

# Learning Axial Green Surrogates for Elliptic Problems with Variable Coefficients

Taeyoung Ha<sup>[0000-0002-9440-1918]</sup>,  
Junhong Jo<sup>[0000-0002-4283-3962]</sup>,  
Chang-Ock Lee<sup>[0000-0002-4704-9499]</sup>

## 1 Introduction

Consider a one-dimensional variable coefficient elliptic problem with Dirichlet boundary conditions

$$\begin{aligned} -\frac{d}{dx}\left(\kappa(x)\frac{du}{dx}(x)\right) &= f(x) \quad \text{for } x \in (a, b), \\ u(a) &= g_a, \quad u(b) = g_b, \end{aligned} \tag{1.1}$$

where  $\kappa(x) > 0$  denotes the coefficient function and  $f$  is a forcing term. The Green's function technique generates the solution of (1.1) as an integral operator representation acting on  $f$ , which is expressed as a convolution of a Green's kernel and  $f$ . For non-constant  $\kappa$ , the Green's kernel relies on the auxiliary solutions of the associated homogeneous differential equations, allowing few simple closed-form expression, making it difficult to use the classical representation directly [6, Section 3.2]. This paper mainly focuses on learning one-dimensional Green's kernels for variable coefficient elliptic operators. We discuss higher-dimensional problems to show how these 1D kernels, modeled by solving (1.1) along individual axes, can provide multidimensional solutions via axial decomposition without learning full high-dimensional kernels.

Recently there have been several data-driven surrogates to learn Green's function, thus eliminating the need for closed-form expressions. Boullé et al. [1] combined rational activation functions with symmetry constraints to approximate Green's func-

---

Taeyoung Ha  
National Institute for Mathematical Sciences, Daejeon 34047, Korea, e-mail: tha@nims.re.kr

Junhong Jo  
National Institute for Mathematical Sciences, Daejeon 34047, Korea, e-mail: jjhong0608@nims.re.kr

Chang-Ock Lee  
Department of Mathematical Sciences, KAIST, Daejeon 34141, Korea, e-mail: colee@kaist.edu

tions directly from paired samples of  $f$  and  $u$ . GF-Net [7] learns the Green's function using PINN for smoothed point sources. BI-GreenNet [3] is a neural network-based method combining fundamental solutions with boundary integral method to compute the Green's function. Neural Green's Functions [8] generate neural features which can be used to learn Green's function. These advances highlight the benefits of the Green's function representation, but they do not guarantee the flux jumps which are characteristic of Green's function, or rely on the closed form of the fundamental solution known only for specific equations. Motivated by these developments, we design a surrogate that (i) preserves the analytical structure of variable coefficient kernels and (ii) can be reused when coupling axial segments in higher dimensions without retraining an operator in the whole domain.

Our contributions are (i) GreenNet, which couples the closed-form Poisson Green's function and its antiderivative with a neural network so that boundary traces and flux jumps are enforced analytically while the network produces only smooth outputs, (ii) CouplingNet, which reuses those kernels to predict horizontal and vertical flux-divergence components and enforce the axial consistency constraint, and (iii) numerical evidence that embedding the analytical corrector yields lower errors than neural-only surrogates, even inside a two-dimensional DeepONet wrapper.

This paper is structured as follows. Section 2 reviews the one-dimensional Green representation, Section 3 details *GreenNet* and outlines *CouplingNet*, and Section 4 reports numerical experiments for the one-dimensional GreenNet learning task and its reuse inside a two-dimensional DeepONet. We conclude this paper by outlining how CouplingNet validation will extend our research.

## 2 One-Dimensional Green Representations

For problem (1.1), we define the Green's function  $G(x, \xi)$  as the solution to

$$\begin{aligned} -\frac{d}{dx}\left(\kappa(x)\frac{dG}{dx}(x, \xi)\right) &= \delta(x - \xi), \\ G(a, \xi) = G(b, \xi) &= 0, \end{aligned} \quad (2.1)$$

with the derivative jump  $\kappa(\xi)[\partial_x G(\xi^+, \xi) - \partial_x G(\xi^-, \xi)] = 1$ . Then the solution of (1.1) can be expressed as

$$u(x) = \int_a^b G(x, \xi)f(\xi) d\xi + g_a \phi_a(x) + g_b \phi_b(x), \quad (2.2)$$

where

$$S(x) = \int_a^x \frac{1}{\kappa(s)} ds, \quad \phi_a(x) = 1 - \frac{S(x)}{S(b)}, \quad \phi_b(x) = \frac{S(x)}{S(b)}.$$

Once  $G$  is known, evaluating (2.2) produces the solution for any source  $f$  without solving an additional boundary value problem.

