LEARNING GREEN'S OPERATORS WITH SEPARABLE MODULATED KERNELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural Green's operators learn a reusable kernel that specifies how a unit source at one point contributes to a linear PDE's solution at another. Existing basis expansions combine spatial functions that remain fixed as the source point moves, which can require many terms to resolve local features. Motivated by fundamental solutions, we introduce the Separable Modulation Neural Operator (SMNO). We represent the Green's function as a sum of learned offset kernels, each multiplied by separate spatial weights at the source and evaluation points. The kernels depend only on the offset between these points and are shared across PDE instances. Both sets of weights are predicted from the domain shape and the coefficient field. On uniform grids, we apply the operator with fast Fourier transforms (FFTs) without constructing a dense kernel matrix. On Poisson problems with unseen 2D and 3D geometries, SMNO reduces the error on higher-frequency sources by 71–94% relative to basis-expansion Green's operators, and on Darcy flow with unseen coefficient fields it reduces the same error by 82% relative to FNO. A variant with a learned boundary map reduces NGF's error on the Mechanical Components Benchmark by 41%. On a 341^2 grid, building the operator and applying it to 64 sources takes 69 ms, versus 135 ms for FNO, 2.1 s for Transolver, and 10 s for a dense pairwise kernel.

1 INTRODUCTION

Classical solvers such as finite elements and finite differences (Hughes, 2003; LeVeque, 2007) mesh the domain and solve a new linear system whenever the geometry, coefficients, or source change, which makes design optimization (Bendsøe et al., 2004), uncertainty quantification (Smith, 2024), and inverse problems (Engl et al., 1996) expensive. Neural operators reduce this cost by learning the map from these inputs to the solution and evaluating it in one forward pass (Kovachki et al., 2023; Azizzadenesheli et al., 2024). The Fourier neural operator learns integral kernels in frequency space on regular grids (Li et al., 2020), DeepONet combines branch and trunk networks (Lu et al., 2021), and transformer-based operators handle irregular meshes with attention over points or learned slices (Cao, 2021; Hao et al., 2023; Alkin et al., 2024; Wu et al., 2024; Luo et al., 2025). We consider linear elliptic PDEs $\mathcal{L}_c u = f$ on a domain Ω, where c is a coefficient field, such as the permeability in Darcy flow, and f is the source term. Existing neural operators learn $(\Omega, c, f) \mapsto u$ directly and pass f through nonlinear layers together with Ω and c, so nothing in the architecture keeps them accurate on sources unlike those seen in training, even though the solution depends linearly on f.

Learned Green's operators exploit this linearity (Yoo et al., 2026; Melchers et al., 2026; Patel et al., 2024). For an instance (Ω, c) with zero boundary data, $u(x) = \int_\Omega G(x, y) f(y)\,\mathrm{d}y$, where the Green's function $G(x, y)$ is the response at the evaluation point x to a unit source at the source point y. These models predict G from the instance alone, so f is separated from the predicted G, which is reused for every source. They represent G by a finite basis expansion (BE), $\hat{G}(x, y) = \sum_{i=1}^{R} p_i(x) q_i(y)$. Moving y only reweights the same R functions p_i, so the discrete operator has rank at most R. Yet G is singular at $x = y$, and this peak moves with y (Figure 1); the basis must rebuild it at every source point. On 3D Poisson, BE with $R = 1024$ has relative error 0.064 on training-like sources but 0.94 on higher-frequency ones.

We keep the separation of f from G and add a second one, inside G. The free-space fundamental solution Φ of the Laplacian depends only on the offset $x - y$; a bounded domain adds a boundary

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

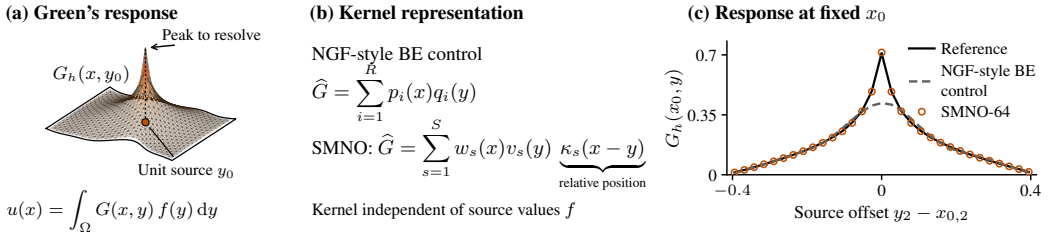

Figure 1: **Resolving the local peak of a Green's kernel.** (a) Reference Poisson kernel for a unit source at y_0. (b) Kernel parameterizations: NGF-style BE following Yoo et al. (2026), with independent factor heads, and SMNO. (c) At a fixed evaluation point x_0, SMNO with 64 kernels follows the reference peak more closely as the source point varies (Appendix D).

correction (Kress, 1989), and a smooth coefficient scales the singular part by $1/c(y)$ near the source. The singular shape is thus common to all source points and instances, while boundary and coefficient effects depend on where x and y lie. The Separable Modulation Neural Operator (SMNO) follows this split:

$$\hat{G}_{\Omega,c}(x,y) = \sum_{s=1}^{S} w_s(x;\Omega,c)\, v_s(y;\Omega,c)\, \kappa_s(x-y). \tag{1}$$

The learned offset kernels κ_s are shared across instances, so moving y translates each kernel instead of rebuilding it. An encoder reads the geometry and coefficients, but not f, and predicts the weights w_s and v_s at the evaluation and source points. Even one matrix with entries $\kappa_s(x_i - y_j)$ can have full rank, so S, unlike R, does not bound the rank. Each term applied to f is a weighted convolution $w_s[\kappa_s * (v_s f)]$, computed with fast Fourier transforms (FFTs) on uniform grids without forming the dense kernel matrix.

Contributions. (1) We propose SMNO, which shares offset kernels across source points and instances and places the instance dependence in weights at both points (Section 3). (2) We show that SMNO transfers to localized and higher-frequency sources on Poisson with held-out geometries and on variable-source Darcy, with higher-frequency errors of 0.055 on 3D Poisson and 0.088 on Darcy against 0.49 for FNO (Section 4). (3) We implement SMNO with FFTs on uniform grids. An extension that learns only a boundary correction to a known particular solution improves on NGF in all five categories of the Mechanical Components Benchmark (Sections 3.4 and 4.2).

2 Related work

Learned Green's operators. Fixed-domain methods learn Green's representations directly (Boullé et al., 2022; Teng et al., 2022) or in learned coordinates for nonlinear problems (Gin et al., 2021); others condition on boundary geometry (Tang et al., 2022). For linear PDEs, NGF and NGO use finite-basis kernels (Yoo et al., 2026; Melchers et al., 2026), while VarMiON maps sampled inputs into a finite output basis (Patel et al., 2024). These source-linear maps have rank at most the output-basis dimension for any instance. SMNO's term count does not bound its rank.

Neural operator architectures. FNO and Geo-FNO interleave spectral convolutions with nonlinear updates (Li et al., 2020; 2023); an Ewald-inspired point-cloud operator combines Fourier interactions with local geometric corrections (Han et al., 2026). BENO uses nonlinear branches for boundary values and interior sources (Wang et al., 2024), while attention models learn input-dependent interactions (Wu et al., 2024; Luo et al., 2025; Hao et al., 2023; Alkin et al., 2024). DeepONet has a finite output span but need not be linear in f (Lu et al., 2021). SMNO's attention encodes the instance independently of f, and its kernel then acts linearly on f.

Analytic information in learned operators. Ling et al. (2026) retain the known leading Fourier part of a parameterized Dirichlet-to-Neumann map (boundary values to normal derivatives) and learn a low-rank residual approximation. Gueyffier (2026) retains an analytic Stokes solution and learns a boundary-integral kernel; Sun et al. (2026) supply singularity estimates as inputs to a Green's-

function network. SMNO also uses distance features. Its primary model learns the full kernel; the optional extension adds a learned boundary-to-interior correction to a known particular solution.

Classical structure. Separable symbols give spatially weighted convolutions (Bao & Symes, 1996; Demanet & Ying, 2011). SMNO learns their offset kernels and makes the weights two-sided and instance-dependent (Section 3.2). Hierarchical methods compress well-separated matrix blocks (Bebendorf & Hackbusch, 2003; Boullé & Townsend, 2023); reduced-basis methods instead restrict the global output space (Hesthaven et al., 2016).

3 Separable Modulation Neural Operator

3.1 Setting and notation

Problem family. An instance $(\Omega, c) \in \mathcal{D}$ comprises a bounded domain $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, and a coefficient field c. We solve

$$\mathcal{L}_c u = f \quad \text{in } \Omega, \qquad u = g \quad \text{on } \partial\Omega, \tag{2}$$

where f is the source, g the boundary data, and $\mathcal{L}_c$ a second-order elliptic operator, with $\mathcal{L}_c u = -\nabla \cdot (c\nabla u)$ for Darcy flow. We vary instances and sources, with $g = 0$ except in the Mechanical Components Benchmark (MCB; Table 2(b)).

Solution map. For a fixed instance, linearity separates the effects of boundary data and interior sources:

$$u = P\, g + G\, f, \qquad (Gf)(x) = \int_\Omega G(x, y)\, f(y)\, \mathrm{d}y. \tag{3}$$

Here P maps boundary values to the solution with $f = 0$, while G maps f to the solution with $g = 0$; $G(x, y)$ is the Green's function (Kress, 1989). The main model predicts G from the instance and applies it linearly to f. Hats denote predictions, and instance dependence is suppressed below. The optional boundary-map extension handles nonzero g (Eq. (10)).

3.2 Background: structure of the Green's function

Write $r = x - y$ for the offset between an evaluation point x and a source point y. For the Laplacian, the fundamental solution Φ depends on the offset alone and is singular at $r = 0$,

$$\Phi(r) = -\tfrac{1}{2\pi} \log \|r\| \quad (d = 2), \qquad \Phi(r) = \tfrac{1}{4\pi \|r\|} \quad (d = 3), \tag{4}$$

so that $-\Delta_x \Phi(x - y) = \delta(x - y)$ (Kress, 1989). On a bounded domain, its Green's function is $\Phi(x - y)$ plus an interior harmonic correction that enforces the boundary condition. Treating a smooth positive coefficient as locally constant near y gives a local approximation, or parametrix, of the form

$$G(x, y) = \Phi(x - y)\, \mu_c(x, y) + \varrho_c(x, y), \tag{5}$$

where $\mu_c(x, y) \to 1/c(y)$ as $x \to y$ for Darcy flow, and ϱ_c carries the remaining coefficient and boundary corrections. For discontinuous c, the parametrix holds locally away from interfaces and need not have a globally smooth remainder. It suggests treating the offset $x - y$ explicitly and learning how the instance modulates it.

Classical symbol methods separate offset kernels from spatial factors (Bao & Symes, 1996; Demanet & Ying, 2011). The *symbol* $\sigma_G(x, \xi)$ of the Green's solution operator G is its spatially varying frequency response: it describes how a frequency ξ contributes at the evaluation point x. Suppose that, in a local Fourier representation, the symbol admits the separable approximation

$$\sigma_G(x, \xi) \approx \sum_{s=1}^{S} \alpha_s(x) b_s(\xi).$$

Let κ_s be the inverse Fourier transform of b_s. Multiplying the source spectrum by b_s is equivalent to convolving the source with κ_s. Thus $(Gf)(x) \approx \sum_s \alpha_s(x)(\kappa_s * f)(x)$, whose integral kernel is

$$G(x, y) \approx \sum_{s=1}^{S} \alpha_s(x)\, \kappa_s(x - y). \tag{6}$$

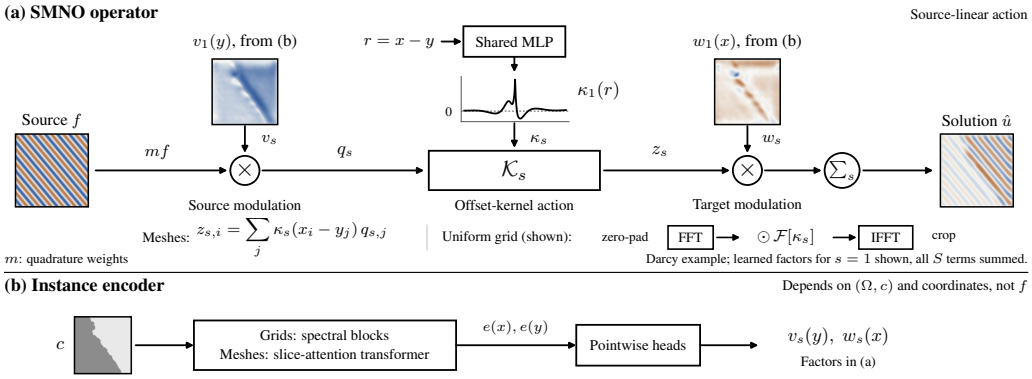

Figure 2: **SMNO forward computation.** (a) Source modulation, offset-kernel action, and target modulation, with $(\mathcal{K}_s)_{ij} = \kappa_s(x_i - y_j)$. (b) Source-independent instance encoder. Meshes use dense quadrature. Uniform grids use FFTs. Thumbnails show a Darcy example: the coefficient c, the learned factors for $s = 1$, and the predicted solution for a held-out wave source, each on its own color scale (Appendix C).

Each term separates position x from frequency ξ and keeps x and y coupled. We retain the spatially weighted convolution structure while learning the offset kernels in physical space and the spatial factors from source–solution pairs.

