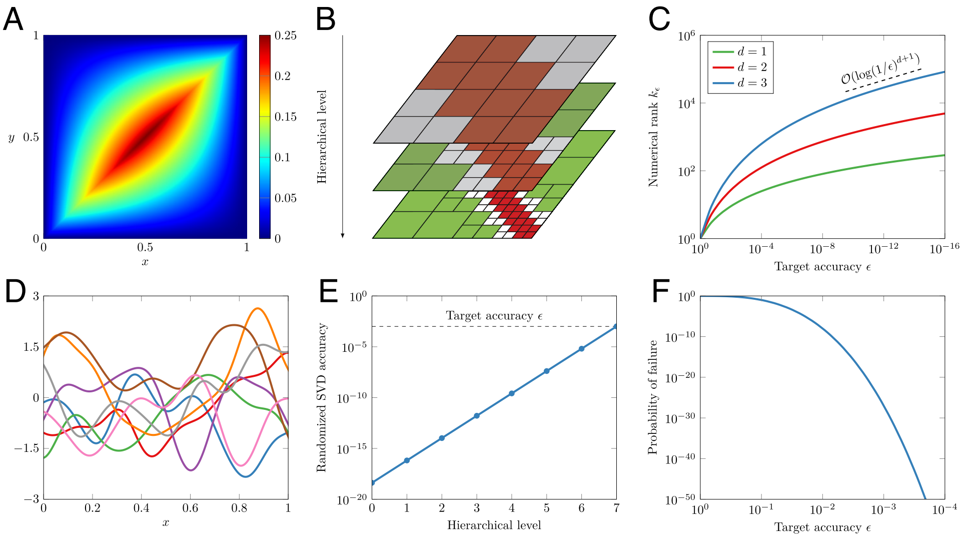

Fig. 2. Properties of elliptic PDEs can be exploited to construct a provably data-efficient algorithm for recovering solution operators. (*A*) The Green's function associated with the 1D Poisson equation, which is the kernel of the solution operator. (*B*) We use the multiscale (hierarchical) structure of a Green's function (8). (*C*) On well-separated domains, the Green's function has rapidly decaying singular values (9), so it is efficiently recovered by the randomized singular value decomposition (SVD) (10). (*D*) Forcing terms for the training dataset are randomly sampled from a Gaussian process. (*E*) The accuracy of the randomized SVD is carefully adapted on each hierarchical level to counterbalance the potential accumulation of errors in the reconstruction process. (*F*) An upper bound on the probability of failure of the reconstruction algorithm as a function of ϵ.

using $\mathcal{O}(\log(1/\epsilon)^5[\log(\log(1/\epsilon)) + \log(1/\Gamma_\epsilon)]^4)$ *input–output pairs with probability* $\geq 1 - e^{-\log(1/\epsilon)^3}$.

The main contribution of Theorem 1 is a theoretical upper bound on the amount of training data required in elliptic PDE learning problems, which should deepen our understanding of existing deep learning techniques. Hence, the exponential convergence rate in Theorem 1 matches the one observed in the deep learning experiments of Fig. 1*A*. We believe that this learning rate is near-optimal, as it exploits the multiscale structure of Green's functions (Fig. 2 *B* and *C*) and depends on the training dataset. The factor $0 < \Gamma_\epsilon \leq 1$ measures the quality of the training dataset at probing the dominant modes of the PDE, and a technical definition is available in *SI Appendix*. We emphasize that the error bound must include a factor that quantifies the quality of the training dataset. If the forcing terms are too smooth, then Γ_ϵ is small. In contrast, choosing the covariance kernel of the GP such that the sampled functions are oscillatory usually ensures that Γ_ϵ is reasonable for learning G. In short, a small number of sufficiently diverse forcing terms is required (Fig. 2*D*).

The algorithm constructed in the proof of Theorem 1 achieves an approximation error measured in the solution operator norm. This mimics the typical measurement of accuracy of PDE learning techniques by comparing true and predicted solutions on a testing dataset of square-integrable forcing terms. Additionally, Theorem 1 employs random input–output pairs, where the forcing terms are sampled from a GP, so there is always some probability of failure. Fortunately, we show that this probability is exceptionally small. For $\epsilon < 10^{-3}$, failure is a once-in-a-cosmic-epoch event (Fig. 2*F*).

Theorem 1 is challenging to prove, and the whole argument is in *SI Appendix*. The proof relies on the fact that the solution operator associated with a 3D elliptic PDE is an integral operator in the form of Eq. **2**. First, the Green's functions related to 3D elliptic operators are square-integrable and have a bounded decay rate away from the diagonal of $\Omega \times \Omega$ (14). Second, they possess a hierarchical structure (9) in the sense that they have rapidly decaying singular values when restricted to off-diagonal parts of the domain (green blocks in Fig. 2*B*). We leverage the hierarchical structure, which has been historically exploited by fast solvers, in a data-driven context where the PDE is unknown. Combining these properties enables a generalization of the randomized SVD (10) known as the peeling algorithm (8) to simultaneously learn the off-diagonal blocks at any level of the hierarchy.