The combination of (1.1), (2.1) and (2.2) shows that learning the kernel is equivalent to learning the solution operator. Consequently, accurately approximating  $G$  as an explicit integral kernel provides computational advantages of neural operator models.

### 3 Neural Approximation of One-Dimensional Kernels

For the sake of convenience, we consider (2.1) in the interval  $(0, 1)$ . GreenNet uses a fully connected network enhanced with analytical primitives that enforce boundary values and flux jump. Let  $G_P$  denote the Green's function for the Poisson equation on  $(0, 1)$  with homogeneous Dirichlet boundary conditions, then it is expressed as

$$G_P(x, \xi) = \begin{cases} x(1 - \xi), & x < \xi, \\ \xi(1 - x), & x \geq \xi, \end{cases}$$

with its antiderivative

$$H_P(x, \xi) = \int_0^x G_P(s, \xi) ds = \begin{cases} \frac{1}{2}x^2(1 - \xi), & x < \xi, \\ \frac{1}{2}\xi(2x - x^2 - \xi), & x \geq \xi. \end{cases}$$

We propose the GreenNet kernel

$$G(x, \xi) = \left[ x\xi(1 - \xi) \left( (1 - x)\widehat{G}_{\text{NN}}(x, \xi) - \frac{1}{2}x \right) + H_P(x, \xi) \right] \frac{\kappa'(\xi)}{\kappa(\xi)\kappa(x)} + \frac{1}{\kappa(x)} G_P(x, \xi), \quad (3.1)$$

where  $\widehat{G}_{\text{NN}}$  is the kernel from a feedforward network. The coefficients multiplied to  $G_P$  and  $H_P$  in (3.1) are selected to reproduce the singular feature generated by  $L = -\partial_x(\kappa(x)\partial_x)$ . Applying  $L$  to  $\frac{1}{\kappa(x)}G_P(x, \xi)$  delivers the Dirac delta, while the term proportional to  $\frac{\kappa'(\xi)}{\kappa(\xi)\kappa(x)}H_P(x, \xi)$  cancels the Heaviside contributions that arise when the operator  $L$  differentiates  $H_P$  twice. Consequently, the result of  $L$  acting on the residual component  $x(1 - x)\xi(1 - \xi)\widehat{G}_{\text{NN}}(x, \xi)$  is continuous, so the neural network produces only smooth outputs without distributional jumps, enforcing the boundary behavior and flux discontinuity exactly. The multiplicative gate  $x(1 - x)\xi(1 - \xi)$  forces the residual to vanish at the boundaries, and subtracting  $\frac{1}{2}x$  removes the nonzero trace of  $H_P$  at  $x = 1$ .

Given paired samples  $\{(f_i, u_i)\}_{i=1}^N$ , GreenNet minimizes the reconstruction loss

$$\mathcal{L}_{\text{Green}} = \frac{1}{N} \sum_{i=1}^N \left\| u_i(x) - \int_0^1 G_\theta(x, \xi) f_i(\xi) d\xi \right\|_{L^2(0,1)}^2,$$

where the  $L^2(0, 1)$  norm is taken with respect to the  $x$  variable. In practice,  $u_i$  and  $f_i$  denote their sampled values on a uniform grid  $\{x_m\}_{m=0}^M \subset (0, 1)$ . We approximate  $\int_0^1 G_\theta(\cdot, \xi) f_i(\xi) d\xi$  by the Simpson's rule on  $\{x_m\}$  and apply the same Simpson's rule to approximate the  $L^2(0, 1)$  norm on  $x$ .

## 4 Extension to Higher-Dimensional Problems

### 4.1 Challenges for multidimensional kernels

We consider a two-dimensional elliptic problem

$$-\nabla \cdot (\kappa(\mathbf{x}) \nabla u(\mathbf{x})) = f(\mathbf{x}), \quad \Omega \subset \mathbb{R}^2 \quad (4.1)$$

with Dirichlet boundary conditions and coefficient  $\kappa(\mathbf{x})$ , which allows a Green's function representation analogous to (2.2). However, the two-dimensional Green's kernel depends on the domain and the coefficient  $\kappa$ . Because their influences are complicated, the kernel cannot usually be written as an explicit analytical formula; instead, it must be computed numerically or approximated. Next, we review how to solve (4.1) by decomposing it into one-dimensional problems along the axis-parallel lines and combining the corresponding one-dimensional solutions.

