LINK: LEARNING PHYSICAL INTERACTION KERNELS FOR NEURAL OPERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural operators on irregular domains require interactions that remain meaningful as discretization and geometry change. Attention provides a flexible mechanism for exchanging information between sampled locations, but its routing is typically determined through generic query–key interactions rather than directly from physical structure. We introduce LINK (Learned Interaction Kernels), a kernel attention mechanism that learns physical-space routing between locations using their coordinates, relative geometry, and the current problem instance. The resulting input-conditioned kernel integrates naturally into neural operator architectures while retaining flexible point-set computation. We characterize the resulting operator from continuum and representation perspectives, and develop scalable realizations for large discretizations. Across several PDE learning settings, LINK improves predictive accuracy and optimization, with the largest gains arising when interactions are nonlocal. These results show that learning routing directly in physical space provides an effective inductive structure for neural operators on irregular domains.

1 INTRODUCTION

Neural operators approximate maps between functions (Kovachki et al., 2023), so every neural operator must determine how information at one physical location influences another. For parametric elliptic partial differential equations (PDEs) on a domain Ω, this interaction is part of the solution operator itself. A representative form is

$$u(x) = \int_\Omega G_a(x, y) f(y)\, \mathrm{d}y, \tag{1}$$

where f is the forcing and $G_a(x, y)$ describes how a source at y influences the response at x under the current coefficient field a. When a changes, the interaction kernel can change with the sample. Thus, a neural operator needs to represent both the information being transported and the instance-dependent relation between pairs of physical locations.

Existing neural operators place that relation in different objects. Graph Kernel Networks (GKN) evaluate a learned function of the physical pair and use the resulting matrix-valued kernel to transform messages (Li et al., 2020a;b). Attention-based operators normalize a scalar score computed from query and key embeddings; Continuum Attention places this construction in function space (Calvello et al., 2025), while the Hyperbolic Neural Operator (HNO) derives the score from hyperbolic distance between separately projected query and key endpoints (Pei et al., 2026). In these models, pairwise interaction is represented either by a matrix-valued message kernel or by separate query and key embeddings. This raises a basic question: *can a neural operator learn the instance-dependent interaction kernel of a PDE directly as its attention routing, and when is this useful?* We introduce LINK (Learned Interaction Kernels), which evaluates the scalar pair score jointly on the two current hidden states and their geometry, both relative and absolute, before row normalization, while a separate value map carries source content. Scoring the two hidden states jointly lets the pair score depend on the pair as a whole. We therefore focus on settings where the interaction depends on both the sample and the spatial pair.

The screened Poisson equation $(-\Delta + \kappa^2)u = f$, which we refer to as Poisson throughout, provides an exact reference: its Green operator is known, and κ sets the range of interaction. We compare the internal routing matrix P with the end-to-end Jacobian $J = \partial\hat{u}/\partial f$ of the prediction $\hat{u}$ with respect

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.

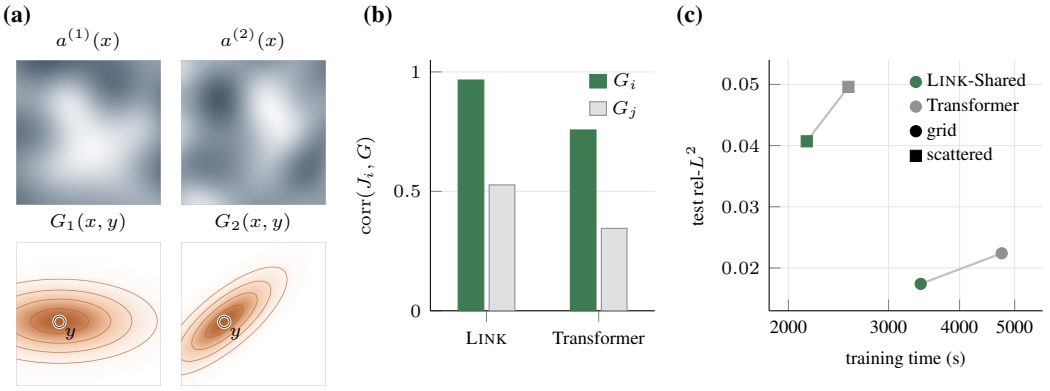

Figure 1: **The end-to-end operator of LINK tracks the Green operator of each sample more closely than the Transformer's, and LINK-Shared is more accurate and faster than the Transformer on Darcy.** (a) Illustrative coefficient fields and schematic Green functions for the same source location y. (b) Mean correlation of the Jacobian J_i with the Green operator of the same sample G_i and a mismatched sample G_j over 12 probes. (c) Three-seed Darcy results for LINK-Shared and the Transformer on grid and scattered discretizations; lower-left is better.

