Learning elliptic PDEs with randomized linear algebra

Alex Townsend
Cornell University
townsend@cornell.edu

Joint work with
Nicolas Boullé
Chris Earls
Introduction

**Question:** Can one “learn” an unknown linear PDE from input-output data?
**Question:** Can one “learn” an unknown linear PDE from input-output data?
**Question:** Can one “learn” an unknown linear PDE from input-output data?

**Input-output data:** \( \{(f_j, u_j)\}_{j=1}^{k+5} \) such that \( \mathcal{L}u_j = f_j, \quad u_j|_{\partial\Omega} = g. \)
**Question:** Can one “learn” an unknown linear PDE from input-output data?

**Input-output data:** \( \{(f_j, u_j)\}_{j=1}^{k+5} \) such that \( \mathcal{L} u_j = f_j, \quad u_j |_{\partial \Omega} = g. \)

For theory, I’ll focus on self-adjoint 2nd-order elliptic PDEs with Dirichlet bcs.
**Question:** Can one “learn” an unknown linear PDE from input-output data?

\[ \{(f_j, u_j)\}_{j=1}^{k+5} \] such that \[ \mathcal{L}u_j = f_j, \quad u_j|_{\partial\Omega} = g. \]

For theory, I’ll focus on self-adjoint 2nd-order elliptic PDEs with Dirichlet bcs.

Small selection of practical work:
[Brunton, Proctor, & Kutz, 16], [Rudy, Brunton, Proctor, & Kutz, 17], [Schaeffer, 17], [Raissi, Perdikaris, & Karniadakis, 17], [Raissi, 18], [Han, Jentzen, and E, 2018], [Khoo, Lu, & Ying, 2018], [Fan, Feliu-Faba, Lin, Ying, Zepeda-Nunez, 2018], [Raissi, Perdikaris, & Karniadakis, 19], [Gin, Shea, Brunton, & Kutz, 21]
Approaches for learning solution operator
Approaches for learning solution operator

\[ \mathcal{L} : \mathcal{H}^2(\Omega) \cap \mathcal{H}^1_0(\Omega) \to L^2(\Omega) \] is unbounded so tricky to make learning precise
Approaches for learning solution operator

\[ \mathcal{L} : \mathcal{H}^2(\Omega) \cap \mathcal{H}^1_0(\Omega) \to L^2(\Omega) \] is unbounded so tricky to make learning precise

Approximate the solution operator

\[ f_j \quad u_j \quad \rightarrow \quad \text{Train NN} \]
Approaches for learning solution operator

\[ \mathcal{L} : \mathcal{H}^2(\Omega) \cap \mathcal{H}^1_0(\Omega) \to L^2(\Omega) \] is unbounded so tricky to make learning precise

Approximate the solution operator

\[ f_j \quad u_j \rightarrow \text{Train NN} \]

Solution operator

\[ \text{Evaluate} \]
Approaches for learning solution operator

\[ L : \mathcal{H}^2(\Omega) \cap \mathcal{H}^1_0(\Omega) \rightarrow L^2(\Omega) \] is unbounded so tricky to make learning precise

Approximate the solution operator

\[
\begin{align*}
  f_j \\
u_j
\end{align*}
\]

Train NN

Solution operator

Evaluate

DeepONet

Fourier Neural Operator

DeepGreen

[Quanta Magazine; Lu et al, 2021]

[Quanta Magazine; Li et al, 2020]

[Gin et al., 2020]
1. Theoretical results

![Graph showing relative error (%) against input-output pairs. The y-axis is on a logarithmic scale ranging from $10^{-1}$ to $10^3$, and the x-axis ranges from 0 to 100.]
Main challenges

1. Theoretical results

- Type and number of training data
- Performance guarantees
- Neural network architectures
- Noise robustness
Main challenges

1. Theoretical results

2. Interpretability of the model

- Type and number of training data
- Performance guarantees
- Neural network architectures
- Noise robustness

[Li et al, 2020]
Main challenges

1. Theoretical results

![Graph showing relative error vs input-output pairs]

- Type and number of training data
- Performance guarantees
- Neural network architectures
- Noise robustness

2. Interpretability of the model

![Images showing initial vorticity and predictions at different times]

- Dominant modes
- Symmetries
- Conservation laws
- Singularities

[Li et al., 2020]
Green’s function
Green’s function

Equivalently, for a uniformly elliptic PDE, learn a Green’s function such that

\[ u_j(x) = \int_\Omega G(x, y)f_j(y)dy, \quad x \in \Omega, \quad 1 \leq j \leq k + 5 \]

This is a Hilbert-Schmidt (HS) integral operator.