3.3 Separable modulation kernels

Equation (6) modulates each convolution at the evaluation point. We extend this representation by introducing an instance-dependent modulation at the source point as well. The two-point structure in Eq. (5) motivates this choice: a separable approximation $\mu_c(x, y) \approx \sum_s w_s(x) v_s(y)$ produces terms of the form $w_s(x) v_s(y) \Phi(x - y)$. Keeping this two-sided form while learning the offset kernels κ_s gives the parameterization

$$\hat{G}(x, y) = \sum_{s=1}^{S} w_s(x)\, v_s(y)\, \kappa_s(x - y), \tag{7}$$

Here, "separable modulation" refers to $w_s(x) v_s(y)$. The weights are computed from the instance by an encoder (Section 3.4) and the κ_s are learned functions of the offset. Setting $v_s \equiv 1$ and $w_s = \alpha_s$ recovers Eq. (6). This modeling extension is not an exact finite-term decomposition implied by Eq. (5). The learned expansion approximates the full Green's function, including its remaining correction.

Offset kernels. At a fixed discretization scale h, the κ_s depend only on the offset r. On the Darcy grid, one multilayer perceptron (MLP) returns all S values:

$$\kappa_s(r; h) = \mathrm{MLP}\big(r,\ \|r\|,\ \log(\|r\| + h/2)\big)_s. \tag{8}$$

On Darcy, $h = 1/85$ enters through the logarithm. Poisson mesh models also supply h as a separate input: the nominal grid spacing after coordinate normalization, fixed within each instance. The distance features capture variation near $r = 0$, and the components of r encode direction. The regularized logarithm is finite at zero, so the network does not impose Φ's analytic singularity. Network parameters are shared across point pairs and instances. We omit h below.

Modulations. The w_s and v_s are read-outs of per-point encoder features at the evaluation and source points. Both sets of weights depend on geometry and coefficients.

Number of terms. S controls the size of the modulation expansion. It does not bound the rank of the discrete operator: even one matrix with entries $\kappa_s(x_i - y_j)$ can have full rank.

3.4 NETWORK ARCHITECTURE

Figure 2 shows how an instance encoder and a shared offset network supply the factors for the source-linear map $(\Omega, c, f) \mapsto \hat{u}$.

Instance encoder. The encoder returns point features $e(x), e(y)$ independent of f and g. On the fixed Darcy square, spectral blocks read coefficients and coordinates. On the constant-coefficient Poisson meshes, a Transolver-style encoder (Wu et al., 2024) reads boundary coordinates, normals, and interior anchors; screened Poisson adds a scalar operator token. Weighted pooling compresses these points into slice tokens, a transformer exchanges information between tokens, and point queries recover features by cross-attention.

Kernel head. Two pointwise heads map $e(x), e(y)$ to $w_s(x), v_s(y)$; a shared S-output MLP evaluates Eq. (8). Their products form the kernel in Eq. (7). Layer sizes and grid/mesh configurations are in Appendix A.1.

Applying the kernel. For source samples $f_j = f(y_j)$, quadrature weights m_j, and evaluation points x_i, the predicted solution is

$$\hat{u}(x_i) = \sum_{s=1}^{S} w_s(x_i) \sum_j \kappa_s(x_i - y_j)\, v_s(y_j)\, m_j f_j. \tag{9}$$

Each term weights the source by v_s, applies the kernel, then weights the output by w_s. Linearity holds since no factor depends on f. It does not imply accuracy on unseen sources. Uniform grids use zero-padded FFTs for the convolution; mesh experiments evaluate the sum directly (Appendix A.2).

Analytic particular-solution extension. When a particular solution is known, one may instead learn only a boundary correction. For Poisson $-\Delta$, the free-space potential $\mathcal{N}f$ gives

$$Gf = \mathcal{N}f - P\big[(\mathcal{N}f)|_{\partial\Omega}\big], \qquad (\mathcal{N}f)(x) = \int_\Omega \Phi(x-y) f(y)\, \mathrm{d}y. \tag{10}$$

Here P maps boundary values to a harmonic solution (Kakutani, 1944). The free-space term is the analytic $S = 1$ case of Eq. (7) with $w_1 = v_1 = 1$ and $\kappa_1 = \Phi$. More generally, any particular solution u_p satisfying $-\Delta u_p = f$ can be corrected as $u = u_p + P[g - u_p|_{\partial\Omega}]$. For MCB, a box Poisson solve supplies u_p, and we learn only $\hat{P}$. A separate geometry network and softplus give each pair of evaluation point i and boundary node k a score $a_{ik} > 0$, normalized as $\hat{P}_{ik} = a_{ik} / \sum_\ell a_{i\ell}$. The prediction $\hat{u} = u_p + \hat{P}[g - u_p|_{\partial\Omega}]$ adds a weighted sum of boundary residuals to the particular solution (Appendix F).

4 EXPERIMENTS

We test transfer across PDE instances and source distributions, measure accuracy and computational cost, and use ablations to examine kernel structure and source coverage.

4.1 SETUP

Training and evaluation. We report relative L^2 solution error. The main Poisson and Darcy comparisons use (f, u) pairs with relative squared solution loss and no Green's-kernel supervision. Encoder architectures are held fixed within each SMNO size sweep. Protocols and implementation details are in Appendices B, A.1, C, and D.

Baselines. Our basis-expansion control (BE-R) adapts NGF's point-feature kernel (Yoo et al., 2026) with independent factors:

$$\hat{G}^{\mathrm{BE}}(x, y) = \sum_{i=1}^{R} p_i(x)\, q_i(y). \tag{11}$$

It is linear in the source and has rank at most R. We also compare FNO (Li et al., 2020), DeepONet (Lu et al., 2021), Transolver (Wu et al., 2024), and Transolver++ (Luo et al., 2025). Mesh baselines include an NGO-style basis-matrix control (Melchers et al., 2026), a source-conditioned boundary-integral network (BIN) in the style of Lin et al. (2021), and a field-prediction control that shares our

(a) Poisson: matched solution-only protocol, seed 0

	2D channels		3D ducts (ny24)	
Model	Fourier	Band	Fourier	Band
SMNO-64	0.0042	0.0155	0.0198	**0.0550**
SMNO-16	**0.0038**	**0.0150**	**0.0160**	0.0567
BE-64 Yoo et al. (2026)	0.0159	0.1454	0.1306	0.9965
BE-256 Yoo et al. (2026)	0.0116	0.0587	0.0690	0.9719
BE-1024 Yoo et al. (2026)	0.0112	0.0539	0.0643	0.9405
NGO-style Melchers et al. (2026)	0.0146	0.1460	0.1282	0.9963
BIN Lin et al. (2021)	0.0164	0.0577	0.0324	0.0932
DeepONet Lu et al. (2021)	0.0999	1.0490	1.0014	1.0121
Transolver++ Luo et al. (2025)	0.3455	1.3856	0.3441	1.4835

(b) Variable-source Darcy: held-out coefficients

Model	Seen	Localized	Band
SMNO-64	0.0134	**0.0119**	**0.0877**
SMNO-16	0.0163	0.0146	0.1279
BE-512 Yoo et al. (2026)	0.0208	0.0450	0.8454
BE-64 Yoo et al. (2026)	0.1054	0.0750	0.9057
FNO Li et al. (2020)	**0.0116**	0.0768	0.4899
Transolver Wu et al. (2024)	0.0188	0.2708	0.6050
Transolver++ Luo et al. (2025)	0.0220	0.4066	0.5991
DeepONet Lu et al. (2021)	0.0600	0.1365	0.9762

Table 1: **Core PDE comparisons.** Relative L^2 error: (a) Pooled median on held-out Poisson geometries, matched solution-only training, seed 0 (2D ny32, 3D ny24, both used in training). (b) Mean over test sources and 200 Darcy coefficient fields; three seeds for SMNO-16/64, BE-512, and FNO, one for the other models. Fourier/Seen follow training distributions; Band/Localized are held out. Bold marks the lowest displayed point estimate and does not imply significance. The unfactorized-kernel control is reported in Table 5. Full results, seed variability, and protocols are in Appendices B and C.

geometry encoder and reads the source as input features. Baseline implementations and budgets are in Appendices B, C, and D. An unfactorized-kernel architectural control is examined in Section 4.4.

4.2 Poisson on held-out geometries

Task. We solve $-\Delta u = f$ on random 2D channels (1,000 training geometries at ny32) and 3D ducts (800 at ny16, 300 of them also at ny24), where ny counts cells across the height. Test shapes are new draws from the same families. We train on both 3D grids and use ny24 for the main comparison. Training mixes low-frequency sinusoidal (*Fourier*), Gaussian, polynomial, and white-noise sources. Held-out categories are higher-frequency sinusoids (*Band*), narrow spatial peaks (*Localized*), and oblique sinusoids (*Wave*). They are distribution shifts despite broadband white noise in training. Table 1(a) shows Fourier and Band medians pooled over shape–source pairs. All models share seed 0, source and query streams, and solution loss. We evaluate final checkpoints after 9.6M pair exposures. Full six-family results, budgets, and geometry-resampling intervals are in Appendix B.

Geometry and source transfer. Both SMNO variants reduce all six source-family medians relative to the tested BE, NGO-style, BIN, DeepONet, and Transolver++ models at all three grids. On 3D Band at ny24, SMNO-64 gives 0.0550 versus 0.9405 for BE-1024. Increasing the term count from 16 to 64 does not consistently improve Poisson accuracy. These are single-seed comparisons under the stated training rules. Figure 3(a,b) shows a fixed test channel and duct.

Screened Poisson. To test a varying screening parameter, we solve $-\Delta u + \sigma^2 u = f$ with one σ per 2D channel, sampled log-uniformly from $[0.5, 8]$ in both training and testing. All models are trained at ny32. On held-out channels at this resolution, SMNO-64 has the lowest displayed errors, including Band error 0.0184 versus 0.0672 for BE-256 (Table 2(a) and Figure 3(c)). As in

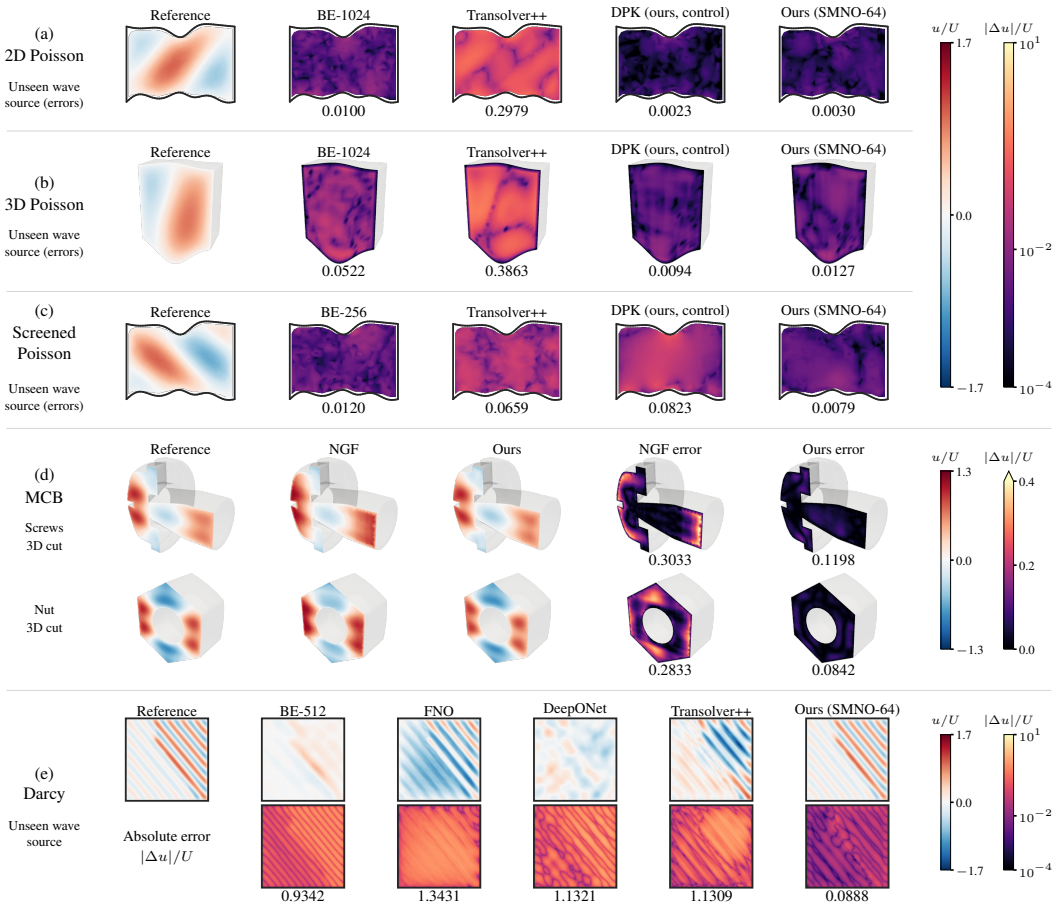

Figure 3: **Qualitative comparison across five tasks.** Rows (a) 2D Poisson, (b) 3D Poisson on a duct, and (c) screened Poisson show the reference and each model's absolute error; (d) MCB, with a screw and a nut, and (e) Darcy also show predictions. Ours is SMNO-64, except the boundary-map variant in (d). Sources in (a,b,c,e) are held-out Wave sources; (d) uses unseen shapes and an unseen problem. 3D shapes are cut by a plane, with the rest drawn as a translucent surface. Fields are scaled by each task's $U = \max |u_{\mathrm{GT}}|$; (d) uses linear scales and the other rows share signed-field and log-error scales. Labels give relative L^2 errors on the evaluation nodes. Panels (d,e) show selected gains rather than averages. The nut also appears in Figure 5. Appendices D, F, and C cover the selection.

the Poisson task above, Band and Localized are held out. Appendix D.1 gives the protocol and full ny32 results.

Poisson on mechanical components. The Poisson benchmark of Yoo et al. (2026) on Mechanical Components Benchmark (MCB) shapes (Kim et al., 2020) varies interior sources and nonzero boundary data on five CAD categories. We evaluate the particular-solution extension, which learns only the boundary map. On jointly unseen shapes and problems, it improves over the published NGF and Transolver results in every category on the same split, with mean error 0.141 versus 0.241 for NGF (Table 2(b)). This comparison evaluates the complete extension; it does not isolate the learned map's contribution from that of the supplied particular solution. The learned boundary map has 0.027M active parameters (0.039M for motor). Figure 3(d) shows two held-out parts, and Figure 5 in Appendix F shows one part per category with the protocol. At one fixed problem, the boundary map has lower nodal error than NGF on 41 of 50 unseen shapes we reviewed (ten per category).