While the peeling algorithm is traditionally used to recover hierarchical matrices efficiently from matrix-vector products, we generalize it to approximate infinite-dimensional integral operators. To do so, we leverage insights from recent work that extends the peeling algorithm to arbitrary hierarchical partitions and dimensions (15). This gives us a strategy to recover the Green's function level by level. However, proving the stability of peeling is an open question in numerical linear algebra. This is because the approximation errors from one level can potentially accumulate exponentially at later levels, thus degrading the convergence rate (8, 11).

We overcome this theoretical obstacle in the infinite-dimensional context by requiring an adaptive approximation accuracy at each level of the hierarchy. The peeling algorithm ensures that the large-scale features of a Green's function are first learned to high accuracy by the randomized SVD. Then, we progressively decrease the accuracy requirement at subsequent levels, ensuring an overall ϵ-approximation on each level of the partition at the end (Fig. 2*E*). The rapidly decaying singular values of the Green's function on off-diagonal parts of the domain (Fig. 2*C*) enable us to maintain a near-optimal exponential convergence rate with respect to the size of the training dataset. We then construct a global ϵ-approximant by neglecting G near the diagonal of the domain.

As one usually employs deep learning techniques to learn solution operators, our theoretical contributions can also lead to practical benefits. We believe that future training datasets benefit from taking into account prior knowledge of the PDE to improve the quality of the forcing terms at learning the solution operator. Similar ideas have already been employed in the field of visual object recognition through data-augmentation techniques. There is also an opportunity to design neural network architectures with hierarchical structures to capture the long-range interactions in PDE models. Finally, enforcing a different accuracy at different scales might improve the computational efficiency of existing PDE learning approaches.

In summary, we constructed a randomized algorithm that provably achieves an exponential convergence rate for approximating the solution operator associated with 3D elliptic PDEs in terms of the size of the training dataset. This provides a theoretical explanation for the observed performance of recent deep learning techniques in PDE learning. The proof techniques can be adapted to include elliptic PDEs in any dimension and time-dependent PDEs (16). Recovering solution operators associated with hyperbolic PDEs, like wave equations, remains a significant open challenge. Moving forward, we plan to ramp up PDE learning techniques to handle noisy experimental data, deal with data from emerging transient dynamics, and enforce conservation laws onto our solutions.

Data, Materials, and Software Availability. The proof of Theorem 1 and details of the numerical experiments are available in *SI Appendix*. Codes and datasets have been deposited in Zenodo (17). All other data are included in the manuscript and/or *SI Appendix*.

ACKNOWLEDGMENTS. This work is supported by NSF grants DMS-1952757, DMS-2045646, and DGE-2139899. N.B. was supported by an INI-Simons Postdoctoral Research Fellowship.

1. G. E. Karniadakis *et al.*, Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).
2. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
3. M. Raissi, A. Yazdani, G. E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* **367**, 1026–1030 (2020).
4. L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **3**, 218–229 (2021).
5. Z. Li *et al.*, "Fourier neural operator for parametric partial differential equations" in *ICLR* (2021).
6. N. Boullé, C. J. Earls, A. Townsend, Data-driven discovery of Green's functions with human-understandable deep learning. *Sci. Rep.* **12**, 1–9 (2022).
7. N. Boullé, Y. Nakatsukasa, A. Townsend, "Rational neural networks" in *NeurIPS* (2020), vol. 33, pp. 14243–14253.
8. L. Lin, J. Lu, L. Ying, Fast construction of hierarchical matrix representation from matrix-vector multiplication. *J. Comput. Phys.* **230**, 4071–4087 (2011).
9. M. Bebendorf, W. Hackbusch, Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients. *Numer. Math.* **95**, 1–28 (2003).
10. N. Halko, P. G. Martinsson, J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* **53**, 217–288 (2011).
11. N. Boullé, A. Townsend, Learning elliptic partial differential equations with randomized linear algebra. *Found. Comput. Math.* **23**, 709–739 (2023).
12. F. Schäfer, H. Owhadi, Sparse recovery of elliptic solvers from matrix-vector products. arXiv [Preprint] (2021). https://doi.org/10.48550/arXiv.2110.05351.
13. P. G. Martinsson, J. A. Tropp, Randomized numerical linear algebra: Foundations and algorithms. *Acta Numer.* **29**, 403–572 (2020).
14. M. Grüter, K. O. Widman, The Green function for uniformly elliptic equations. *Manuscripta Math.* **37**, 303–342 (1982).
15. J. Levitt, P. G. Martinsson, Randomized compression of rank-structured matrices accelerated with graph coloring. arXiv [Preprint] (2022). http://arxiv.org/abs/2205.03406 (Accessed 3 August 2023).
16. N. Boullé, S. Kim, T. Shi, A. Townsend, Learning Green's functions associated with time-dependent partial differential equations. *J. Mach. Learn. Res.* **23**, 1–34 (2022).
17. N. Boullé, D. Halikias, A. Townsend, Code for "Elliptic PDE learning is provably data-efficient". Zenodo. https://doi.org/10.5281/zenodo.7701683. Deposited 6 March 2023.

Downloaded from https://www.pnas.org by "CORNELL UNIVERSITY, E-RESOURCES AND SERIALS MANAGEMENT" on May 22, 2024 from IP address 132.236.187.103.