### 4.2 Axial decomposition with CouplingNet

Now, we introduce the Axial Green's function method (AGM) to solve (4.1), which was first proposed in [2]. For each interior point of  $\Omega$ , we consider segments parallel to the axis passing through that point. Let  $(\bar{x}, \bar{y})$  be an interior point of  $\Omega$ . Along a horizontal segment  $X_{\bar{y}}$  and a vertical segment  $Y_{\bar{x}}$  the solution of (4.1) satisfies

$$\begin{aligned} -\partial_x (\kappa(x, \bar{y}) \partial_x u(x, \bar{y})) &= \varphi(x, \bar{y}), \\ -\partial_y (\kappa(\bar{x}, y) \partial_y u(\bar{x}, y)) &= \psi(\bar{x}, y), \end{aligned}$$

where  $\varphi$  and  $\psi$  satisfy  $\varphi(\bar{x}, \bar{y}) + \psi(\bar{x}, \bar{y}) = f(\bar{x}, \bar{y})$ . Then these axial components must satisfy the coupled integral relation

$$\int_{x_-}^{x_+} G_x(\bar{x}; \xi) \varphi(\xi, \bar{y}) d\xi + \int_{y_-}^{y_+} G_y(\bar{y}; \eta) \varphi(\bar{x}, \eta) d\eta = \int_{y_-}^{y_+} G_y(\bar{y}; \eta) f(\bar{x}, \eta) d\eta, \quad (4.2)$$

where  $G_x(\bar{x}; \xi)$  and  $G_y(\bar{y}; \eta)$  are 1-D Green's functions on the segments parallel to the  $x$ -axis and  $y$ -axis crossing  $(\bar{x}, \bar{y})$ , respectively. Discretizing this integral equation yields a linear system for  $\varphi$ . The matrix in this system is generally less sparse than stiffness matrices arising from standard finite difference or finite element discretizations, which is more expensive to solve.

In this work, we instead propose *CouplingNet* for (4.2) as a neural component that approximates  $\varphi$  and  $\psi$  by leveraging one-dimensional GreenNet kernels. CouplingNet is of a DeepONet-style branch–trunk form; the branch network takes sampled values of  $\kappa$  and  $f$  restricted to an axial segment, and the trunk network does the spatial coordinate  $(x, y)$  on that segment. The same network outputs a scalar flux-divergence component, denoted by  $\varphi_\theta(x, \bar{y})$  on the horizontal segment and  $\psi_\theta(\bar{x}, y)$  on the vertical segment. We train the network with a loss that enforces the axial flux-divergence balance relation  $\varphi + \psi = f$  on the set  $\mathcal{B}$ , where  $\mathcal{B}$  denotes the set of intersection points between the horizontal and vertical axial segments used in a training batch. We denote by  $G_\theta$  the GreenNet surrogate kernel used on each segment to reconstruct the corresponding one-dimensional solution. Note that  $G_\theta$  is not used in the CouplingNet forward pass; they are used to reconstruct the segment solutions  $u_\theta^{(x)}$  and  $u_\theta^{(y)}$  inside the consistency loss:

$$\mathcal{L}_{\text{FDB}} = \frac{1}{N} \sum_{j=1}^N \|\varphi_{\theta,j} + \psi_{\theta,j} - f_j\|_{L^2(\mathcal{B})}^2,$$

$$\mathcal{L}_{\text{consistency}} = \frac{1}{N} \sum_{j=1}^N \|u_{\theta,j}^{(x)} - u_{\theta,j}^{(y)}\|_{L^2(\mathcal{B})}^2,$$

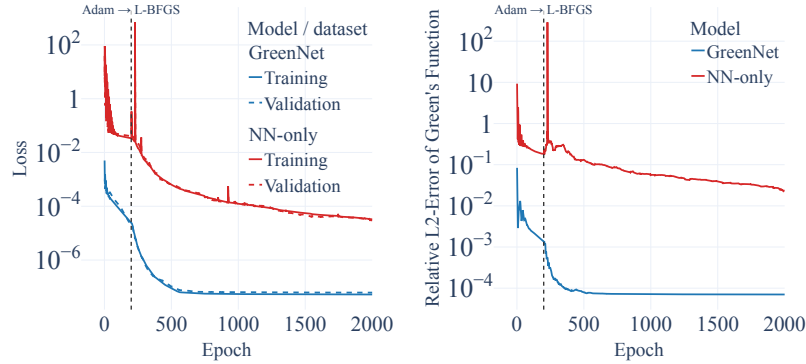
where  $u_{\theta,j}^{(x)}$  and  $u_{\theta,j}^{(y)}$  are solutions obtained using Green’s function  $G_\theta$  along the horizontal and vertical segments, respectively. The overall objective  $\mathcal{L} = \lambda_1 \mathcal{L}_{\text{FDB}} + \lambda_2 \mathcal{L}_{\text{consistency}}$  balances the two terms with positive weights  $\lambda_1$  and  $\lambda_2$ . As mentioned earlier, the main goal of this paper is to propose GreenNet to find accurate one-dimensional Green’s kernels, and to conceptually illustrate higher-dimensional extension through AGM-based axial decomposition with CouplingNet. We do not include numerical experiments verifying CouplingNet, and its empirical evaluation is left to future work.