to the forcing: P records where a layer sends information, while J records the operator realized by the complete model. Training produces Green-like routing and a Jacobian closely aligned with the Green operator. Green-like routing also emerges without the Gaussian locality prior, and the learned operator adapts to the Green operator of the current sample, as illustrated in Figure 1.

We next vary the interaction range. The gap in Jacobian–Green correlation between LINK and a coordinate-augmented Transformer is largest for the most global operators. On variable-coefficient Darcy flow, where the interaction also changes from sample to sample, LINK achieves lower error on grid and scattered discretizations, and the Shared variant achieves both lower error and lower training time than the Transformer. The Landmark variant evaluates the same joint pair score on point sets of up to $65{,}536$ points.

Contributions. Our main contributions are as follows:

- We introduce LINK, which combines joint pair scoring with normalized routing: each pair score is learned from both hidden states and their geometry before row normalization, while values remain pointwise. The resulting routing matrix is an explicit object of the model.
- We establish two layer-level properties. A fixed continuous head has a continuum limit under refinement of point sets representing the same sampling measure. After removing the row-wise softmax gauge, a width-r bilinear head gives a row-centered log-kernel of rank at most r. A joint pair network is not subject to this fixed-width rank constraint.
- We test how the learned routing relates to the target operator and to task performance. On Poisson it develops Green-like routing, changes with the current sample in its early layers, and remains after removing the locality prior. Replacing a sample's routing with another sample's increases the prediction error, while the Jacobian-correlation gap to bilinear attention is largest for long-range interaction. On Darcy, matched-capacity controls isolate the contribution of the learned pair network and row normalization, LINK achieves lower error than GKN and a hyperbolic score following HNO at matched capacity, and the Dense, Shared, and Landmark variants give different accuracy–cost tradeoffs across discretizations and point-set sizes.

2 RELATED WORK

Physical-pair kernel operators. Graph Kernel Networks learn a matrix-valued integral kernel from the physical pair and the input values at both endpoints and apply it directly to the transported representation (Li et al., 2020a); the Multipole Graph Neural Operator (MGNO) factors the same operator view across scales using recursively introduced coarse points (Li et al., 2020b). These

methods establish joint pair kernels and multiscale approximations for operator learning. LINK uses the pair network to produce a scalar score, whose row normalization defines the routing, while the value transform remains separate.

Normalized attention operators. Scoring a pair of endpoints with a small MLP has a long history in attention: additive attention (Bahdanau et al., 2015), graph attention (Veličković et al., 2018; Brody et al., 2022), and point-cloud attention (Zhao et al., 2021) compute scores from learned features of both endpoints. LINK applies joint pair scoring to physical locations, conditioning the score on both hidden states and on their absolute and relative geometry, and studies the resulting routing as an operator-learning object. A related normalized-routing approach is HNO: it maps query and key endpoints separately into Lorentz space and converts their hyperbolic distance into a Gibbs routing kernel, with a separate value map (Pei et al., 2026). LINK instead evaluates both hidden states and their absolute and relative geometry jointly before producing the scalar pair score. Continuum Attention gives the function-space formulation of bilinear attention and its discretization (Calvello et al., 2025); Learning Operators with Coupled Attention (LOCA) combines attention with integral transforms (Kissas et al., 2022), while Galerkin and Fourier attention (Cao, 2021), the General Neural Operator Transformer (GNOT) (Hao et al., 2023), and Transolver (Wu et al., 2024) provide other global mixing constructions for irregular domains. Proposition 2 lower-bounds the approximation error of fixed-width *bilinear* softmax heads, and Appendix D compares the joint pair score directly with a hyperbolic score following HNO.