(a) Screened Poisson
Held-out channels

Model	Seen	Band	Localized
SMNO-64	**0.0076**	**0.0184**	**0.0065**
BE-256	0.0125	0.0672	0.0172
Yoo et al. (2026)			
Transolver	0.0204	0.4954	0.2193
Wu et al. (2024)			
Transolver++	0.0454	0.2914	0.1496
Luo et al. (2025)			

(b) MCB
Boundary-map extension

Category	Transolver Wu et al. (2024)	NGF Yoo et al. (2026)	Boundary map (ours)
fitting	0.180	0.160	**0.118**
gear	0.281	0.243	**0.126**
nut	0.320	0.275	**0.126**
screws	0.221	0.189	**0.082**
motor	0.407	0.338	**0.255**
mean	0.282	0.241	**0.141**

Table 2: **Screened Poisson and mechanical components.** Relative L^2 errors: (a) Screened Poisson on 100 held-out ny32 geometries (seed 0). Band and Localized medians pool 64 sources per geometry; Seen averages the Fourier, Gaussian, and polynomial medians. (b) The boundary-map variant on MCB, reporting mean error on the unseen-shape, unseen-problem split. Ours averages three seeds per category except fitting (one); baseline values are from the published benchmark, without retraining. Bold marks the best displayed value within each comparison. Full results and protocols are in Appendices D.1 and F.

4.3 Variable-source Darcy

Task. We use the public Darcy coefficients, split, 85^2 grid, and Dirichlet condition, but vary f in $-\nabla \cdot (c\nabla u) = f$ on $[0,1]^2$. Labels are computed with the native 421^2 conservative operator and then downsampled. We train on random combinations of an 81-dimensional low-frequency Fourier bank. Seen uses new combinations from this bank; Band uses higher frequencies and Localized uses narrow spatial peaks. All models see one million coefficient–source pairs, use Seen-only validation selection, and are evaluated in single precision (FP32) on 200 test coefficients. Pairwise kernels use 32 queries per field, while BE and field models use the full field. Thus pair counts match, but point counts differ. Direct SMNO ablations match both.

Transfer beyond the training source span. SMNO-64 has the lowest Localized and Band errors while remaining competitive with FNO, the best Seen model (Table 1(b)). Its Band error is 0.0877, versus 0.4899 for FNO and 0.8454 for BE-512. Results on additional source families, an NGF-style symmetric control, and basis-matrix controls support this comparison (Appendix C). Figure 3(e) shows a wave-source example.

Computational cost. On the 85^2 grid ($N = 7225$ points) and one H100, SMNO-64 takes 2.8 ms to build the operator for one coefficient field and 4.0 ms to apply it to 64 sources with FFTs, or 6.8 ms in total versus 11.2 ms for FNO. SMNO-64 stays faster than FNO on every tested grid up to 421^2. The setup can be reused for further sources. A cached dense kernel applies faster on this small grid, but SMNO-16 overtakes it from 171^2 and SMNO-64 by 341^2, where the dense matrix takes 10 s to build and 54 GB to store; SMNO also stores $O(SN)$ rather than N^2 values (Table 21).

4.4 Ablation and mechanism analysis

Kernel factorization. We replace the separable head with a dense pairwise kernel (DPK), keeping the encoder architecture, offset features, and source linearity fixed. SMNO approaches DPK's Poisson accuracy: three-seed mean Band errors are 1.44% versus 1.39% in 2D and 5.24% versus 4.63% in 3D (SMNO versus DPK). Its Band errors are lower on Screened Poisson and Darcy (Table 5). SMNO is also faster to set up: on the 85^2 Darcy grid, building the operator for one coefficient field takes 2.8 ms, versus 47.5 ms for DPK, whose build step forms all N^2 kernel entries. Once built, DPK's dense product applies to 64 sources in 0.16 ms, versus 4.0 ms for SMNO-64 (Table 21). Yet factorization enables grid FFTs and $O(SN)$ storage (Appendices A.2 and G).

Direct SMNO ablations. We test distance features and source-side modulation using the same Darcy encoder architecture, $S = 64$, training and sampling rules, and three seeds per variant (Table 3). Removing log distance, removing both radial features (offset only), or setting $v_s = 1$ worsens Band transfer. All three error increases have positive paired 95% intervals under resampling of test coefficients, given the trained models and training environments (Appendix B). Offset only has a

Variant	Seen	Localized	Wave	Band
Full SMNO-64	0.0134 ± 0.0002	0.0119 ± 0.0003	0.0718 ± 0.0032	0.0877 ± 0.0023
No log	0.0137 ± 0.0005	0.0135 ± 0.0003	0.1005 ± 0.0107	0.1857 ± 0.0243
Offset only	0.0130 ± 0.0003	0.0190 ± 0.0011	0.2692 ± 0.0256	0.4697 ± 0.0523
One-sided ($v_s = 1$)	0.0341 ± 0.0003	0.0260 ± 0.0003	0.2060 ± 0.0032	0.2133 ± 0.0046

Table 3: **Direct SMNO-64 ablations on Darcy.** Mean relative L^2 error $\pm$ sample SD over three seeds. Each variant uses one million training pairs, 32 queries per field, and Seen-only selection. Band is the prespecified primary comparison. Full uses the Table 1(b) runs. Training GPUs and active parameter counts differ (Appendix B).

slightly lower Seen error. Because distance features are functions of the offset, these ablations change parameterization, not available information. They do not isolate optimization mechanisms or show that the exact singularity is preserved or that each part is always needed.

Kernel parameterization and size. Additional controls test restrictions on the offset kernels. With the same encoder architecture and $S = 16$, isotropic or Fourier-parameterized offset kernels worsen Darcy Band error. A control with a fixed fundamental solution is also worse. Increasing S gives diminishing gains on Darcy, and a low-rank correction helps modestly. Offset inputs help DPK and attention controls on Darcy, but removing DPK offsets changes little on broadband Poisson. These controls use seed 0 (Appendices C.2 and D).

Representation and source coverage. We distinguish limits of the output basis from insufficient coverage by training sources. On Darcy, replacing half the training pairs with white-noise sources reduces BE-512's Band error from 0.8458 to 0.4425 at seed 0, with rank and target discretization unchanged. Thus, limited rank alone does not explain its transfer error. For a fixed instance, solution pairs constrain the operator on the span of observed sources. Beyond it, predictions depend on structural assumptions. Poisson training uses white noise, so a held-out category can emphasize weakly excited directions without necessarily leaving the observed training span (Boullé & Townsend, 2023).

Exact truncation oracles apply truncated reference operators without fitting a model, measuring the error of that particular truncation. They are not universal lower bounds under source-relative error. The Darcy oracle uses an 85^2 operator, whereas learned models use downsampled 421^2 targets, so the error gap cannot separate representation from source-coverage error (Appendices C.1 and E). Neither a superposition penalty nor linearizing FNO around the zero source recovers the transfer accuracy of offset kernels, so linearity alone is not enough (Appendix C.2).

5 Conclusion

SMNO combines shared offset kernels with two-sided spatial modulation in a source-linear Green's operator. Poisson results support it as a structured alternative to the tested basis expansions. Three-seed Darcy ablations support its distance features and two-sided modulation for source transfer. On grids, its convolutions use FFTs, and new sources are handled without rebuilding the operator. When a particular solution is known, a separate extension learns only a boundary correction.

Limitations and future work. We target linear elliptic problems with Dirichlet boundary conditions whose Green's functions are positive and non-oscillatory; extending the offset kernels to oscillatory Green's functions such as high-frequency Helmholtz, to Neumann or Robin conditions, and to nonlinear operators requires further mathematical analysis. On meshes the pairwise sum is currently evaluated directly at quadratic cost; future work could bring the grid speedups to large 3D meshes with hierarchical or nonuniform FFT summation. Finally, like other supervised neural operators, SMNO learns from source–solution pairs produced by a classical solver; training it from PDE residuals instead would remove this data-generation cost.

Ethics statement

This work uses only synthetic PDE data and public benchmark data, and involves no human subjects or personal data.

Reproducibility statement

All models use fixed final checkpoints or checkpoints selected on validation data from the training source families; no test data enter model selection. Appendices A–G give the architectures, data generation, training and selection rules, seeds, and per-run results. Code, checkpoints, and the scripts for every table and figure will be released upon publication.

AI use statement

We used a large language model assistant to aid and polish the writing: grammar, phrasing, and readability edits; consistency of terminology and notation; and checks of the manuscript's compliance with the conference template and of the reference list. The authors reviewed every change and take full responsibility for the content of the manuscript.

References

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems*, 37:25152–25194, 2024.

Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5):320–328, 2024.

Gang Bao and William W Symes. Computation of pseudo-differential operators. *SIAM Journal on Scientific Computing*, 17(2):416–429, 1996.

Mario Bebendorf and Wolfgang Hackbusch. Existence of $\mathcal{H}$-matrix approximants to the inverse fe-matrix of elliptic operators with L^∞-coefficients. *Numerische Mathematik*, 95(1):1–28, 2003.

Martin P Bendsøe, Ole Sigmund, et al. *Topology optimization: theory, methods, and applications*, volume 2. Springer Berlin, 2004.

Nicolas Boullé and Alex Townsend. Learning elliptic partial differential equations with randomized linear algebra. *Foundations of Computational Mathematics*, 23(2):709–739, 2023.

Nicolas Boullé, Christopher J Earls, and Alex Townsend. Data-driven discovery of green's functions with human-understandable deep learning. *Scientific reports*, 12(1):4824, 2022.

Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.

Laurent Demanet and Lexing Ying. Discrete symbol calculus. *SIAM review*, 53(1):71–104, 2011.

Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

Craig R Gin, Daniel E Shea, Steven L Brunton, and J Nathan Kutz. Deepgreen: deep learning of green's functions for nonlinear boundary value problems. *Scientific reports*, 11(1):21614, 2021.

Denis Gueyffier. Solver exactness, learned flexibility: Equivariant boundary-correction operators for stokes flow. *arXiv preprint arXiv:2606.25075*, 2026.

Mingyu Han, Daniel Zhengyu Huang, Yuhan Wang, Yanshu Zhang, and Jiayi Zhou. Geometric generalization of neural operators from kernel integral perspective. *arXiv preprint arXiv:2602.01498*, 2026.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International conference on machine learning*, pp. 12556–12569. PMLR, 2023.

Jan S. Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2016.

Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2003.

Shizuo Kakutani. 143. two-dimensional brownian motion and harmonic functions. *Proceedings of the Imperial Academy*, 20(10):706–714, 1944.

Sangpil Kim, Hyung-gun Chi, Xiao Hu, Qixing Huang, and Karthik Ramani. A large-scale annotated mechanical components benchmark for classification and retrieval tasks with deep neural networks. In *European conference on computer vision*, pp. 175–191. Springer, 2020.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Rainer Kress. *Linear integral equations*, volume 82. Springer, 1989.

Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.

Guochang Lin, Pipi Hu, Fukai Chen, Xiang Chen, Junqing Chen, Jun Wang, and Zuoqiang Shi. Binet: learning to solve partial differential equations with boundary integral networks. *arXiv preprint arXiv:2110.00352*, 2021.

Shuo Ling, Wenjun Ying, and Han Zhou. Principal-part decomposition for neural operator learning of dirichlet-to-neumann maps. *arXiv preprint arXiv:2606.25952*, 2026.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Huakun Luo, Haixu Wu, Hang Zhou, Lanxiang Xing, Yichen Di, Jianmin Wang, and Mingsheng Long. Transolver++: An accurate neural solver for pdes on million-scale geometries. *arXiv preprint arXiv:2502.02414*, 2025.

HA Melchers, JHM Prins, and Michael RA Abdelmalik. Neural green's operators for parametric partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 455: 118893, 2026.

Dhruv Patel, Deep Ray, Michael RA Abdelmalik, Thomas JR Hughes, and Assad A Oberai. Variationally mimetic operator networks. *Computer Methods in Applied Mechanics and Engineering*, 419:116536, 2024.

Ralph C Smith. *Uncertainty quantification: theory, implementation, and applications*. SIAM, 2024.

Qi Sun, Shengyan Li, Bowen Zheng, Lili Ju, and Xuejun Xu. Learning singularity-encoded green's functions with application to iterative methods. *Journal of Computational Physics*, pp. 114894, 2026.

Jingwei Tang, Vinicius C Azevedo, Guillaume Cordonnier, and Barbara Solenthaler. Neural green's function for laplacian systems. *Computers & Graphics*, 107:186–196, 2022.

Yuankai Teng, Xiaoping Zhang, Zhu Wang, and Lili Ju. Learning green's functions of linear reaction-diffusion equations with application to fast numerical solver. In *Mathematical and Scientific Machine Learning*, pp. 1–16. PMLR, 2022.

Haixin Wang, Jiaxin Li, Anubhav Dwivedi, Kentaro Hara, and Tailin Wu. Beno: Boundary-embedded neural operators for elliptic pdes. In *International Conference on Learning Representations*, volume 2024, pp. 8308–8329, 2024.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

Seungwoo Yoo, Kyeongmin Yeo, Jisung Hwang, and Minhyuk Sung. Neural green's functions. *Advances in Neural Information Processing Systems*, 38:47961–47982, 2026.