## 5 Numerical Experiments

### 5.1 A 1-D problem with high contrast coefficient function

We first consider (1.1) with  $\kappa(x) = 5 \exp(x) - 4.9$  in the interval  $(0, 1)$  with the homogeneous Dirichlet boundary conditions ( $g_a = g_b = 0$ ). This choice yields a high-contrast coefficient, providing a challenging benchmark for GreenNet, where  $\widehat{G}_{\text{NN}}$  in (3.1) is the output of a simple fully connected network with two hidden layers of width 256 and rational activations in [1]. Training functions are sampled from the Gaussian process  $u \sim \mathcal{GP}(0, k_\ell)$  with a squared exponential kernel

$$k_\ell(x, x') = \exp\left(-\frac{(x - x')^2}{2\ell^2}\right),$$



**Fig. 1** Training and validation losses (left) and relative  $L^2$  kernel error (right) for GreenNet (blue) and the neural-network-only surrogate (red) for  $\kappa(x) = 5 \exp(x) - 4.9$ . The dashed vertical line marks the switch of optimizers from Adam to L-BFGS.

and each sample is a random sample of scale length  $\ell$  drawn uniformly from  $[0.05, 0.25)$ . Samples are evaluated on a uniform grid with mesh size  $1/256$ . We adjust them to satisfy  $u(0) = u(1) = 0$  by subtracting the affine interpolant matching the sampled endpoints. We extract 100 functions for training and 20 functions for validation. We do not prescribe  $f$  directly. Instead, analytic differentiation of each GP sample yields the right-hand side  $f = -\frac{\partial}{\partial x}(\kappa \frac{\partial}{\partial x} u)$ .

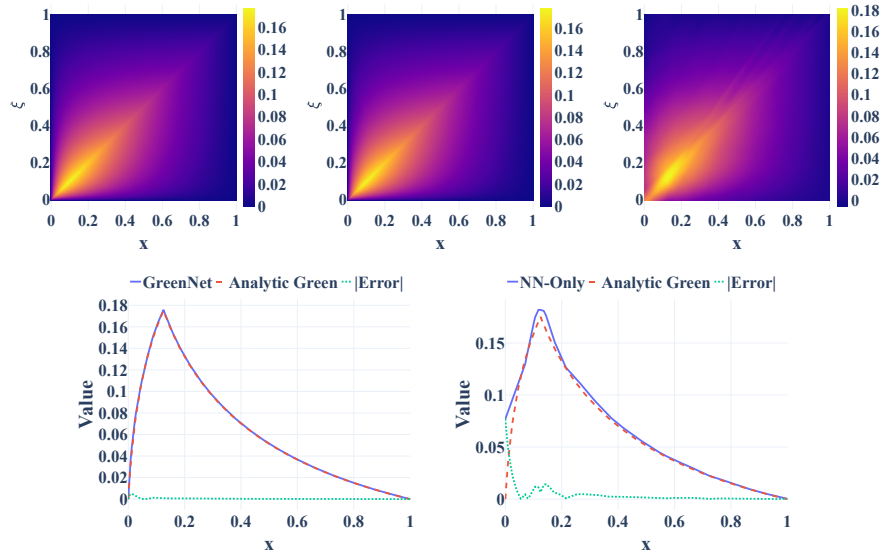
These draws therefore cover both highly oscillatory inputs (small  $\ell$ ) and slowly varying inputs (large  $\ell$ ) so that the learned kernels experience a wide range of forcing patterns [5].

We compare GreenNet with a neural-network-only multilayer perceptron that shares the same structure as  $\widehat{G}_{\text{NN}}$  and optimization schedule, but omits the analytical corrector. Figure 1 shows the loss histories and kernel errors; GreenNet achieves both metrics near  $10^{-8}$  and  $10^{-4}$ , respectively, while the vanilla network remains around  $10^{-4}$  and  $10^{-2}$ , respectively.

Figure 2 visualizes the learned kernels; GreenNet captures the singular feature and keeps absolute errors below  $5 \times 10^{-3}$ , while the neural-network-only surrogate smooths the interior and accumulates boundary errors near  $7 \times 10^{-2}$ .