Physical interpretation and Green-function structure. The Nonlocal Attention Operator (NAO) interprets a data-dependent attention kernel in inverse physical problems and uses it for hidden-physics discovery (Yu et al., 2024). Boullé et al. (2022) directly learn Green functions for fixed linear operators, while Neural Green's Functions predict a geometry-dependent decomposition of the solution operator for linear PDEs (Yoo et al., 2025). LINK uses routing as an internal communication object and learns it end to end from input–output data. We use the known Green structure of Poisson to examine both the routing P and the end-to-end Jacobian J of the trained model.

Geometry, discretization, and scale. The Fourier Neural Operator (FNO) learns global convolution in a spectral basis on regular grids (Li et al., 2021); Geo-FNO extends this idea to varying geometries through learned deformation (Li et al., 2023a), and the Geometry-Informed Neural Operator (GINO) couples local graph integration with a latent Fourier grid for large 3-D problems (Li et al., 2023b). MeshGraphNet uses local message passing over an explicit mesh graph (Pfaff et al., 2021), while DeepONet represents operators through learned branch/trunk bases (Lu et al., 2021). Efficient attention methods reduce quadratic mixing through kernelization, random features, or landmarks (Katharopoulos et al., 2020; Choromanski et al., 2021; Xiong et al., 2021). LINK-Landmark brings the landmark idea to the joint pair score.

3 LINK: LEARNING THE PHYSICAL INTERACTION KERNEL

3.1 OPERATOR SETTING

We observe samples of an operator $\mathcal{G}$ that maps input fields on a compact domain $\Omega \subset \mathbb{R}^c$ to a solution u; depending on the task, the input is the coefficient field a, the forcing f, or both. A sample is represented on a point set $T = \{t_1, \dots, t_n\} \subset \Omega$ that may change between samples and between training and evaluation. A pointwise lift maps the input values at t, together with the location t, to a hidden state $z(t) \in \mathbb{R}^d$ at every $t \in T$; mixing layers are the only components that communicate across locations; and a pointwise readout produces the prediction $\hat{u}(t)$.

3.2 JOINT PAIR SCORING FOR ROUTING

For each target point $t \in T$, the model predicts a distribution over source points $t' \in T$ that depends on the current field. We learn a scalar score for the physical pair and normalize it across sources. For head h and pair (t, t'), define

$$\ell^{(h)}(t,t') = \phi_h([z(t), z(t'), t, t', t-t', \|t-t'\|]) - \frac{\|t-t'\|^2}{2\sigma_h^2}, \quad P^{(h)}(t,t') = \frac{e^{\ell^{(h)}(t,t')}}{\sum_{t'' \in T} e^{\ell^{(h)}(t,t'')}}. \tag{2}$$

Here $\phi_h : \mathbb{R}^{2d+3c+1} \to \mathbb{R}$ is a small multilayer perceptron (MLP), the pair network, of hidden width d_ϕ, σ_h is a learned bandwidth, and $V_h : \mathbb{R}^d \to \mathbb{R}^{d_v}$ is the value map. The head output at t is $\sum_{t' \in T} P^{(h)}(t, t')\, V_h z(t')$ inside a standard pre-norm residual block, and we write P for $P^{(h)}$ when the head is clear.

Softmax turns the pair scores into a distribution over source points, so the pair network determines where to route information, while $V_h z(t')$ determines what is transported. This separation also gives two ways to compare the learned interaction with a physical kernel: $P(t, t')$ measures the spatial routing pattern, while the end-to-end Jacobian J measures the resulting input-output operator. Because each row of P sums to one, it describes how information is distributed across source points; the value map and subsequent layers determine its magnitude and sign. We compare P with G as a measure of spatial structure, and compare the end-to-end Jacobian J with G as an operator. The separation of routing and values is standard in attention; LINK changes the score, giving one pair network access to both hidden states and their geometry before normalization. Absolute locations give the pair network access to boundaries and non-translation-invariant geometry, while relative displacement and distance give it direction and scale. This also distinguishes the pair score from a purely feature-based attention rule: two pairs with similar hidden states can receive different scores when their physical configurations differ. The score can therefore represent interactions that depend jointly on the current field and on where the two locations occur in the domain. The last term of Equation (2) is a Gaussian locality prior with a learned bandwidth, and Section 5.1 shows that the learned routing structure emerges without it.