[Feliu-Faba, Fan, & Ying, 2019]
Green’s function

Equivalently, for a uniformly elliptic PDE, learn a Green’s function such that

\[ u_j(x) = \int_{\Omega} G(x, y)f_j(y)dy, \quad x \in \Omega, \quad 1 \leq j \leq k + 5 \]

This is a Hilbert-Schmidt (HS) integral operator.

[Feliu-Faba, Fan, & Ying, 2019]

---

**Poisson equation**

\[-\nabla^2 u = f\]

\[u(0) = u(1) = 0\]

**Helmholtz equation**

\[\nabla^2 u + k^2 u = f\]

\[u(0) = u(1)\]
Self-adjoint elliptic PDEs in 1D, 2D, or 3D of the form:

\[ \mathcal{L}u := -\nabla \cdot (A(x) \nabla u) = f \quad \Rightarrow \quad u(x) = \int_{\Omega} G(x, y) f(y) \, dy \]

**Theorem** [Boullé & T., 2021]

There is a randomized algorithm that, for any \( \epsilon > 0 \), can construct an approximation \( \tilde{G} \) of \( G \) with \( O(\epsilon^{-6} \log^{4}(1/\epsilon)/\Gamma_{\epsilon}) \) input-output pairs \((f, u)\) such that

\[ \| G - \tilde{G} \|_{L^{2}(D \times D)} = O(\Gamma_{\epsilon}^{-3/4} \log^{3}(1/\epsilon) \epsilon) , \]

with high probability.

**Proof**

1. Randomized numerical linear algebra
2. Regularity of the Green’s function
Randomized numerical linear algebra
We can learn symmetric low-rank matrices via matrix-vector products $v \mapsto Xv$:

<table>
<thead>
<tr>
<th>Randomized SVD:</th>
</tr>
</thead>
</table>

[Halko, Martinsson, & Tropp, 2011], [Martinsson & Tropp, 2020]
We can learn symmetric low-rank matrices via matrix-vector products $v \mapsto Xv$:

Randomized SVD:

1. $n \times (k + 5)$

$$Y = \begin{pmatrix} \vdots \\
\end{pmatrix}$$

Tall-skinny Gaussian matrix with iid indep. entries [Halko, Martinsson, & Tropp, 2011], [Martinsson & Tropp, 2020]
We can learn symmetric low-rank matrices via matrix-vector products $v \mapsto Xv$:

**Randomized SVD:**

1. $n \times (k + 5)$

   \[ Y = \begin{pmatrix} \ldots \end{pmatrix} \]

   Tall-skinny Gaussian matrix with iid indep. entries

2. $Z = XY$

   Input-output data

[Halko, Martinsson, & Tropp, 2011], [Martinsson & Tropp, 2020]
We can learn symmetric low-rank matrices via matrix-vector products $v \mapsto Xv$:

**Randomized SVD:**

1. $n \times (k + 5)$
   
   Tall-skinny Gaussian matrix with iid indep. entries
   
   Input-output data

2. $Z = XY$
   
   orthonormal basis for $\text{col}(Z)$

3. $Q = \text{orth}(Z)$
   
   $A_k = QQ^*X$

[Halko, Martinsson, & Tropp, 2011], [Martinsson & Tropp, 2020]
We can learn symmetric low-rank matrices via matrix-vector products $v \mapsto Xv$:

**Randomized SVD:**

1. $n \times (k + 5)$
   
   $Y = \begin{pmatrix} \vdots \\ \vdots \end{pmatrix}$
   
   Tall-skinny Gaussian matrix with iid indep. entries

2. $Z = XY$
   
   Input-output data

3. $Q = \text{orth}(Z)$
   
   orthonormal basis for $\text{col}(Z)$

   $A_k = QQ^*X$

**Theorem (Halko, Martinsson, Tropp, 2011).**

*We can construct an approximation $A_k$ of $A$ from $k + 5$ random input vectors such that*

$$\mathbb{P} \left[ \|A - A_k\|_F \leq (1 + 15\sqrt{k + 5})\epsilon_k \right] \geq 0.999$$
We can learn symmetric low-rank matrices via matrix-vector products $v \mapsto Xv$:

**Randomized SVD:**

1. $n \times (k + 5)$
2. $Z = XY$
3. $Q = \text{orth}(Z)$ orthonormal basis for $\text{col}(Z)$

$A_k = QQ^*X$

Theorem (Halko, Martinsson, & Tropp, 2011).