## 5.2 Two-dimensional extension with DeepONet

To demonstrate the reuse of one-dimensional GreenNet kernels in higher dimensions, we consider a 2-D problem (4.1) in  $\Omega = (0, 1)^2$  with  $\kappa(x, y) = 5 \exp(x + y) - 4.9$ . We train DeepONet [4] with a branch–trunk output acting as  $\widehat{G}_{\text{NN}}$  in the GreenNet architecture and compare the results to those of the neural-network-only DeepONet obtained by removing the analytical corrector from GreenNet without changing the hyperparameters. Both the branch and trunk networks are simple fully connected

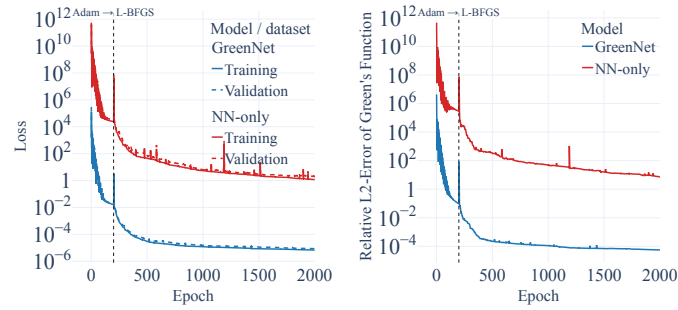


**Fig. 2** Green’s kernels for  $\kappa(x) = 5 \exp(x) - 4.9$ . Top row: exact Green’s function (left), GreenNet surrogate (center), and neural-network-only surrogate (right). Bottom row: vertical sections at  $\xi = 0.125$  comparing each surrogate with the exact Green’s function (left: GreenNet, right: neural-network-only).

networks with two hidden layers of width 256 and rational activations. On each axis-parallel line, we sample  $u$  from the one-dimensional GP, adjust it to satisfy the homogeneous boundary conditions as in the previous section, and compute  $f$  by analytic differentiation with the restricted coefficient  $\kappa$  on the axis-parallel line. We use 30 axis-parallel lines (mesh size 1/16), with 50 training and 10 validation functions per line. Figure 3 shows that embedding GreenNet inside DeepONet yields losses near  $10^{-5}$  and kernel errors near  $10^{-4}$ , while the neural-network-only model remains near 1, indicating that the analytical corrector still matters within the operator learning architecture.

## 6 Conclusion

In this paper, we proposed GreenNet, which combines an analytical corrector with a neural network to learn accurate one-dimensional Green’s kernels for variable coefficient elliptic operators. We validated GreenNet in a one-dimensional experiment and observed that kernel errors are more than two orders of magnitude lower than those of a neural-network-only surrogate. We also proposed CouplingNet for extension to higher-dimensional problems using 1-D Green’s kernels. We demonstrated how the learned one-dimensional kernels can be reused on multiple axis-parallel lines



**Fig. 3** Loss histories (left) and relative  $L^2$  kernel errors (right) for the two-dimensional DeepONet experiment with  $\kappa(x, y) = 5 \exp(x + y) - 4.9$ ; GreenNet (blue) is compared with the neural-network-only counterpart (red).

by embedding GreenNet in a two-dimensional DeepONet wrapper. The numerical validation and performance assessment of CouplingNet will be addressed in future work.

## References

1. Boullé, N., Earls, C.J., Townsend, A.: Data-driven discovery of Green's functions with human-understandable deep learning. *Sci. Rep.* **12**, 4824 (2022)
2. Kim, D.W., Park, S.K., Jun, S.: Axial Green's function method for multi-dimensional elliptic boundary value problems. *Int. J. Numer. Methods Eng.* **76**, 697–726 (2008)
3. Lin, G., Chen, F., Hu, P., Chen, X., Chen, J., Wang, J., Shi, Z.: Bi-GreenNet: Learning Green's functions by boundary integral network. *Commun. Math. Stat.* **11**, 103–129 (2023)
4. Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G.E.: Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **3**, 218–229 (2021)
5. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. MIT Press, Cambridge (2006)
6. Stakgold, I., Holst, M.J.: *Green's Functions and Boundary Value Problems*, 3 edn. John Wiley & Sons, Hoboken (2011)
7. Teng, Y., Zhang, X., Wang, Z., Ju, L.: Learning Green's functions of linear reaction–diffusion equations with application to fast numerical solver. *Proc. Mach. Learn. Res.* **190**, 1–16 (2022)
8. Yoo, S., Yeo, K., Hwang, J., Sung, M.: Neural Green's functions. *arXiv preprint arXiv:2511.01924* (2025)