The pair network ϕ_h depends only on hidden states and locations. A new discretization changes only the point pairs on which ϕ_h is evaluated, so the same pair network applies to new point sets. The explicit pair locations also make the routing directly comparable to a physical two-point kernel such as $G(x, y)$.

LINK variants. LINK-Dense evaluates all n^2 pairs. Its first pair-MLP layer factorizes into query, key, and distance contributions, so the full $n \times n$ concatenated tensor is never materialized. Query blocking computes complete softmax rows block by block; each row is still normalized over all keys, so blocking reduces memory without changing the layer output. LINK-Shared computes the learned pair-network contribution once and reuses it across layers, while each layer retains its own learned bandwidth and value map. For large point sets, LINK-Landmark uses m learned landmarks. A gather matrix $\Gamma \in \mathbb{R}^{m \times n}$ routes the stacked values $V \in \mathbb{R}^{n \times d_v}$, whose rows are $V_h z(t)$, to the landmarks and a redistribution matrix $\Lambda \in \mathbb{R}^{n \times m}$ routes them back. Both are row-normalized and scored by the same pair network, which gives the output $\Lambda(\Gamma V)$. Landmark routing is a distinct routing operator with $O(nm)$ pair evaluations.

Computational complexity. Like bilinear attention, LINK-Dense has quadratic cost in n, with the pair network replacing the inner product; LINK-Shared evaluates it once for all layers instead of once per layer, and LINK-Landmark evaluates $O(nm)$ pairs. Query blocking bounds the pair-activation memory by the block size, as we measure in Section 5.3 and Appendix K.

4 PROPERTIES OF LINK

We study two properties of the routing layer in Equation (2) for a single head. In Section 4.1, we show that a fixed routing head converges as the point set is refined, a layer-level consistency result relevant to point-set transfer, which Section 5.2 tests on the complete frozen model. In Section 4.2, we show that the joint pair score can represent normalized kernels that cannot be approximated by a width-r bilinear head, motivating the form of Equation (2).

4.1 CONTINUUM CONSISTENCY UNDER REFINEMENT

For a fixed physical sample and continuous hidden field, write the score in Equation (2) as $\ell(x, y)$ and the value field as $v(y)$. For a probability measure ν on Ω, define

$$(\mathcal{A}_\ell^\nu v)(x) = \frac{\int e^{\ell(x,y)} v(y)\, \mathrm{d}\nu(y)}{\int e^{\ell(x,y)}\, \mathrm{d}\nu(y)}. \tag{3}$$

The discrete row-softmax is Equation (3) with the empirical measure $\nu_n = n^{-1}\sum_{t\in T}\delta_t$, where δ_t is the Dirac measure at t, the same normalized integral form used in Continuum Attention (Calvello et al., 2025).

Proposition 1 (Continuum consistency of a fixed routing head)**.** *Let Ω be compact, let $\nu_n \Rightarrow \mu$ weakly for Borel probability measures ν_n, μ on Ω, and let $\ell \in C(\Omega\times\Omega)$ and $v \in C(\Omega;\mathbb{R}^{d_v})$. Then*

$$\sup_{x\in\Omega} \|(\mathcal{A}_\ell^{\nu_n} v)(x) - (\mathcal{A}_\ell^{\mu} v)(x)\| \longrightarrow 0. \tag{4}$$

The proof, given in Appendix A, uses weak convergence for each fixed query and compactness of the query-indexed family of integrands to make the convergence uniform. The result covers deterministic grids and i.i.d. random point sets whenever their empirical measures converge to the same μ. With nonnegative quadrature weights that sum to one, added to the scores as $\log w_t$, the statement extends to point sets drawn from a different sampling measure (Appendix A). The consistency result concerns the discretization of a fixed routing function; it does not require the routing to be bilinear or to have low rank. The second result addresses the complementary question of what routing functions can be represented by a fixed-width bilinear score.