APPENDIX

A Architecture and implementation details **14**

A.1 Network configurations 14

A.2 FFT application and storage 14

A.3 Unfactorized control (DPK) 15

B Matched comparisons and direct ablations **15**

C Variable-source Darcy: protocol and results **18**

C.1 Source-span diagnostics 20

C.2 Kernel and linearity controls 20

C.3 Baselines, qualitative examples, and per-run results 21

D Poisson: protocol, supplementary runs, and figure details **22**

D.1 Screened Poisson: protocol and evaluation 24

E Exact truncation oracles **26**

F Boundary map on mechanical components **26**

G Cost accounting **28**

A Architecture and implementation details

A.1 Network configurations

Table 4 gives the configurations of the encoders and heads of Section 3.4. The Darcy encoder reads the normalized coefficient and two coordinate channels; each spectral block combines a spectral and a pointwise convolution, followed by group normalization and GELU. Its input is zero-padded from 85^2 to 96^2 and cropped after the encoder. On meshes, Fourier features of boundary coordinates and normals form one token type, and interior anchors form a second. Softmax slice assignments are multiplied by quadrature weights and normalized within each slice. A pre-norm transformer processes the pooled tokens together with the operator token, and separate query and source decoders cross-attend from coordinate Fourier features to these shared latent tokens to produce $e(x)$ and $e(y)$.

Component	Darcy grid	2D meshes	3D meshes
Encoder	spectral convolution	slice transformer	slice transformer
Encoder layers / width	4 / 32	4 / 192	4 / 256
Fourier modes per axis	12	–	–
Slices / heads / query blocks	–	32 / 4 / 2	48 / 4 / 2
Each modulation head	linear, $32 \to S$	MLP, $192 \to 96 \to S$	MLP, $256 \to 128 \to S$
Offset MLP	$4 \to 64 \to 64 \to S$	$5 \to 64 \to 64 \to S$	$6 \to 64 \to 64 \to S$
Main expansion sizes	16, 32, 64	16, 32, 64	16, 32, 64

Table 4: Configurations of the SMNO models; hidden MLP layers use GELU. The offset inputs are r, $\|r\|$, and $\log(\|r\| + h/2)$, and mesh models also receive h. The grid encoder's Fourier modes parameterize coefficient features, not the learned offset kernels.

On Darcy, the modulation and offset heads together use fewer than 2×10^4 parameters for $S \le 64$. The mesh models have about 2.5M (2D) and 4.4M (3D) parameters in total. DPK and SMNO share the encoder configuration. Darcy BE uses a factor head in place of the pairwise head and the same spectral encoder at width 48, which accounts for its larger parameter count. A learned scalar output scale is absorbed into the kernel notation. Full parameter counts and timings are in Tables 21 and 20.

A.2 FFT application and storage

On an n^d uniform grid with equal quadrature weights $m_j = 1/N$, $N = n^d$ (the Darcy convention), Eq. (9) can be evaluated as

$$\begin{aligned} \hat{u} &= \frac{1}{N} \sum_{s=1}^{S} w_s \odot \mathrm{crop}(z_s), \\ z_s &= \mathrm{FFT}^{-1}[\mathrm{FFT}[\mathrm{pad}(v_s \odot f)] \odot \mathrm{FFT}[\mathrm{embed}(\kappa_s)]] \,. \end{aligned} \tag{12}$$

Here pad places the weighted source in the first n^d entries of a zero-padded grid with at least $2n-1$ points per axis (192 for the 85^2 Darcy grid), embed places the $(2n-1)^d$ kernel offsets at their signed indices modulo the padded grid size, and crop keeps the first n^d output entries, which contain exactly the required pairwise sum without wrap-around.

After training, the S kernel tables are shared by all instances on the same grid. In the cached implementation, an instance stores $2SN$ modulation values and the shared tables another $O(SN)$ values. Application costs $O(SN \log N)$ per source, compared with $O(N^2)$ for a dense kernel product and $O(RN)$ for a rank-R basis expansion. The FFT evaluates the physical-space kernel and does not restrict it to a few Fourier modes. Darcy training evaluates the pairwise sum directly at the sampled query points, and this direct evaluation agrees with the FFT to within a relative error of 10^{-6} in our checks. Mesh experiments evaluate the pairwise sum directly for both SMNO and DPK. Table 20 reports the observed mesh training and full-evaluation times.

A.3 UNFACTORIZED CONTROL (DPK)

Our unfactorized control replaces the separable head with a dense pairwise kernel (DPK), retaining the encoder architecture and offset inputs:

$$\hat{G}^{\mathrm{DPK}}(x, y) = \mathrm{MLP}\big(e(x), e(y), r, \|r\|, \log(\|r\| + h/2)\big). \tag{13}$$

On grids, the DPK head projects the two point features and the offset inputs to a common width, adds the projections, and reads out a scalar; on meshes, it concatenates them. Both architectures are source-linear. Table 5 tests the effect of constraining this head to shared offset kernels and two-sided modulation. Build and apply times on the Darcy grid are in Appendix G.

Architecture	Poisson 2D	Poisson 3D	Screened Poisson	Darcy
SMNO-64	0.0144 $\pm$0.0010	0.0524 $\pm$0.0028	**0.0184**	**0.0877**
Unfactorized kernel (DPK)	**0.0139** $\pm$0.0001	**0.0463** $\pm$0.0112	0.1070	0.2046

Table 5: **Kernel-factorization ablation against an unfactorized head.** Band relative L^2 error. Poisson entries (2D ny32, 3D ny24) are the mean $\pm$ sample SD of pooled medians over three matched seeds (0, 1, 2). Screened Poisson entries (ny32) are seed-0 pooled medians, and Darcy entries are three-seed means over test sources and 200 coefficients. The 3D test uses unseen geometries at a trained resolution. Bolding the smaller point estimate does not imply significance. Full results are in Appendices B, D.1, and C.

B MATCHED COMPARISONS AND DIRECT ABLATIONS

All ten Poisson models are trained on source–solution pairs under the common training budget of Table 6. The protocol matches the data budget, not model capacity or computation.

Protocol	B	P	Q	Updates	Fields / points	Checkpoint
2D Poisson, ny32	8	8	64	150k	9.6M / 614.4M	fixed final
3D, ny16 (even updates)	8	8	48	100k	6.4M / 307.2M	final, 200k total
3D, ny24 (odd updates)	4	8	24	100k	3.2M / 76.8M	final, 200k total
Darcy direct ablations	4	1	32	250k	1M / 32M	Seen minimum
Screened Poisson, ny32	8	16	64	150k	19.2M / 1228.8M	Seen minimum

Table 6: **Effective training budgets.** B: instances per update; P: sources per instance; Q: supervised points per source. Sampling is with replacement, so the counts can include repeats. Every protocol uses solution loss only. The Poisson rows apply to all ten models of Table 7, the Darcy row to each seed of the four direct-intervention variants, and the Screened row to all eight models.

Poisson training protocol. Data, boundary conditions, mass-normalized source families, and coordinate conventions are given in Appendix D. All models except DeepONet observe the complete source field; DeepONet reads it at fixed sensors. No model receives reference Green's-kernel values as inputs or targets. Each update uses equal numbers of white-noise, Fourier, Gaussian, and polynomial sources. Models and data use separate random streams, so all models of one dimension see the same sequence of sources and query points. The loss is the mean squared prediction error divided by each field's reference energy plus 10^{-3} times the mean reference energy of the batch. AdamW uses peak learning rate 10^{-3} in 2D and 7×10^{-4} in 3D, 2,000 warmup updates, cosine decay to 10^{-5}, weight decay 10^{-4}, and gradient-norm clipping at 1. Training uses FP32 with TF32 matrix multiplications and no mixed precision; evaluation uses FP32 with TF32 disabled. Seen-family validation every 5k updates is used for monitoring only. All ten models share this loss and optimizer and are evaluated on all six families at the final weights after 150k (2D) or 200k (3D) updates. BIN's analytic terms are computed from geometry, not from the true domain Green's kernel. Some 2D models were trained on A100 GPUs and the others on H200 GPUs.

Model	Fourier	Gaussian	Poly.	Band	Localized	Wave
(a) 2D ny32: 300 geometries and 64 sources per family						
SMNO-64	0.0042	0.0024	0.0027	0.0155	0.0031	0.0039
SMNO-16	0.0038	**0.0020**	**0.0021**	0.0150	0.0030	0.0036
DPK (control)	**0.0036**	0.0020	0.0022	**0.0139**	**0.0027**	**0.0033**
BE-64 Yoo et al. (2026)	0.0159	0.0064	0.0079	0.1454	0.0346	0.0150
BE-256 Yoo et al. (2026)	0.0116	0.0053	0.0060	0.0587	0.0118	0.0107
BE-1024 Yoo et al. (2026)	0.0112	0.0052	0.0060	0.0539	0.0097	0.0104
NGO-style Melchers et al. (2026)	0.0146	0.0056	0.0067	0.1460	0.0341	0.0137
BIN Lin et al. (2021)	0.0164	0.0078	0.0107	0.0577	0.0108	0.0160
DeepONet Lu et al. (2021)	0.0999	0.0403	0.0366	1.0490	0.3109	0.1879
Transolver++ Luo et al. (2025)	0.3455	0.1269	0.1713	1.3856	0.5574	0.4350
(b) 3D ny16: 24 geometries and 48 sources per family						
SMNO-64	0.0205	0.0108	0.0134	0.0784	0.0209	0.0182
SMNO-16	0.0187	0.0090	**0.0080**	0.0879	0.0216	0.0159
DPK (control)	**0.0127**	**0.0076**	0.0090	**0.0478**	**0.0129**	**0.0113**
BE-64 Yoo et al. (2026)	0.1292	0.0331	0.0340	0.9952	0.2645	0.1040
BE-256 Yoo et al. (2026)	0.0676	0.0244	0.0292	0.9648	0.1670	0.0541
BE-1024 Yoo et al. (2026)	0.0607	0.0248	0.0304	0.9315	0.1366	0.0513
NGO-style Melchers et al. (2026)	0.1268	0.0315	0.0318	0.9951	0.2647	0.1003
BIN Lin et al. (2021)	0.0402	0.0232	0.0344	0.1520	0.0464	0.0317
DeepONet Lu et al. (2021)	1.0050	0.6019	0.2807	1.0672	0.9622	1.0020
Transolver++ Luo et al. (2025)	0.3319	0.1464	0.2303	1.5214	0.6745	0.5491
(c) 3D ny24: 24 geometries and 48 sources per family						
SMNO-64	0.0198	0.0104	0.0153	0.0550	0.0141	0.0187
SMNO-16	0.0160	0.0086	**0.0088**	0.0567	0.0152	0.0148
DPK (control)	**0.0126**	**0.0072**	0.0098	**0.0337**	**0.0094**	**0.0120**
BE-64 Yoo et al. (2026)	0.1306	0.0326	0.0353	0.9965	0.2533	0.1075
BE-256 Yoo et al. (2026)	0.0690	0.0251	0.0308	0.9719	0.1607	0.0569
BE-1024 Yoo et al. (2026)	0.0643	0.0250	0.0346	0.9405	0.1307	0.0548
NGO-style Melchers et al. (2026)	0.1282	0.0308	0.0325	0.9963	0.2523	0.1006
BIN Lin et al. (2021)	0.0324	0.0168	0.0297	0.0932	0.0254	0.0291
DeepONet Lu et al. (2021)	1.0014	0.6355	0.2861	1.0121	0.9753	1.0019
Transolver++ Luo et al. (2025)	0.3441	0.1556	0.2564	1.4835	0.6855	0.5659

Table 7: **Matched Poisson: all six source families.** Pooled lower median of the relative L^2 error over all shape–source pairs of each family, with 512 queries per field. All models use seed 0, solution-only supervision, and the fixed final checkpoint. Fourier, Gaussian, and Poly. are training categories; Band, Localized, and Wave are held out. Both 3D resolutions are trained. Bolding the smallest point estimate in each panel does not imply significance. At the 95th percentile (linear interpolation), both SMNO variants are below every baseline other than DPK in every family and grid, except that SMNO-64 and BIN tie to four decimals on 3D ny24 Gaussian (0.0366).

Transolver++ baseline. We use the released Transolver++ code with the architecture of its aircraft experiments: four layers, width 256, eight heads, 32 slices, MLP ratio 2, and dropout 0.1 (parameter counts in Table 20). Since no Poisson training recipe is released, we train it with the same source mixture, AdamW schedule, number of updates, and final-checkpoint rule as the other Poisson models of its dimension. The backbone receives all interior source nodes and boundary nodes with concatenated geometric and source features, and its outputs are read at the query nodes without an additional learned head. Its Gumbel slice sampling stays active at evaluation, and we report a single realization. No held-out-family result was used to select a checkpoint or hyperparameter.

Evaluation and uncertainty. Each field error is the unweighted Euclidean relative error over its 512 query values. Mass weighting normalizes sources but does not enter this norm. The test suites contain all 300 2D geometries with 64 sources per family and all 24 3D geometries per resolution with 48 sources per family. Family medians are lower medians pooled over all shape–source errors, not averages of per-shape medians, and 95th percentiles use linear interpolation. The paired bootstrap resamples test geometries with replacement 2,000 times, keeps all sources of each selected geometry, uses the same resamples for all models and families, and recomputes the pooled median each

Grid	SMNO-64 minus	Fourier	Band
2D ny32	BE-1024	-0.00699 $[-0.00725, -0.00675]$	-0.03834 $[-0.03891, -0.03770]$
	DPK	$+0.00063$ $[+0.00055, +0.00070]$	$+0.00160$ $[+0.00144, +0.00176]$
	Transolver++	-0.34132 $[-0.34555, -0.33682]$	-1.37012 $[-1.38249, -1.35728]$
3D ny16	BE-1024	-0.04027 $[-0.04435, -0.03776]$	-0.85305 $[-0.86869, -0.83840]$
	DPK	$+0.00781$ $[+0.00548, +0.00985]$	$+0.03062$ $[+0.02313, +0.03517]$
	Transolver++	-0.31142 $[-0.32057, -0.30406]$	-1.44297 $[-1.49739, -1.39406]$
3D ny24	BE-1024	-0.04452 $[-0.04877, -0.04134]$	-0.88547 $[-0.90111, -0.87196]$
	DPK	$+0.00726$ $[+0.00438, +0.00946]$	$+0.02130$ $[+0.01557, +0.02548]$
	Transolver++	-0.32426 $[-0.33325, -0.31658]$	-1.42853 $[-1.48574, -1.36962]$

Table 8: **Paired geometry-bootstrap differences (Fourier and Band).** Difference of pooled medians, SMNO-64 minus the comparator, with a pointwise 95% interval from resampling test geometries; negative favors SMNO-64. The intervals are conditional on the trained models, exclude training-seed variability, and are not adjusted for multiple comparisons. All 54 intervals over three comparators, six families, and three grids exclude zero: SMNO-64 has a lower median than BE-1024 and Transolver++ and a higher median than DPK in every family and grid.