We can construct an approximation $A_k$ of $A$ from $k+5$ random input vectors such that

$$\mathbb{P} \left[ \| A - A_k \|_F \leq (1 + 15\sqrt{k + 5})\epsilon_k \right] \geq 0.999$$
Generalization of the randomized SVD

Theorem [Boullé & T., 2021]

We can construct an approximation $A_k$ of $A$ from $k+5$ correlated random input vectors such that

$$\mathbb{P} \left[ \|A - A_k\|_F \leq (1 + 18\sqrt{k/\gamma_k})\varepsilon_k \right] \geq 0.999$$
Generalization of the randomized SVD

Standard Gaussian vectors

Prior knowledge about $A$ helps:

Correlated Gaussian vectors

**Theorem** [Boullé & T., 2021]

*We can construct an approximation $A_k$ of $A$ from $k + 5$ correlated random input vectors such that*

$$\mathbb{P} \left[ \| A - A_k \|_F \leq (1 + 18 \sqrt{k/\gamma_k}) \epsilon_k \right] \geq 0.999$$
Singular values of a function

Singular value expansion of a square-integrable function $G : \Omega_1 \times \Omega_2 \rightarrow \mathbb{R}$:

$$G(x, y) = \sum \sigma_j u_j v_j^T$$

where $\Sigma = \text{diagonal, } U, V = \text{orthonormal columns}$.

At least formally:

$$\sigma_1 \geq \sigma_2 \geq \cdots > 0$$

$$(G)_{L^2(\Omega_1 \times \Omega_2)} = \sum_{j=k+1}^{\infty} \sigma_j^2 \leq \epsilon \|G\|_{L^2(\Omega_1 \times \Omega_2)}$$

$$\sqrt{\sum_{j=k}^{\infty} \sigma_j^2} > \epsilon \|G\|_{L^2(\Omega_1 \times \Omega_2)}$$

$$(x, y) \mapsto (\sigma_k, u_k, v_k)$$

$k = \text{rank}_\epsilon(G)$ means

[Schmidt 1907], [Weyl 1912], [Hammerstein 1923], [Smithies 1937]
Singular values of a function

Singular value expansion of a square-integrable function $G : \Omega_1 \times \Omega_2 \to \mathbb{R}$:

$$G(x, y) = \sum_{j=k+1}^{\infty} \sigma_j u_j \otimes v_j$$

$\Sigma = \text{diagonal}$, $U, V = \text{orthonormal columns}$

$\sigma_1 \geq \sigma_2 \geq \cdots > 0$

$\epsilon = \text{rank}_\epsilon(G)$

means

$$\sqrt{\sum_{j=k+1}^{\infty} \sigma_j^2} \leq \epsilon \|G\|_{L^2(\Omega_1 \times \Omega_2)}$$

$$\sqrt{\sum_{j=k}^{\infty} \sigma_j^2} > \epsilon \|G\|_{L^2(\Omega_1 \times \Omega_2)}$$

Also, see: [T. & Trefethen, 14], [Hashemi & Trefethen, 17], [Gorodetsky, Karaman, & Marzouk, 18]
Randomized SVD for Green’s functions [Boulle & T., 21]

We can learn kernel in a self-adjoint HS integral operator $f \mapsto \int_{\Omega} G(x, y)f(y)dy$:

**Randomized SVD for HS operators:**
We can learn kernel in a self-adjoint HS integral operator $f \mapsto \int_{\Omega} G(x, y)f(y)dy$:

<table>
<thead>
<tr>
<th>Randomized SVD for HS operators:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $\Omega \times (k + 5)$</td>
</tr>
</tbody>
</table>

$$Y = \begin{bmatrix} \vdots \end{bmatrix}$$

Cols are drawn from Gaussian process $GP(0, C)$
Randomized SVD for Green’s functions

We can learn kernel in a self-adjoint HS integral operator $f \mapsto \int_{\Omega} G(x, y)f(y)dy$:

Randomized SVD for HS operators:

1. $\Omega \times (k + 5)$

$Y = \text{Cols are drawn from Gaussian process } GP(0, C)$
Randomized SVD for Green’s functions