4.2 Normalized routing beyond fixed-width bilinear scores

Fix a Borel probability measure μ on Ω. Softmax is invariant to adding an arbitrary query-only term, $\ell(x,y) \mapsto \ell(x,y)+\gamma(x)$, so the rank of raw pair scores depends on this choice of gauge. For a strictly positive routing kernel P on $\Omega\times\Omega$ normalized with respect to μ, that is, $\int_\Omega P(x,y)\,\mathrm{d}\mu(y) = 1$ for every x, we remove this row-wise gauge using the row-centered log-kernel

$$\widetilde{R}_P(x,y) = \log P(x,y) - \int_\Omega \log P(x,y')\,\mathrm{d}\mu(y'). \tag{5}$$

For a bilinear head of width r, with query and key embeddings $q, k : \Omega \to \mathbb{R}^r$,

$$P_{q,k}(x,y) = \frac{e^{q(x)^\top k(y)}}{\int e^{q(x)^\top k(y')}\,\mathrm{d}\mu(y')}, \qquad \widetilde{R}_{P_{q,k}}(x,y) = q(x)^\top\big(k(y)-\bar{k}\big), \quad \bar{k} = \int k(y')\,\mathrm{d}\mu(y'), \tag{6}$$

so the Hilbert–Schmidt operator on $L^2(\mu)$ induced by $\widetilde{R}_{P_{q,k}}$ has rank at most r.

Proposition 2 (Fixed-width separation after normalization)**.** *Let μ be a Borel probability measure on compact Ω, let $\mathcal{P}_r$ be the family of kernels $P_{q,k}$ of Equation* (6) *with continuous $q, k : \Omega \to \mathbb{R}^r$, and let $P^\star$ be strictly positive, continuous, and normalized with respect to μ. If the Hilbert–Schmidt operator on $L^2(\mu)$ with kernel $\widetilde{R}_{P^\star}$ has singular values $s_1 \ge s_2 \ge \cdots$ with $s_{r+1} > 0$, then*

$$\inf_{P\in\mathcal{P}_r} \|P - P^\star\|_\infty \ge \min\Big\{\frac{\eta}{2}, \frac{\eta\, s_{r+1}}{4}\Big\} > 0, \qquad \eta = \min_{x,y} P^\star(x,y), \tag{7}$$

where $\|\cdot\|_\infty$ is the supremum over $\Omega\times\Omega$.

The proof, given in Appendix A, compares centered log-kernels: every bilinear head gives rank at most r, the error of every rank-r approximation is at least s_{r+1}, and the logarithm is Lipschitz on kernels bounded below by $\eta/2$. The proposition captures one representational consequence of joint pair scoring: at fixed bilinear width, separate query and key embeddings retain a rank-r row-centered log factorization after normalization. A joint pair score can approximate any strictly positive continuous normalized kernel given sufficient MLP capacity, as we show in Appendix A. The comparison is for a single head of fixed width; adding width, heads, or depth, or using nonlinear factorized scores such as HNO's hyperbolic-distance kernel, yields families larger than $\mathcal{P}_r$. The separation comes from the learned joint pair score, not from the locality prior: modulo an x-only term that softmax cancels, the locality prior equals $x^\top y/\sigma_h^2 - \|y\|^2/(2\sigma_h^2)$ and has $c+1$ separable components.

5 Experiments

We evaluate LINK through four questions: (Q1) Does LINK learn the interaction kernel of the PDE when the Green operator is known? (Q2) Does the learned routing adapt to each sample, and does the

prediction use it? (Q3) When does joint pair scoring help most? (Q4) How does LINK compare with competitive neural operators in accuracy, transfer across discretizations, and cost?

Baselines. We compare LINK with (1) bilinear attention, a coordinate-augmented Transformer that receives the same physical tokens but scores the pair from separate query and key embeddings; (2) neural operator transformers, namely Galerkin attention (Cao, 2021), GNOT (Hao et al., 2023), and Transolver (Wu et al., 2024); and (3) pair-kernel operators, namely GKN (Li et al., 2020a) and a same-backbone hyperbolic score following HNO (Pei et al., 2026). Matched-capacity ablations replace the pair network by a Gaussian-only router and the softmax by the unnormalized rule e^{ℓ}/n. Appendix F gives data, models, training, and the hyperparameter selection protocol, and Appendices G and H give the selection results and every convergence-ladder budget.

Protocol. Every $\pm$ in this paper is a sample standard deviation over three independently seeded runs, except where a caption states a different number of seeds; tables without $\pm$ state what they report. Test error is the relative L^2 error on the test set, and matched capacity means equal parameter counts. Because the per-step computational cost differs across methods, we report both optimizer steps (E) and wall-clock training time. Because final error changes with the budget, each selected configuration is retrained from scratch at increasing budgets, a convergence ladder, and we compare models both at the budget with their lowest measured error and at a common number of steps. During hyperparameter selection, LINK additionally considers depths up to $L = 24$, which give the grid LINK configuration of Table 3 (Appendix G).