Model	Fourier	Gaussian	Poly.	Band	Localized	Wave
(a) 2D ny32: 300 geometries and 64 sources per family						
SMNO-64	0.0038 $\pm$0.0004	0.0022 $\pm$0.0003	0.0025 $\pm$0.0004	0.0144 $\pm$0.0010	0.0029 $\pm$0.0003	0.0036 $\pm$0.0004
DPK	0.0037 $\pm$0.0001	0.0021 $\pm<$0.0001	0.0023 $\pm$0.0001	0.0139 $\pm$0.0001	0.0028 $\pm<$0.0001	0.0034 $\pm$0.0001
(b) 3D ny16: 24 geometries and 48 sources per family						
SMNO-64	0.0183 $\pm$0.0026	0.0094 $\pm$0.0015	0.0109 $\pm$0.0024	0.0736 $\pm$0.0079	0.0193 $\pm$0.0019	0.0158 $\pm$0.0028
DPK	0.0216 $\pm$0.0084	0.0127 $\pm$0.0053	0.0176 $\pm$0.0081	0.0574 $\pm$0.0087	0.0164 $\pm$0.0033	0.0207 $\pm$0.0089
(c) 3D ny24: 24 geometries and 48 sources per family						
SMNO-64	0.0166 $\pm$0.0033	0.0090 $\pm$0.0015	0.0123 $\pm$0.0031	0.0524 $\pm$0.0028	0.0132 $\pm$0.0009	0.0157 $\pm$0.0031
DPK	0.0245 $\pm$0.0109	0.0130 $\pm$0.0056	0.0203 $\pm$0.0093	0.0463 $\pm$0.0112	0.0130 $\pm$0.0033	0.0240 $\pm$0.0108

Table 9: **Three-seed SMNO-64/DPK comparison.** Mean $\pm$ sample SD (denominator $3-1$) over training seeds 0, 1, 2 of the pooled lower median of the relative L^2 error, with 512 queries per field. All runs use solution-only supervision and fixed final checkpoints (150k updates in 2D, 200k in mixed-resolution 3D training), and every trained seed is included. The 3D ny24 Band median is the prespecified primary endpoint; its values for seeds 0, 1, 2 are 0.0550, 0.0529, 0.0494 for SMNO-64 and 0.0337, 0.0549, 0.0504 for DPK. All other cells are descriptive.

time. Paired intervals are computed only for SMNO-64 against BE-1024, DPK, and Transolver++ (Table 8); the other models have point estimates only.

Three-seed SMNO-64/DPK comparison. Table 9 reports SMNO-64 and DPK trained at seeds 0, 1, and 2 under the protocol above. Seed 0 is the run of Table 7 (and, for SMNO-64, Table 1(a)); seeds 1 and 2 use the same model, sampling protocol, and test sources. On the primary endpoint, the 3D ny24 Band median, the ordering varies across seeds and the three-seed mean favors DPK.

Direct Darcy interventions. The four variants share $S = 64$, the spectral encoder and offset-MLP sizes, the 81-dimensional training source span, the budget of Table 6, BF16 training, and FP32 evaluation with TF32 disabled; the remaining Darcy protocol is in Appendix C. Each variant is trained with three seeds. We select the checkpoint with the lowest Seen-validation error among 41 evaluations during training and test it on all 200 test coefficients, all 77 test sources, and all 7,225 nodes. The no-log variant masks the log-distance input, the offset-only variant masks both $\|r\|$ and the log distance while keeping r, and the one-sided variant sets the source modulation to $v_s = 1$. Masking keeps the layer sizes, so the ablated variants have slightly fewer active parameters than the full model (1,197,121): 64 fewer without the log distance, 128 fewer for offset only, and 2,112

Variant / seed	Constant	Seen	Localized	Wave	Band	Signed sums
Full SMNO-64 / 0	0.0081	0.0131	0.0116	0.0752	0.0897	0.0109
Full SMNO-64 / 1	0.0094	0.0135	0.0122	0.0688	0.0852	0.0116
Full SMNO-64 / 2	0.0084	0.0135	0.0119	0.0714	0.0880	0.0113
No log / 0	0.0091	0.0141	0.0134	0.0920	0.1601	0.0124
No log / 1	0.0077	0.0131	0.0132	0.1125	0.2083	0.0119
No log / 2	0.0094	0.0137	0.0138	0.0972	0.1889	0.0127
Offset only / 0	0.0088	0.0133	0.0190	0.2606	0.4769	0.0165
Offset only / 1	0.0080	0.0128	0.0200	0.2980	0.5180	0.0173
Offset only / 2	0.0084	0.0129	0.0179	0.2490	0.4142	0.0156
One-sided ($v_s = 1$) / 0	0.0116	0.0341	0.0258	0.2030	0.2092	0.0230
One-sided ($v_s = 1$) / 1	0.0120	0.0344	0.0264	0.2093	0.2182	0.0235
One-sided ($v_s = 1$) / 2	0.0119	0.0337	0.0258	0.2058	0.2125	0.0227

Table 10: **Direct Darcy interventions: all seeds and source families.** Each cell averages all 200 coefficients and all sources of that family. Signed sums are solution errors rather than superposition defects. The summaries in Table 3 are the mean and sample SD (denominator $3 - 1$) of the three per-seed means.

Ablation	Band difference	95% coefficient-cluster interval	Ratio
No log	+0.09807	$[+0.09701, +0.09912]$	2.119
Offset only	+0.38205	$[+0.37883, +0.38529]$	5.358
One-sided ($v_s = 1$)	+0.12560	$[+0.12402, +0.12719]$	2.433

Table 11: **Primary paired Darcy endpoint.** Ablation minus full SMNO-64; positive is worse. Intervals come from a paired percentile bootstrap over 200 coefficient clusters with 2,000 resamples. Each resample keeps all sources and all three training seeds of a coefficient together. The intervals are conditional on the trained models. Ratios compare three-seed means. Every individual seed also has a positive Band interval.

fewer for the one-sided variant. Table 11 gives the paired Band contrasts. The full variant reuses the three SMNO-64 checkpoints of Table 1(b), which were trained on H100 GPUs, whereas the three ablated variants were trained on H200 GPUs. Because the training environments and active parameter counts differ, the comparison is not a strictly controlled causal test.

C VARIABLE-SOURCE DARCY: PROTOCOL AND RESULTS

We use the coefficient fields and split of the standard Darcy benchmark of FNO and Transolver. For each coefficient, one sparse factorization of the 421^2 conservative discretization generates all source responses before stride-five sampling; regenerated $f = 1$ labels differ from the public labels by less than 2×10^{-4} in median relative error. The training source bank consists of the constant mode and sine–cosine pairs with $\|k\|_\infty \le 4$. Each training example draws a new normalized Gaussian coefficient vector in this basis. The test bank contains 16 seen combinations, 16 localized sources with widths 0.025–0.055, 16 waves with continuously sampled directions and frequencies, 16 combinations of modes in $4 < \|k\|_\infty \le 8$, and 12 signed sums of 2, 4, or 8 localized sources. The constant column is the basis direction $f = 1$; it lies in the training span, but random draws in 81 dimensions almost surely do not reproduce that direction.

The pairwise kernels (SMNO, DPK, and the attention control ATT) sample 32 output locations per training solution; basis expansions and field models use the full field. All models minimize only the relative action loss and never receive exact operator rows as targets. FNO, DeepONet, and the Transolver models use their native field outputs on the same ordered sequence of coefficient–source pairs. Training uses BF16 mixed precision, and all reported errors are computed from the selected checkpoints in FP32 with TF32 disabled. For DPK, FNO, DeepONet, Transolver, Transolver++, and BE-64, the maximum learning rate was selected from three candidates on Seen-validation error, and each candidate observed the same one million training pairs. The selected rates are 3×10^{-3} for DPK, DeepONet, and BE-64 and 10^{-3} for FNO, Transolver, and Transolver++. The selected runs

Model	Constant	Seen	Localized	Wave	Band	Signed sums	$\mathcal{E}_{\mathrm{lin}}$
(a) Models of Table 1(b) and DPK							
SMNO-64, seed 0	0.0081	0.0131	0.0116	0.0752	0.0897	0.0109	–
SMNO-64, seed 1	0.0094	0.0135	0.0122	0.0688	0.0852	0.0116	–
SMNO-64, seed 2	0.0084	0.0135	0.0119	0.0714	0.0880	0.0113	–
SMNO-16, seed 0	0.0101	0.0174	0.0156	0.1038	0.1272	0.0146	–
SMNO-16, seed 1	0.0087	0.0150	0.0134	0.0947	0.1197	0.0124	–
SMNO-16, seed 2	0.0091	0.0165	0.0147	0.1151	0.1368	0.0136	–
DPK, seed 0	0.0126	0.0288	0.0253	0.1726	0.2011	0.0236	–
DPK, seed 1	0.0119	0.0293	0.0257	0.1684	0.1972	0.0238	–
DPK, seed 2	0.0120	0.0296	0.0260	0.1749	0.2157	0.0243	–
BE-512, seed 0	0.0088	0.0206	0.0449	0.8557	0.8458	0.0396	–
BE-512, seed 1	0.0092	0.0209	0.0452	0.8571	0.8465	0.0402	–
BE-512, seed 2	0.0094	0.0210	0.0450	0.8541	0.8439	0.0399	–
BE-64	0.0159	0.1054	0.0750	0.9215	0.9057	0.0625	–
FNO, seed 0	0.1869	0.0117	0.0687	0.6280	0.5612	0.0303	0.0425
FNO, seed 1	0.1628	0.0115	0.0815	0.4916	0.4775	0.0367	0.0448
FNO, seed 2	0.2097	0.0115	0.0803	0.3981	0.4309	0.0327	0.0467
Transolver	0.1848	0.0188	0.2708	0.5076	0.6050	0.0953	0.2576
Transolver++	0.1294	0.0220	0.4066	0.4815	0.5991	0.2313	0.4270
DeepONet	0.0636	0.0600	0.1365	0.9732	0.9762	0.0809	0.0562
(b) Basis-expansion and white-noise controls							
BE-256	0.0098	0.0237	0.0470	0.8670	0.8604	0.0415	–
BE-1024	0.0089	0.0196	0.0432	0.8436	0.8299	0.0384	–
NGF-style control, 128 features	0.0105	0.0445	0.0498	0.8808	0.8649	0.0431	–
NGF-style control, 256 features	0.0100	0.0224	0.0464	0.8624	0.8548	0.0410	–
Basis-matrix control, 128 bases	0.0398	0.0820	0.1396	2.5740	2.6072	0.1192	–
Basis-matrix control, 256 bases	0.0389	0.0810	0.1145	2.0409	2.1269	0.1031	–
BE-512, half white-noise sources	0.0128	0.0385	0.0325	0.3789	0.4425	0.0279	–
DPK, half white-noise sources	0.0129	0.0337	0.0259	0.1841	0.1970	0.0242	–
(c) FNO linearity controls							
FNO + superposition penalty	0.1831	0.0115	0.0820	0.4938	0.5287	0.0351	0.0490
FNO, post-hoc linearization (affine)	0.2676	0.3250	0.2704	0.5898	0.7122	0.2955	0.0139
FNO, post-hoc linearization (tangent)	0.2705	0.3280	0.2802	0.5132	0.6549	0.2962	–

Table 12: **Variable-source Darcy errors on every source family.** Mean relative L^2 error over the sources of each family and 200 test coefficient fields; rows without a seed label use seed 0. Panel (a) lists the Darcy runs used in Tables 1(b), 5, and 13, and its SMNO-64 rows are the full-model runs of Tables 3 and 10. $\mathcal{E}_{\mathrm{lin}}$ is the superposition defect of Eq. (14); a dash marks models that are linear in the source, for which it is zero up to floating-point rounding. Other seed-0 DPK, attention, and SMNO kernel controls are reported in Appendix C.2.

took 0.90 (DPK), 0.75 (FNO), 0.40 (DeepONet), 3.25 (Transolver), 3.15 (Transolver++), and 0.80 (BE-64) H100 hours of wall time, excluding the search. Parameter counts are in Table 21. Every other kernel variant, including SMNO, uses the DPK rate and every other FNO variant the FNO rate, so no test family enters checkpoint or hyperparameter selection. The superposition defect reported in the tables is

$$\mathcal{E}_{\mathrm{lin}} = \frac{\|\hat{u}(\alpha f_1 + \beta f_2) - \alpha\hat{u}(f_1) - \beta\hat{u}(f_2)\|_2}{\|\alpha\hat{u}(f_1) + \beta\hat{u}(f_2)\|_2} \tag{14}$$

over pairs of test sources with random unit-norm (α, β).