We can learn kernel in a self-adjoint HS integral operator $f \mapsto \int_\Omega G(x, y)f(y)dy$:

**Randomized SVD for HS operators:**

1. $\Omega \times (k + 5)$

$Y = \text{Cols are drawn from Gaussian process } GP(0,C)$
We can learn kernel in a self-adjoint HS integral operator $f \mapsto \int_{\Omega} G(x, y)f(y)dy$:

**Randomized SVD for HS operators:**

1. $\Omega \times (k + 5)$

$Y = \ldots$

Cols are drawn from Gaussian process $GP(0, C)$
We can learn kernel in a self-adjoint HS integral operator $f \mapsto \int_{\Omega} G(x, y)f(y)dy$:

**Randomized SVD for HS operators:**

1. $\Omega \times (k + 5)$

2. $Z_i(x) = \int_{\Omega} G(x, y)Y_i(y)dy$

Input-output data

Cols are drawn from Gaussian process $GP(0,C)$
We can learn kernel in a self-adjoint HS integral operator $f \mapsto \int_\Omega G(x, y)f(y)dy$:

**Randomized SVD for HS operators:**

1. $\Omega \times (k + 5)$
2. $Z_i(x) = \int_\Omega G(x, y)Y_i(y)dy$
3. $Q = \text{orth}(Z)$
   - orthonormal basis for $\text{col}(Z)$
   - \`$G_k = QQ^*G$``

Input-output data $Y = \ldots$

Cols are drawn from Gaussian process $GP(0, C)$
Randomized SVD for Green’s functions

We can learn kernel in a self-adjoint HS integral operator $f \mapsto \int_{\Omega} G(x, y)f(y)dy$:

**Randomized SVD for HS operators:**

1. $\Omega \times (k + 5)$
2. $Y = \begin{bmatrix} Z_1(x) = \int_{\Omega} G(x, y)Y_i(y)dy \end{bmatrix}$
3. $Q = \text{orth}(Z)$
   - **Input-output data**
   - **Cols are drawn from Gaussian process $GP(0, C)$**
   - Orthonormal basis for col$(Z)$
   - $\text{``}G_k = QQ^*G\text{''}$

**Theorem [Boulle & T., 2021]**

We can construct an approximation $G_k$ of $G$ from $k+5$ random input functions $f$ such that

$$\mathbb{P} \left[ \|G - G_k\|_{L^2} \leq \mathcal{O} \left( \sqrt{k^2/\gamma_k} \right) \epsilon_k \right] \geq 0.999$$
Randomized SVD for Green’s functions

We can learn kernel in a self-adjoint HS integral operator $f \mapsto \int_{\Omega} G(x, y)f(y)dy$:

**Randomized SVD for HS operators:**

1. $\Omega \times (k + 5)$
2. $Z_i(x) = \int_{\Omega} G(x, y)Y_i(y)dy$
   - Input-output data
3. $Q = \text{orth}(Z)$
   - Orthonormal basis for $\text{col}(Z)$
   - $G_k = QQ^*G$

- Cols are drawn from Gaussian process $GP(0, C)$

**Theorem [Boullé & T., 2021]**

We can construct an approximation $G_k$ of $G$ from $k+5$ random input functions $f$ such that

$$\mathbb{P} \left[ \|G - G_k\|_{L^2} \leq O \left( \frac{\sqrt{k^2 + \gamma_k}}{\epsilon_k} \right) \epsilon_k \right] \geq 0.999$$

**Problem:**

Green’s functions typically do not have rapidly decaying singular values.

$\epsilon_k$ decays very slowly with $k$
Smoothness implies low-rank

Suppose \( X_{ij} = G(x_i, y_j) \), where \( G : [-1, 1]^2 \to \mathbb{R} \) is a continuous function.

\[
G(\cdot, y) \text{ is } \nu\text{-times diff. with bounded variation: } \sigma_k(G) = O(k^{-\nu})
\]

\[
G(\cdot, y) \text{ is bounded analytic in neighborhood of } [-1, 1]: \sigma_k(G) = O(\rho^{-k})
\]

[Reade, 83], [Little & Reade, 84], [Ibragimov & Rjasanow, 09], [Khoromskij, 10], [Trefethen, 13]

\[
G(x, y) = \sum_{j=1}^{300} e^{-\gamma((x-x_j)^2+(y-y_j)^2)}
\]