5.1 LINK ALIGNS WITH THE GREEN OPERATOR OF POISSON

For the Poisson equation $(-\Delta + \kappa^2)u = f$, we compare the model with the known Green operator at two levels: the row-stochastic routing P in each layer and the end-to-end Jacobian $J = \partial\hat{u}/\partial f$.

LINK aligns with the Green operator. As shown in Table 1, LINK learns the solution operator more accurately than the Transformer, and its Jacobian aligns more closely with the Green operator: LINK reaches a test error of 0.0044 ± 0.0001, compared with 0.0157 ± 0.0012 for the Transformer. The Jacobian–Green correlation is 0.984 ± 0.002 for LINK and 0.869 ± 0.003 for the Transformer. During training, routing becomes much more aligned with the Green operator: its strongest middle-layer correlation increases from about 0.38 at initialization to 0.82, and the learned routing develops a finite spatial decay scale of the same order as the physical kernel.

Table 1: LINK is more accurate and closer to G. Poisson, $\kappa = 8$, $n = 400$; two seeds (three without the prior). The top two results are marked by **First**, **Second**.

Method ↓ / metric →	test rel-L^2 ↓	corr(J, G) ↑
Transformer	0.0157±.0012	0.869±.003
LINK (Ours)	**0.0044±.0001**	**0.984±.002**
LINK (Ours, no prior)	**0.0050±.0004**	**0.974±.007**

The alignment is learned without the locality prior. When we train the same model without the locality prior of Equation (2), the test error remains low (0.0050 ± 0.0004 versus 0.0044 ± 0.0001; on the two seeds shared by both settings, the mean error without the prior is 0.0048), and $\mathrm{corr}(J, G)$ remains 0.974 ± 0.007. Removing the prior lowers the routing-Green correlation at initialization from 0.376 to 0.038 on the first seed, and over three seeds training raises the peak-layer correlation to a mean of 0.717. Training therefore recovers substantial Green alignment even without the explicit locality prior. Tables 21 and 24 in Appendix J contain the layerwise and seedwise measurements.

The learned operator adapts to each sample. When each sample has its own coefficient field, $(-\Delta + a^{(i)}(x))u = f$, so that G_i changes from sample to sample, the LINK Jacobian has mean correlation 0.966 with the Green operator of the same sample and 0.527 with that of another sample, across 12 probes, against 0.757 and 0.345 for the Transformer. The matched correlation is higher in all 12 cases ($p < 0.001$). The routing itself tracks the sample in the early layers: for layer 1, the matched–mismatched difference is 0.202, again with the matched value higher in all 12 probes, as reported in Table 20 in Appendix J.

The prediction uses the sample-specific routing. Swapping in another sample's routing increases the error at every layer, reaching about seven times the baseline error at the first layer, as shown in Table 10 in Appendix B. Reinjecting a sample's own routing reproduces its output to numerical precision, permuting the source points of the routing raises the error further, and donors with more

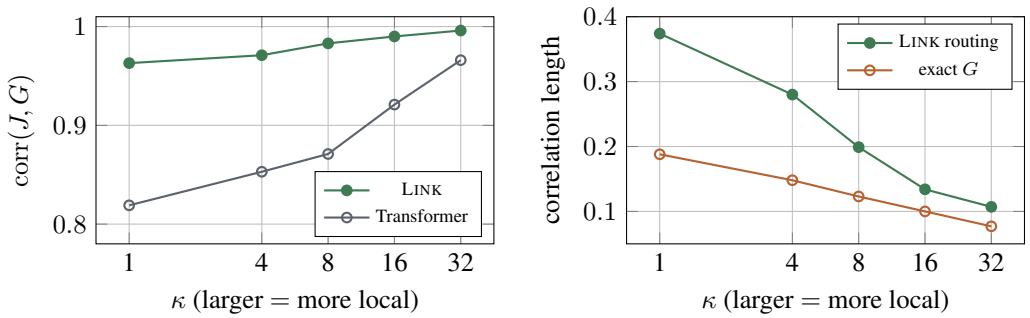

Figure 2: **The advantage of LINK is largest for global interaction, and its routing length tracks the Green function.** Curves show seed 0. Left: both models become more aligned with the exact Green operator as κ increases. Right: the spatial scale of LINK's routing contracts with that of the Green function. Table 22 in Appendix J reports both seeds and the full routing diagnostics.