Additional basis-expansion controls. Besides BE-R, which reads two factor fields from a spectral encoder, we train two other parameterizations under the same protocol and the DPK learning rate. The NGF-style symmetric control (Yoo et al., 2026) reads one feature field Φ of dimension R and forms the symmetric kernel $(Q\Phi)(Q\Phi)^\top$, of rank at most R, with a learned $R \times R$ matrix Q. The basis-matrix control, motivated by NGO and VarMiON (Melchers et al., 2026; Patel et al., 2024), uses coefficient-independent basis networks ψ_i, ϕ_j of the coordinates (random Fourier features followed by a three-layer MLP, $n \in \{128, 256\}$ bases) and a dense $n \times n$ Green's matrix $N(c)$ predicted from pooled encoder features of c, so that only the matrix depends on c. We implement

Model	Seen	Localized	Band
SMNO-64	0.0002	0.0003	0.0023
SMNO-16	0.0012	0.0011	0.0086
DPK (control)	0.0004	0.0004	0.0097
BE-512	0.0002	0.0002	0.0014
FNO	0.0001	0.0071	0.0660

Table 13: Sample standard deviation (denominator $3-1$) across three training seeds of the per-seed mean relative L^2 error on variable-source Darcy, taken over the test sources and 200 coefficient fields. These SDs, the means in Table 1(b), and the DPK Band mean in Table 5 use unrounded per-seed values, so they can differ in the last digit from averages of the rounded rows of Table 12. Direct-intervention statistics are in Table 3.

this control with a shared spectral encoder, whereas NGO uses prescribed trial and test bases and VarMiON uses a learned linear branch on sampled sources. On the Band family, the NGF-style control reaches 0.8649 and 0.8548 at $R = 128$ and 256, and the basis-matrix control reaches 2.6072 and 2.1269 with 128 and 256 bases while remaining near 0.08 on Seen sources (Table 12).

Declared and effective rank. A basis expansion whose factor fields are read out of a w-wide feature through a pointwise affine head has rank at most $w + 1$, due to its bias, regardless of the declared rank. The BE-64 row uses a width-48 head. With this head, a declared rank of 512 reproduces the rank-64 errors on every source family to within 0.003 at seed 0 (Band: 9.09×10^{-1} against 9.06×10^{-1}). The BE-256, BE-512, and BE-1024 rows widen the last hidden layer of the factor head to the declared rank, and the NGF-style control uses its declared feature dimension without this bottleneck.

C.1 Source-span diagnostics

White-noise training and identification. For each training coefficient, we solve 16 white-noise sources, defined on the 85-grid with unit root-mean-square value and block-constant on the 421-grid, with the same conservative operator, and replace half of the training pairs by the resulting source–solution pairs. We select on Seen validation and use the test protocol of the other runs. This augmentation expands supervision beyond the low-frequency training span, but 16 sources per coefficient do not by themselves identify every entry of a full-grid operator. Without white-noise training, the seed-0 full-width basis expansions with 256, 512, and 1024 factors reach 8.604×10^{-1}, 8.458×10^{-1}, and 8.299×10^{-1} on the held-out Band family, despite exact truncation errors of 2.07×10^{-1}, 9.5×10^{-2}, and 2.45×10^{-2} (Table 19). This gap shows that action supervision on the 81-dimensional training span does not determine the operator outside it. The oracle uses the 85^2 operator, while learned errors use downsampled 421^2 targets, so their difference would not isolate the identification error. On the same downsampled targets, white-noise training reduces the Band error of BE-512 to 4.425×10^{-1}, while that of DPK changes from 2.011×10^{-1} to 1.97×10^{-1}. This small change suggests that offset inputs already constrain the kernel outside the training span. In the supplementary Poisson runs (Tables 14 and 15), BE-256 reaches 4.77×10^{-2} on the 2D Band family against a 1.4×10^{-2} truncation error, and BE-1024 reaches 6.15×10^{-1} in 3D against 1.11×10^{-1}.

Hybrid kernels. We also trained hybrid kernels that are dense at short range and low-rank at long range; they are not shown in Table 12 or the Poisson tables. On Darcy, none reached the seed-0 SMNO-16 Band error of 0.127: the best reached 0.589, or 0.243 when half of the training sources were white noise. In the supplementary Poisson runs, the hybrid reached Band medians of 0.0154 in 2D and of 0.041 at ny16 and 0.108 at ny24 in 3D.

C.2 Kernel and linearity controls

Unless stated otherwise, the Darcy runs in this subsection use seed 0 and the learning-rate rule of Appendix C.

Offset inputs in DPK and attention. Removing all offset inputs from DPK, leaving only $e(x), e(y)$, raises its seed-0 Band error from 2.011×10^{-1} to 8.477×10^{-1}; removing only the log-distance feature raises it to 2.643×10^{-1}. In the supplementary 2D Poisson runs, both removals give 1.35×10^{-2} versus

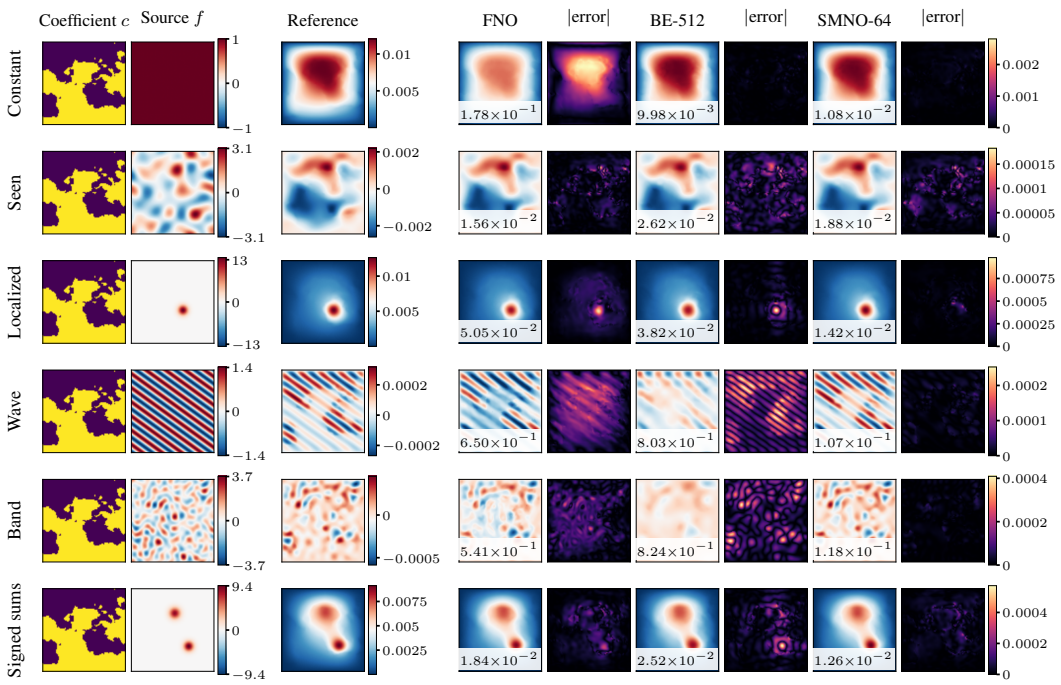

Figure 4: **One Darcy coefficient field, one source from each test family.** Every row uses the first test coefficient and the first test source of its family, with no selection by error; only f changes. FNO, the rank-512 basis expansion, and SMNO-64 are the seed-0 checkpoints underlying Table 1(b). Within a row, the reference and all predictions share the reference color scale; absolute-error panels share a second scale. Numbers are relative L^2 errors.

1.31×10^{-2} with all inputs (Table 14). The attention control ATT, with geometry-only queries and keys, reaches 5.529×10^{-1} on the Darcy Band family, and a learned radial logit bias $b(|r|)$ reduces this to 1.873×10^{-1}.

Expansion size and parameterization. With the encoder architecture fixed, the Darcy Band errors for $S = 1, 4, 16, 32, 64, 128$ are $8.219\times10^{-1}, 2.54\times10^{-1}, 1.272\times10^{-1}, 1.019\times10^{-1}, 8.97\times10^{-2}, 7.83\times10^{-2}$. Over seeds 0, 1, and 2, the mean Band errors for $S = 16, 32, 64$ are 1.279×10^{-1}, 9.65×10^{-2}, and 8.77×10^{-2}. At $S = 16$, isotropic offset kernels give 1.433×10^{-1}, and Fourier-parameterized offset kernels with 12 and 42 modes give 6.679×10^{-1} and 1.2898×10^{0}, so adding independently parameterized frequencies does not help the kernels transfer beyond the training band. A fixed analytic Φ ($S = 1$) with learned modulation and a rank-64 remainder gives 2.434×10^{-1}, whereas adding a rank-64 remainder to learned SMNO-16 and SMNO-64 changes 1.272×10^{-1} to 1.226×10^{-1} and 8.97×10^{-2} to 8.54×10^{-2}, respectively. Relative to plain SMNO-64 (Seen error 1.31×10^{-2}), SMNO-128 without a remainder and SMNO-64 with one improve only on the Wave and Band families, with Seen errors of 1.42×10^{-2} and 1.39×10^{-2}.

Linearity controls. At seed 0, a soft superposition penalty (weight 0.1) changes FNO's Darcy Band error from 5.612×10^{-1} to 5.287×10^{-1} and that of the field predictor in the supplementary 2D Poisson runs from 9.62×10^{-1} to 9.375×10^{-1}. Post-hoc Jacobian linearization of FNO around the zero source gives 7.122×10^{-1} (affine) and 6.549×10^{-1} (linear tangent). Enforcing or encouraging linearity therefore does not by itself recover the transfer of the learned offset kernels.

C.3 Baselines, qualitative examples, and per-run results

We use the released Transolver code and change only its input width from one field to two, (c, f). For Darcy, we implement Transolver++ from its paper (Luo et al., 2025). Table 13 gives three-seed standard deviations, and Table 12 gives per-run errors on every source family for the models of Table 1(b), DPK, and the basis-expansion, white-noise, and linearity controls. Figure 3(e) illustrates a case selected for a large accuracy gain. Among all 200 test coefficients and 16 unseen Wave

sources, we maximize the ratio of the lowest baseline error (FNO, DeepONet, or Transolver++) to the SMNO-64 error. For the selected pair, FNO, DeepONet, Transolver++, and SMNO-64 have relative errors 1.3431, 1.1321, 1.1309, and 0.0888; the BE-512 panel, which played no role in the selection, has 0.9342. This example is not an estimate of average performance. Table 1(b) reports aggregate results, and Figure 4 shows cases chosen without regard to error. Figure 2 shows the same case: its thumbnails show the first term ($s = 1$) of SMNO-64, the κ_1 curve is the section at zero vertical offset, and the prediction uses all 64 terms.

D POISSON: PROTOCOL, SUPPLEMENTARY RUNS, AND FIGURE DETAILS

Data and discretization. All ordinary Poisson runs solve $-\Delta u = f$ with $u = 0$ on the whole boundary. A 2D channel occupies $0 \le x_1 \le W$ with $W \sim \mathcal{U}(0.8, 1.6)$, between walls that perturb $x_2 = 0$ and $x_2 = 1$ by random six-term sine series in x_1/W; a symmetric Gaussian neck is added with probability one half, and draws thinner than 0.30 or with wall slope above 1.3 are rejected. A 3D duct occupies $0 \le x_1 \le W$, $0 \le x_2 \le 1$, between bottom and top surfaces that perturb $x_3 = 0$ and $x_3 = 1$ by random sums of $\sin(k_1\pi x_1/W)\sin(k_2\pi x_2)$ with $k_1, k_2 \le 3$, so its side faces are planar; ducts with a gap below 0.35 are rejected. Structured transfinite meshes have ny cells across the height (and, in 3D, across the unit width) and ny W cells, rounded to an integer, along x_1; each cell is split into two triangles or six tetrahedra. Over all meshes, a channel has 775 to 1,550 interior nodes at ny32 in every split, and a duct has 2,700 to 5,625 at ny16 and 9,522 to 19,573 at ny24; the validation and test ducts have at most 5,400 nodes at ny16, and the test ducts at most 19,044 at ny24. References are P1 finite-element solutions $u_I = A_{II}^{-1}\,\mathrm{diag}(m) f_I$ with lumped nodal mass m, computed by sparse LU factorization at 512 interior evaluation nodes drawn once per mesh. The 2D split has 1,000 training, 100 validation, and 300 test channels at ny32, and every ordinary 2D Poisson run trains on all 1,000. In 3D, 800 training ducts are meshed at ny16 and the first 300 of them also at ny24, 24 validation ducts at ny16, and 24 test ducts at both resolutions. Training, validation, and test geometries are generated from disjoint random seeds. Training sources are drawn with replacement from a bank of 64 realizations per family and training mesh, generated once before training.

Poisson mesh baselines. BE-R is our learned basis expansion of rank R with separate evaluation- and source-side factor heads. The NGO-style basis-matrix control couples rank-64 evaluation- and source-side bases by a 64×64 core matrix predicted from the pooled geometry encoding. DPK, SMNO, BE, and the NGO-style control apply a source-independent kernel to the nodal source, so their predictions are linear in f. BIN is a source-conditioned boundary-integral network: it adds a single-layer potential to the free-space Newtonian potential of f, and a boundary transformer predicts the layer density from geometry features and the boundary trace of that Newtonian potential. Both potentials use free-space kernels, and the prediction is not constrained to be linear in f. DeepONet's branch network reads the source at 512 evenly spaced indices of the zero-padded nodal array (1,024 in 3D), its trunk network reads the query coordinates, and both also receive the pooled geometry encoding. In 3D, DeepONet's geometry encoder has width 192 with 32 slices and 256 interior anchors, whereas the other 3D models built on our geometry encoder use width 256 with 48 slices and 512 anchors. The matched runs reuse the architectures and parameter counts of Table 16. Table 20 adds 3D BE-1024 and Transolver++. The matched 2D SMNO-16 has 2,460,753 parameters, and the matched 3D BE-256 has 4,555,313.