Extensions to multivariate functions and tensors [Khoromskij, 10].
Aside: Covariance quality factor

**Theorem**

We can construct an approximation $G_k$ of $G$ from $k+5$ random input functions $f$ such that

$$\mathbb{P} \left[ \| G - G_k \|_{L^2} \leq \mathcal{O} \left( \sqrt{\frac{k^2}{\gamma_k}} \right) \epsilon_k \right] \geq 0.999$$

**Definition:**

$$\gamma_k = \frac{k}{(\lambda_1 \text{Tr}(C^{-1}))}$$

$$C_{ij} = \int_{D \times D} v_i(x)K(x,y)v_j(y) \, dx \, dy$$

where $v_i$ is the $i$th right singular vectors of $G$.

$$f \sim \mathcal{GP}(0, K)$$

where $K(x,y)$ is the covariance kernel

- $0 < \gamma_k \leq 1$
- We can impose prior knowledge on the covariance kernel
- Explicit bounds for the covariance quality factor are available
Regularity of Green’s functions
Green’s functions are low rank on separated blocks

One dimension

Very slow decaying singular values

Rapidly decaying singular values
Green’s functions are low rank on separated blocks

One dimension

Hierarchical structure

Level 2

Level 3

Level 4
Green's functions are low rank on separated blocks

One dimension

Three dimensions

Hierarchical structure

Low-rank structure on well separated domains.
[Bebendorf, Hackbusch, 2003]
Green’s functions are low rank on separated blocks

One dimension

Three dimensions

Hierarchical structure

Low-rank structure on well separated domains.
[Bebendorf, Hackbush, 2003]

Related approaches for matrices:
[Martinsson, 2008], [Lin, Lu, Ying, 2010],
[Martinsson, 2016]
Off-diagonal decay

Green’s function of the Laplace operator:

$$-\nabla^2 u = f$$

Green’s functions are smooth and decay off the diagonal. [Grüter, Widman, 1982]

$$G(x, y) \leq \frac{1}{||x-y||}$$

Hierarchical structure

Level 2

Level 3

Level 4
**Theorem [Boullé & T., 2021]**

There is a randomized algorithm that, for any $\epsilon > 0$, can construct an approximation $\tilde{G}$ of $G$ with $O(\epsilon^{-6} \log^4(1/\epsilon)/\Gamma_\epsilon)$ input-output pairs $(f, u)$ such that

$$||G - \tilde{G}||_{L^2(D \times D)} = O(\Gamma_\epsilon^{-3/4} \log^3(1/\epsilon) \epsilon),$$

with high probability.

---

**Randomized linear algebra**

- $g(x)$

**Low-rank structure**

- $h(y)$

**Off-diagonal decay**

$$G(x, y) \leq \frac{1}{||x-y||}$$
PDE learning with a rigorous “learning rate”

**Theorem** [Boullé & T., 2021]

There is a randomized algorithm that, for any $\epsilon > 0$, can construct an approximation $\tilde{G}$ of $G$ with $O(\log^4(1/\epsilon)/\Gamma_\epsilon)$ input-output pairs $(f, u)$ such that

$$
||G - \tilde{G}||_{L^2(D \times D)} = O(\Gamma_\epsilon^{-3/4} \log^3(1/\epsilon) \epsilon),
$$

with high probability.

**Randomized linear algebra**

**Low-rank structure**

**Off-diagonal decay**

$$
G(x, y) \leq \frac{1}{||x-y||}
$$
Learning Green’s functions in practice
Deep learning method

[Boullé, Earls, T., 2021]
Schrödinger equation, double well potential

\[ \mathcal{L} u = -(0.1)^2 \frac{d^2 u}{dx^2} + V(x)u, \quad u(\pm 3) = 0 \]
Advection-diffusion equation

Equation:
\[ \mathcal{L}u = 0.1 \frac{d^2 u}{dx^2} + (x \geq 0) \frac{du}{dx}, \quad u(-1) = 2, \quad u(1) = -1 \]
Recovering PDE properties from its Green’s function
Recovering PDE properties from its Green’s function

Question for the audience:
What PDE properties can we recover from a noisy /inaccurate Green’s function?
Summary

1. Theory for learning Green’s functions

\[ \mathcal{L}u = -\nabla \cdot (A(x)\nabla u) \]

2. Generalization of the randomized SVD

3. Deep learning approach

Python package

`pip install greenlearning`

https://github.com/NBoulle/greenlearning