Table 2: **Matched-capacity controls and comparisons.** The first two rows remove the learned pair network and row normalization; the last two compare with a hyperbolic score and with GKN. Test rel-L^2 on Darcy, three seeds; protocols in Appendices C, D and G. The better result in each row is marked with **First**.

Element ↓	Alternative	params	Alternative ↓	LINK (Ours) ↓
Pair network	Gaussian-only router	12.23M	0.1110±0.0022	**0.0522±0.0039**
Row normalization	unnormalized e^{ℓ}/n	0.406M	0.0721±0.0022	**0.0669±0.0021**
Score family	hyperbolic score, as in HNO	0.665M	0.0574±0.0049	**0.0491±0.0003**
Pair-kernel family	GKN	0.162M	0.1314±0.0007	**0.0460±0.0011**

dissimilar Green operators cause larger increases in the first two layers (Spearman $\rho = 0.52$ and 0.44).

The advantage is largest for global interaction. As illustrated in Figure 2, increasing κ shortens the range of the Green function, and both models align more closely with the Green operator as the problem becomes more local. Averaged over two seeds, the difference in Jacobian–Green correlation is 0.168 at $\kappa = 1$, where the interaction is most global, and 0.031 at $\kappa = 32$. The spatial scale learned by LINK changes with the PDE as well: its layer-2 correlation length contracts from 0.374 to 0.107, while the correlation length of the Green function changes from 0.188 to 0.077. The routing–Green alignment also remains similar across independently trained resolutions: at $n = 4096$ and $16{,}384$, the routing–Green correlation stays near 0.82, and the LINK Jacobian stays closer to G than the Transformer's at every n, as reported in Table 23 in Appendix J.

5.2 JOINT PAIR SCORING LOWERS ERROR ON DARCY

In Darcy, $-\nabla \cdot (a \nabla u) = f$, the interaction again changes with the coefficient field $a(x)$ for each sample.

Matched-capacity controls isolate the pair network and row normalization. As shown in Table 2, at matched capacity the learned pair network more than halves the error of the Gaussian-only router (0.0522 versus 0.1110), and row normalization lowers the error of the same pair network from 0.0721 to 0.0669. The pair-network contrast is measured at width $d = 384$, and Appendix G shows that it also holds at twice the budget. Replacing the joint pair score with a hyperbolic score following HNO raises the error from 0.0491 to 0.0574 on Darcy, as reported in Table 11 in Appendix D. Against GKN, the closest joint-pair predecessor, LINK reaches 0.0460 ± 0.0011 versus 0.1314 ± 0.0007 with $161{,}995$ versus $161{,}697$ parameters; GKN's mean error is 2.9 times that of LINK, and the per-seed error ranges do not overlap, as detailed in Appendix C.

LINK achieves lower error than the Transformer on both discretizations. As can be seen from Table 3, LINK reaches 0.0184 ± 0.0009 against 0.0214 ± 0.0012 on the grid and 0.0376 ± 0.0009 against 0.0496 ± 0.0004 on the scattered discretization, corresponding to relative improvements

Table 3: **LINK achieves lower error than the Transformer on both discretizations and with a fixed operator.** Darcy, three-seed convergence ladders, each model at the budget with its lowest measured error; scattered LINK is shown at the budget chosen during hyperparameter selection, and Appendix H lists every budget. The fixed-operator LINK uses two additional architectural options, multi-scale locality and a mixture-of-experts feed-forward block (Appendix F). The better result in each block is marked by **First**.

Task ↓	Method ↓	params	steps ↓	test rel-L^2 ↓	s/run ↓
grid	Transformer	0.67M	768,000	0.0214±0.0012	18,968
	LINK (Ours)	2.87M	24,000	**0.0184±0.0009**	**14,997**
scattered	Transformer	0.45M	384,000	0.0496±0.0004	**2,550**
	LINK (Ours)	40.6M	6,000	**0.0376±0.0009**	5,104
fixed operator	Transformer	0.15M	192,000	0.0250±0.0015	**2,315**
	LINK (Ours)	6.81M	96,000	**0.0192±0.0016**	24,871

Table 4: **LINK-Shared is more accurate than the Transformer in less training time.** Darcy, three-seed convergence ladders; each row is one ladder point. The top three results in the grid block and the better result in the scattered block are marked by **First**, **Second**, **Third**.