Supplementary Poisson runs. Tables 14, 15, and 16 report runs under protocols that differ from that of Appendix B; their seed statistics are not pooled with the matched comparison. The four 2D SMNO-64 runs of Table 14 (seeds 0–3) were trained independently on source–solution pairs only, for 150,000 updates on the 1,000 training channels, with 100 validation channels, batch size 8, 64 sampled query nodes, 16 sources per geometry (four each from white noise, Fourier, Gaussian, and polynomial distributions), and the 2D AdamW schedule of Appendix B. They are not repeats of the eight-source matched runs in Table 1(a). Validation uses only the three structured training families, and each of these runs is evaluated at its fixed final checkpoint. In these 2D runs, DPK, SMNO-16/32, BE at every width (including BE-1024), and the NGO-style control add a 0.1-weighted relative operator-row loss, whereas BIN, DeepONet, and the field predictor are solution-only; the field predictor trained with a superposition penalty has its own row. Every displayed BE-R uses separate factor heads of hidden width R (full width), and BE-1024 follows the BE-256 recipe at seed 0. In 3D, DPK, SMNO, BE, and the NGO-style control include the 0.1 row loss, and BIN, DeepONet, and the field predictor are solution-only; kernel models use 16 sources per geometry per update,

Model	Fourier	Gaussian	Poly.	Band	Localized	Wave	$\mathcal{E}_{\mathrm{lin}}$
SMNO-64, seed 0	0.0032	0.0018	0.0019	0.0133	0.0025	0.0029	–
SMNO-64, seed 1	0.0040	0.0023	0.0026	0.0149	0.0031	0.0037	–
SMNO-64, seed 2	0.0035	0.0019	0.0021	0.0145	0.0027	0.0032	–
SMNO-64, seed 3	0.0034	0.0019	0.0020	0.0144	0.0027	0.0031	–
DPK	0.0032	0.0017	0.0018	0.0131	0.0024	0.0030	–
DPK without the log-distance feature	0.0038	0.0021	0.0024	0.0135	0.0027	0.0035	–
DPK without offset inputs	0.0035	0.0019	0.0021	0.0135	0.0026	0.0033	–
SMNO-16, seed 0	0.0043	0.0023	0.0025	0.0160	0.0032	0.0040	–
SMNO-16, seed 1	0.0037	0.0019	0.0019	0.0149	0.0028	0.0034	–
SMNO-16, seed 2	0.0037	0.0019	0.0020	0.0155	0.0029	0.0034	–
SMNO-32	0.0039	0.0022	0.0025	0.0149	0.0029	0.0036	–
Field predictor (same encoder)	0.0448	0.0166	0.0182	0.9620	0.1655	0.2244	0.3537
Field predictor + superposition penalty (weight 0.1)	0.0375	0.0274	0.0151	0.9375	0.1709	0.1293	0.2319
BE-64	0.0133	0.0044	0.0051	0.1365	0.0330	0.0126	–
BE-256	0.0088	0.0035	0.0037	0.0477	0.0090	0.0079	–
BE-1024	0.0093	0.0042	0.0049	0.0415	0.0073	0.0086	–

Table 14: Supplementary 2D channel runs. Entries are median relative errors pooled over 64 sources per family on 300 held-out channels at ny32, for seed 0 unless stated; training configurations are described under Supplementary Poisson runs. Fourier, Gaussian, and Poly. are training families; Band, Localized, and Wave are held out. $\mathcal{E}_{\mathrm{lin}}$ is the superposition defect; a dash marks models that are linear in the source, for which it is zero up to floating-point rounding. Matched results are in Table 7.

Model	Fourier	Gaussian	Poly.	Band	Localized	Wave	$\mathcal{E}_{\mathrm{lin}}$	ny24 Fourier	ny24 Band
DPK	0.0094	0.0052	0.0060	0.0326	0.0089	0.0080	–	0.0086	0.0234
SMNO-16	0.0185	0.0094	0.0085	0.0775	0.0208	0.0158	–	0.0150	0.0522
SMNO-32	0.0275	0.0155	0.0201	0.0832	0.0232	0.0264	–	0.0316	0.0645
SMNO-64	0.0280	0.0157	0.0223	0.0806	0.0221	0.0256	–	0.0336	0.0643
Field predictor (same encoder), 8 sources	0.1162	0.0355	0.0490	1.0121	0.2780	0.4882	0.4006	0.1174	1.0111
Field predictor, 16 sources	0.0686	0.0245	0.0342	1.0027	0.4853	0.5021	0.3322	0.0689	1.0034
BE-1024	0.0436	0.0165	0.0188	0.6150	0.0946	0.0343	–	0.0454	0.6217

Table 15: Supplementary 3D duct runs with varying supervision and budgets. All rows use seed 0. Entries are median relative errors pooled over 24 held-out ducts and 48 sources per family at ny16; the last two columns evaluate the same checkpoints at ny24, also over 24 ducts and 48 sources per family. Training uses both resolutions. Columns as in Table 14. The DPK runs with seeds 1 and 2 have ny16 Band medians of 0.0624 and 0.1063, and the SMNO-16 runs 0.0790 and 0.1139. BE-64, BE-256, and BE-512 were also trained, as were BE-512 and BE-1024 with narrow factor heads (rank at most 257). These runs are not shown, and each has a higher Band error than BE-1024 at both resolutions. Matched results are in Table 7.

DeepONet and the field predictor use eight (except the field-predictor row labeled 16 sources), and BIN uses four and supervises all 512 evaluation nodes.

Source distributions. For dimension $d \in \{2, 3\}$, the Fourier and Band families use

$$f(x) \propto \prod_{j=1}^{d} \sin(\pi k_j \tilde{x}_j + \varphi_j), \qquad \varphi_j \sim \mathcal{U}(0, \pi), \tag{15}$$

where $\tilde{x}_1 = x_1/W$ for channel or duct length W; the remaining coordinates use the unit-height scale. Each k_j is sampled independently from $\{1, 2, 3\}$ for Fourier and $\{4, 5, 6\}$ for Band. All sources are normalized to unit discrete, mass-weighted L^2 norm. Training also uses independent Gaussian white-noise values at mesh nodes, Gaussian sources with log-uniform widths 0.10–0.30, and random polynomials of degree at most two. The held-out Localized and Wave families use narrow Gaussians (log-uniform widths 0.03–0.08) and oblique sinusoidal waves. In 2D, Gaussian centers are uniform on $[0.15W, 0.85W] \times [0.2, 0.8]$, and Wave uses $f(x) \propto \sin(k(\cos\theta\, x_1/W + \sin\theta\, x_2) + \varphi)$ with $k \sim \mathcal{U}(6, 12)$, $\theta \sim \mathcal{U}(0, 2\pi)$, and $\varphi \sim \mathcal{U}(0, \pi)$. In 3D, Gaussian centers are uniform on $[0.15W, 0.85W] \times [0.2, 0.8]^2$, and Wave uses $\sin(k\,\nu \cdot x + \varphi)$ in physical coordinates, with ν uniform on the unit sphere and k, φ as in 2D. Band, Localized, and Wave are held out as

2D: 1,000 channels at ny32, 150k updates			3D: 800/300 ducts at ny16/24, 200k updates		
model	parameters	seeds	model	parameters	seeds
SMNO-64	2.47M	4	DPK	4.46M	3
DPK	2.51M	3	SMNO-16/32/64	4.37–4.38M	3/1/1
BE-64	2.51M	2	BE	4.46M	1
BE-256	2.65M	1	NGO-style	5.58M	1
BE-1024	4.91M	1	Field predictor	4.37M	1
NGO-style	3.34M	2	DeepONet	2.23M	1
BIN	2.48M	2	BIN*	4.41M	1
DeepONet	1.89M	2			
Field predictor	2.46M	3			
Transolver++	1.74M	1			

Table 16: Supplementary Poisson runs: parameter counts and seeds (not the matched comparison). *The 3D BIN trains on 300 ducts at ny16 for 150k updates. The 2D learning rate is 0.001; the shared 3D recipe uses 0.0007, while the 3D DeepONet uses 0.0003. Hyperparameters were not tuned systematically or with equal effort across models. Held-out source families are evaluation-only.

source categories. Their spectral support is not held out, since white-noise training sources already contain broad frequencies.

Matched-encoder field control. The field predictor is our own control model and is distinct from the released Transolver baseline. It uses the same geometry-encoder architecture as the kernel models and adds a source branch: source coordinates, quadrature weights, and values of f are embedded, pooled into slices, and processed by two transformer layers. Query cross-attention reads the combined geometry and source tokens, and an MLP predicts $u(x)$ directly. This source path is nonlinear in f, whereas applying a source-independent kernel is linear. The slice design follows Wu et al. (2024). The 2D source branch supports point sampling, whereas the 3D branch processes the full interior point cloud.

Figure details. Figure 1 uses the first test channel at ny32 and the seed-0 BE-64 and SMNO-64 checkpoints at 150,000 updates. The model labeled NGF-style in the figure is BE-64 in the tables: our rank-64 implementation with separate evaluation- and source-side factor heads instead of the published shared-feature symmetric factorization. The target x_0 is the evaluation node nearest the mid-channel point, fixed before inference. If T denotes the discrete source-to-solution matrix, the plotted values are $G_h(x_0, y_j) = T_{i_0 j}/m_j$ for both the reference and the learned models, so the reference is a row of the inverse FEM stiffness matrix A_{II}^{-1}, and each value is the response at x_0 to the unit discrete source $f_j = 1/m_j$ at y_j with all other entries zero. The plotted kernel section contains all 31 source nodes in the coordinate column of x_0, plotted against $y_2 - x_{0,2}$, without interpolation, symmetrization, or rescaling. Its finite diagonal value is included. The plot compares discrete kernels and is not evidence of a divergent continuous singularity. Panel (a) renders the reference response to such a source at $y_0 = x_0$ on the original interior FEM triangles, using the symmetry $G_h(x, y_0) = G_h(y_0, x)$ of the Poisson reference. This reciprocity is applied only to the reference surface and to neither learned kernel. Figure 3(a) uses the second test channel, the first Wave test source, and the original 512 evaluation nodes. Its four predictions use the matched seed-0 final checkpoints. Neither the channel nor the source was selected by prediction error, and the panels keep the reference amplitude without rescaling. Figure 3(b) follows the same rule with the second test duct at ny24 and its first Wave test source, using the matched seed-0 final checkpoints. For display, each model is evaluated at every mesh node, with boundary values set to zero, and the duct is cut at mid-width. The labels give errors on the 512 evaluation nodes, as in the tables. Figure 3(c) uses the second screened test channel ($\sigma = 1.31$) and its first Wave test source, with the checkpoints of Table 2(a) and the Screened Poisson DPK control of Table 5.

D.1 Screened Poisson: protocol and evaluation

Data and discretization. We solve $-\Delta u + \sigma^2 u = f$ with homogeneous Dirichlet data on the 2D channel family, with one screening parameter per geometry sampled log-uniformly from $[0.5, 8]$.

Model	Parameters	Selected update
SMNO-64	2,473,185	145,000
SMNO-32	2,464,897	150,000
DPK (control)	2,508,514	105,000
BE-64 Yoo et al. (2026)	2,513,825	150,000
BE-256 Yoo et al. (2026)	2,645,409	150,000
Field predictor (control)	2,458,050	135,000
Transolver Wu et al. (2024)	716,225	150,000
Transolver++ Luo et al. (2025)	1,743,013	150,000

Table 17: **Screened Poisson: model sizes and selected updates.** Every model is trained for 150,000 updates; the models are not parameter-matched.

Model	Fourier	Gaussian	Poly.	Band	Localized	Wave
Median						
SMNO-64	**0.0081**	**0.0065**	**0.0082**	**0.0184**	**0.0065**	**0.0081**
SMNO-32	0.0135	0.0098	0.0134	0.0247	0.0088	0.0140
DPK (control)	0.0708	0.0438	0.0631	0.1070	0.0320	0.0712
BE-64 Yoo et al. (2026)	0.0219	0.0106	0.0120	0.1765	0.0486	0.0213
BE-256 Yoo et al. (2026)	0.0165	0.0093	0.0116	0.0672	0.0172	0.0160
Field predictor (control)	0.0503	0.0239	0.0386	0.5876	0.2107	0.0831
Transolver Wu et al. (2024)	0.0315	0.0135	0.0163	0.4954	0.2193	0.0437
Transolver++ Luo et al. (2025)	0.0553	0.0361	0.0447	0.2914	0.1496	0.0645
95th percentile						
SMNO-64	0.0608	0.0326	0.0518	**0.0884**	**0.0299**	0.0636
SMNO-32	0.0912	0.0489	0.0797	0.1205	0.0419	0.0955
DPK (control)	0.1959	0.1537	0.1508	0.2425	0.1754	0.1736
BE-64 Yoo et al. (2026)	0.0591	0.0429	0.0447	0.4214	0.1320	0.0591
BE-256 Yoo et al. (2026)	**0.0525**	0.0413	0.0441	0.1248	0.0549	**0.0483**
Field predictor (control)	0.1405	0.0739	0.1064	0.8228	0.5458	0.2070
Transolver Wu et al. (2024)	0.0588	**0.0287**	**0.0357**	0.9211	0.4377	0.0772
Transolver++ Luo et al. (2025)	0.0962	0.0603	0.0687	0.4142	0.3409	0.1129

Table 18: **Screened Poisson: full ny32 results.** Relative L^2 errors pooled over 100 held-out shapes and 64 sources per shape and family. All models use seed 0, are trained at ny32, and are selected on Seen validation. Fourier, Gaussian, and polynomial are training categories; Band, Localized, and Wave are held out. Bold marks the lowest displayed value within each statistic and column.

The P1 finite-element system is $(A + \sigma^2 \operatorname{diag}(m))u = \operatorname{diag}(m)f$, where A is the interior stiffness matrix and m the lumped nodal mass. The split contains 300 training, 100 validation, and 100 test geometries; only the first 24 validation geometries are used for model selection. Training and evaluation use ny32, and all eight models use the same geometries, screening parameters, and targets in every split.

Supervision and selection. Every model observes all nodal source values and is trained against reference solutions at 64 sampled query nodes. Each update uses eight geometries and 16 sources per geometry, four each from white noise, Fourier, Gaussian, and polynomial distributions. Source definitions, normalization, and held-out categories are those of ordinary Poisson above. All runs use seed 0 and complete 150,000 updates with the 2D AdamW schedule of Appendix B. Validation every 5,000 updates uses 16 fixed sources per Seen family; the selected checkpoint minimizes the mean of the Fourier, Gaussian, and polynomial validation medians. Neither test data nor held-out source categories enter selection.

Architectures and baseline implementation. The six core configurations are SMNO with 32 and 64 terms, BE with 64 and 256 independent factor outputs (our NGF-style form of Yoo et al. (2026), modified to use separate source- and evaluation-side heads), DPK, and the matched-encoder field predictor (Appendix D, Matched-encoder field control). They share the width-192 geometry-encoder configuration and receive $\log(\sigma)/3$ as an operator token. The Transolver baseline uses the

released irregular-mesh code of Wu et al. (2024) with the architecture of its elasticity experiments (8 layers, width 128, 8 heads, 64 slices). The backbone is unchanged; only the input features and the readout at the supervised nodes are adapted. It processes all source and boundary nodes under the same data, source, query, and update budgets, and it is not parameter-matched to the core models. Transolver++ uses the released code, architecture, input features, and evaluation mode of Appendix B, with the same budget and Seen-only selection. Table 17 gives parameter counts and selected updates.

Metrics and scope. Evaluation uses 100 test shapes and 64 fixed source samples per shape and family. Relative errors use the unweighted nodal L^2 norm on the evaluation nodes. Each family median and 95th percentile pools all 6,400 shape–source observations, with the percentile conventions of Appendix B; Seen is the arithmetic mean of the three in-distribution family medians, not a median pooled across families. Table 18 shows all six source families and eight models at ny32; SMNO-64 has the lowest median in every family. These single-seed results support source-distribution transfer at the training resolution but do not establish statistical significance.

E Exact truncation oracles

Any $(R+1)$-dimensional source subspace contains a direction that a rank-R operator maps to zero, so a uniform relative error below one on a band of eigenmodes requires rank at least the number of modes in the band. Weyl's law ties this count to domain volume and frequency rather than to mesh resolution. Table 19 reports the Band error of exact spectral truncations as a function of the retained rank. These errors alone do not explain the transfer error of the learned basis expansions (Appendix C.1).

We use 100 Darcy test coefficients, 24 2D test shapes, and 12 3D test shapes per resolution, reporting means for Darcy and medians for Poisson. For each instance, we form the true discrete operator: the stiffness and lumped mass matrices (A, M) on meshes, or the 85-grid five-point conservative operator on Darcy. On meshes, dense double-precision diagonalization of $M^{-1/2}AM^{-1/2}$ gives $A\phi_j = \lambda_j M\phi_j$ with $\phi_j^\top M\phi_k = \delta_{jk}$ and $0 < \lambda_1 \le \lambda_2 \le \cdots$. We evaluate $u_r = \sum_{j\le r} \lambda_j^{-1}(\phi_j^\top Mf)\phi_j$ on the same source families as the learned models. Darcy uses the corresponding Euclidean expansion and is compared with the full 85-grid solve, so the reported number is the representation error alone. That solve differs from the 421-grid targets by 1.9×10^{-2}–7.7×10^{-2}, depending on the family. This gap comes from the coarse solver's discretization, and a learned model's error can fall below it, since a learner may approximate corrections to that solver. Spectral truncation is optimal in unitarily invariant norms of the mass-symmetrized inverse (the Euclidean inverse on Darcy); under the reported relative action metric, it shows what the retained modes capture but is not a lower bound for every learned model.

Problem / mesh	r=64	128	256	512	1024	2048
Darcy, $N = 7225$	0.748	0.429	0.207	0.095	0.0245	0.0082
3D duct ny16, $N = 5625$	0.993	0.976	0.778	0.272	0.111	0.033
3D duct ny24, $N = 19573$	0.993	0.975	0.77	0.233	0.095	0.04
2D channel ny32	0.111	0.037	0.014	0.005		

Table 19: **Spectral truncation diagnostic.** Held-out Band error after retaining the r lowest-eigenvalue modes of the differential operator (the largest eigenvalues of its inverse). The two 3D resolutions give similar errors at a given r.

F Boundary map on mechanical components

The Poisson benchmark of NGF uses shapes from MCB dataset B (Kim et al., 2020) and contains problems with varying boundary values and interior sources on five CAD categories; its hardest split combines an unseen shape with an unseen problem (Yoo et al., 2026). The model adds a learned boundary correction to a particular solution (Section 3.4). Its boundary map is positive with unit row sums. A small network encodes per-node geometric statistics and coordinates, and a pair MLP combines evaluation- and boundary-node features with regularized distances. The particular

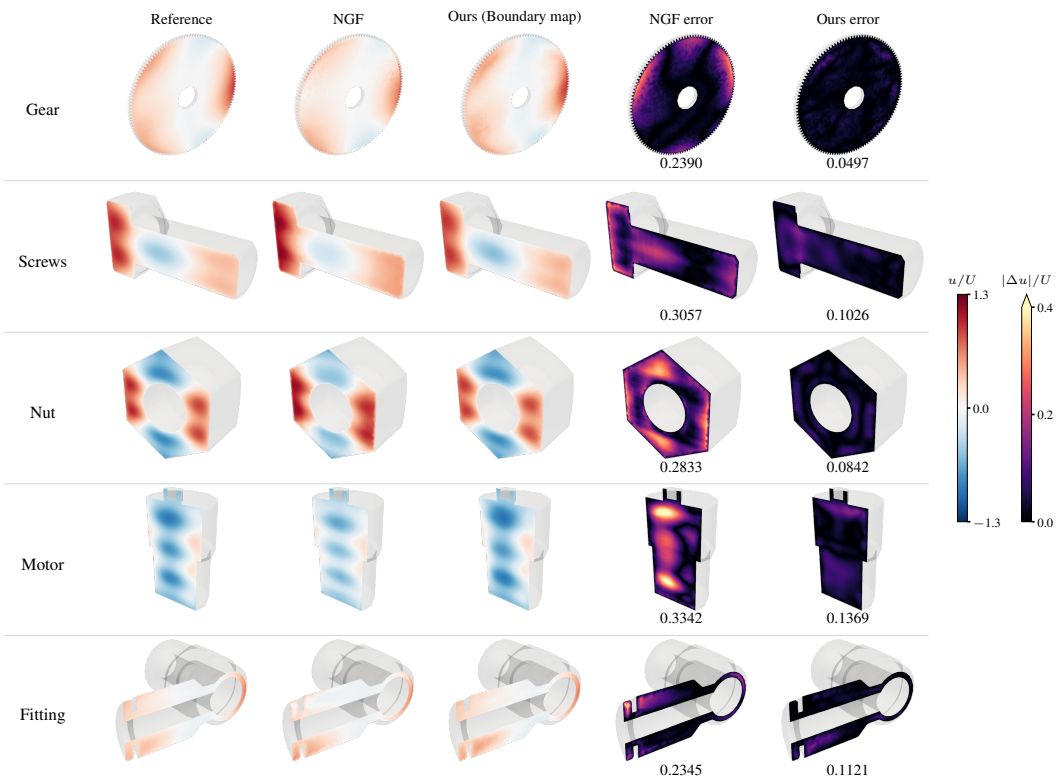

Figure 5: **Additional MCB shapes, selected for large error gaps.** One unseen shape per category at the fixed unseen problem, shown as in Figure 3(d). Labels give unweighted nodal relative L^2 error on all mesh vertices. The boundary map has lower error than NGF on 41 of the 50 reviewed shapes.

solution uses a discrete sine transform Poisson solve on a 129^3 box, with nearest-neighbor extension of the source and trilinear sampling back to the mesh. The boundary map has 0.027M parameters (0.039M for motor). These counts exclude the particular-solution computation and preprocessing, whose cost we did not measure. The Transolver and NGF values in Table 2(b) are taken from the benchmark's original tables and do not use the analytic particular solution; we use the same splits, mass matrices, and metric (mean unweighted nodal relative L^2 error).

Training and aggregation. Each update samples one training shape uniformly and uses all its problem combinations, supervising at most 6,144 randomly sampled interior nodes. The loss averages unsquared nodal relative L^2 errors over these problems. AdamW uses learning rate 0.0008, weight decay 0.0001, 1,000 warmup updates, cosine decay to 0.00001, and gradient clipping at 1. We evaluate the final checkpoint of each run after 20,000 updates (30,000 for motor) and do not select checkpoints by validation error. Training uses the known-shape, known-problem split. Our test errors include prescribed boundary values and at most 32,768 interior nodes per mesh, and average over the 320 unseen-shape, unseen-problem cases per category, then over training seeds (three for gear, nut, screws, and motor; one for fitting). The overall mean weights the five categories equally.

Qualitative comparison. We reviewed the first ten unseen shapes of each category, in order of shape ID, from the unseen-shape, unseen-problem split, all at one fixed unseen problem that had been chosen on a fitting example before this review. The first row of Figure 3(d) shows a screws-and-bolts shape, chosen for illustration because its geometry and the qualitative comparison are easy to read, and its errors do not estimate average performance. The NGF field is the per-sample prediction released by the NGF authors. Our field comes from the seed-0 screws-and-bolts model after 20,000 updates and includes the prescribed boundary values. Neither model is retrained for the figure. All panels share one cut plane, chosen from the geometry before inference, and fixed color scales. The labels, 0.3033 for NGF and 0.1198 for the boundary map, are unweighted nodal relative

L^2 errors over all mesh vertices. Figure 5 shows, for each category, the reviewed shape with the largest ratio of NGF error to boundary-map error; in screws it shows the third largest, because the top two are slender bolts whose cut is hard to read. The nut in the second row of Figure 3(d) also appears in this gallery. The gallery shows best-case gaps, which are not typical.

G COST ACCOUNTING

Tables 20 and 21 report matched-Poisson and Darcy-grid costs.

Task / model	Parameters	Train loop (h)	Full eval. (s)	Peak (GiB)
2D / SMNO-64	2,473,185	0.894	86.0	6.92
2D / DPK (control)	2,508,514	0.981	81.8	8.39
2D / BE-1024 Yoo et al. (2026)	4,909,473	1.026	74.2	5.56
2D / Transolver++ Luo et al. (2025)	1,743,013	7.212	138.5	5.40
3D / SMNO-64	4,383,217	7.188	36.7	31.80
3D / DPK (control)	4,458,034	6.552	38.5	33.33
3D / BE-1024 Yoo et al. (2026)	6,917,681	6.514	40.0	31.08
3D / Transolver++ Luo et al. (2025)	1,744,037	12.118	71.3	35.91

Table 20: **Observed matched-Poisson cost** (one H200 per run). Training time includes periodic validation. Each 3D evaluation covers ny16 and ny24. Peak memory includes the training data, which stay on the GPU (both resolutions in 3D). All times are wall-clock measurements from implementations not tuned for speed, and we report no FLOP counts. In 3D, SMNO trains no faster than BE or DPK.

		Build / apply to 64 sources (ms)			
Model	Parameters	85^2	171^2	341^2	421^2
SMNO-16	1.19M	2.8 / 1.0	2.6 / 3.9	2.8 / 16.2	4.3 / 27.5
SMNO-32	1.20M	2.8 / 2.1	2.7 / 8.1	2.8 / 33.6	4.4 / 56.9
SMNO-64	1.20M	2.8 / 4.0	2.7 / 15.8	2.8 / 65.9	4.5 / 114.4
DPK	1.19M	47.5 / 0.16	628 / 5.5	9975 / 96.2	–
BE-64	2.67M	2.4 / 0.06	2.4 / 0.05	3.7 / 0.13	5.9 / 0.20
BE-512	3.21M	2.4 / 0.07	2.4 / 0.13	9.2 / 0.45	14.1 / 0.65
FNO	4.74M	11.2	37.3	135.0	169.9
DeepONet	2.64M	3.8	4.4	12.2	19.0
Transolver	2.83M	142.5	527.4	2066	3159
Transolver++	1.65M	113.3	424.8	1672	2536

Table 21: **Cost on $n \times n$ Darcy grids** (one H100, FP32 with TF32 disabled). The trained 85^2 checkpoints run unchanged on the larger grids with random inputs, since cost does not depend on input values. For repeated sources on one instance, source-independent quantities can be cached without changing the forward map. *Build* includes encoder evaluation and cache construction (modulations and kernel tables for SMNO, factors for BE, dense entries for DPK); *apply* acts on 64 sources (a dense product for DPK, two thin matrix products for BE, the S FFT convolutions of Eq. (12) for SMNO). Field models have no cached operator, and their entry is the time of 64 forward passes in batches of 16. At 421^2, DPK's $N \times N$ matrix does not fit on the GPU. Times are means over repeated runs after warm-up on an otherwise idle GPU, with fewer repetitions on the larger grids.

For one new coefficient field and the first batch of 64 sources, build plus application takes 6.8 ms for SMNO-64 and 47.7 ms for DPK. BE (0.06–0.07 ms), DPK (0.16 ms), and DeepONet (3.8 ms) process 64 sources faster than the 4.0 ms FFT application of SMNO-64, but these models have larger errors on the tested held-out sources (Tables 1(b) and 5). At $N = 7225$ a cached dense or rank-R operator is applied by one matrix product, whereas SMNO performs $2S$ zero-padded FFTs per source, so its application is slower on this small grid. On larger grids the order reverses. SMNO-16 applies faster than DPK's cached product from 171^2 (3.9 versus 5.5 ms), and SMNO-64 is faster

at 341^2 (65.9 versus 96.2 ms), where DPK needs 10 s to build its 54 GB matrix. At 421^2 the matrix does not fit on the GPU. The DPK cache holds $N^2 \approx 52$M values at $N = 7225$ and would require about 125 GB in FP32 at 421^2, whereas SMNO stores $2SN$ instance modulations plus shared kernel tables, so its storage scales as $O(SN)$ (Appendix A.2).