Task ↓	Method ↓	steps ↓	test rel-L^2 ↓	s/run ↓
grid	Transformer	192,000	0.0224±0.0013	**4,744**
	Transformer	768,000	**0.0214±0.0012**	18,968
	LINK-Shared (Ours), $L = 24$	24,000	**0.0174±0.0003**	**3,417**
	LINK-Shared (Ours), $L = 24$	96,000	**0.0143±0.0013**	**13,677**
scattered	Transformer	384,000	0.0496±0.0004	2,550
	LINK-Shared (Ours), $L = 8$	12,000	**0.0407±0.0012**	**2,154**

of 14% and 24%. At matched capacity, a 39.39M Transformer reaches 0.0508 against 0.0334 for 40.60M LINK, while the best Transformer of that sweep reaches 0.0454, as reported in Table 18 in Appendix I.1.

LINK achieves lower error than all evaluated neural operator baselines. At a common budget of 192,000 steps, LINK-Shared reaches 0.0157 on the grid and 0.0313 on the scattered discretization, against 0.0224 and 0.0503 for the Transformer, relative improvements of 30% and 38%, and against at least 0.0735 and 0.0829 for Galerkin attention, GNOT, and Transolver, as reported in Table 17 in Appendix H. LINK also achieves the lowest error during hyperparameter selection, as reported in Tables 15 and 16 in Appendix G.

Sharing across depth lowers the training time. As shown in Table 4, LINK-Shared is both more accurate and faster than the Transformer on both discretizations, and on the grid LINK achieves lower error in less training time as well. Sharing the pair-network output across layers reduces how often it is evaluated while keeping the routing function, which lowers the cost of each update. With more budget, the grid ladder reaches 0.0143, below the Transformer's lowest measured error of 0.0214, which is still improving. The per-layer comparison in Table 3 is a separate run set; Table 25 in Appendix K reports the full sharing study, in which a deeper shared model with $L = 24$ reaches 0.0313 on the scattered discretization, while Figure 3 in Appendix G gives the grid training-time frontier.

The trained model transfers across discretizations. We train each model on 512 points and evaluate the frozen model at all measured densities, including an independently sampled off-grid point set. As shown in Table 5, LINK has the lowest error at every density, falling from 0.1084 ± 0.003 at 512 points to 0.0822 ± 0.002 at 4096, and its error falls the most as the evaluation density grows: the $4096/512$ ratio is 0.76, against 0.86–0.91 for the other models.

LINK also achieves lower error with a fixed operator. The last block of Table 3 fixes one coefficient field $a^\star$ across the dataset while the forcing varies. LINK achieves lower error at every common

Table 5: **LINK has the lowest transfer error at every density, and its error falls the most as the density grows.** Models trained on 512 points, evaluated without parameter updates; "Off" is an off-grid 1024-point draw, and ratio is the 4096-point over the 512-point error. The top three methods are marked by **First**, **Second**, **Third**.

Method ↓ / eval. $n \rightarrow$	512	1024	2048	4096	Off	ratio ↓
Transformer	**0.1360±.007**	**0.1252±.007**	**0.1199±.007**	**0.1166±.007**	**0.1270±.007**	**.86**
Galerkin	**0.1846±.005**	**0.1717±.005**	**0.1680±.004**	**0.1627±.004**	**0.1714±.002**	**.88**
GNOT	0.1994±.014	0.1866±.013	0.1816±.013	0.1776±.012	0.1900±.012	.89
Transolver	0.2041±.007	0.1936±.009	0.1879±.008	0.1856±.009	0.1988±.010	.91
LINK (Ours)	**0.1084±.003**	**0.0949±.002**	**0.0874±.002**	**0.0822±.002**	**0.0979±.003**	**.76**

budget, with a margin of 23.8% at 96,000 steps and 23.2% between the lowest measured errors (0.0192 against 0.0250). The advantage therefore does not rely on an interaction that changes between samples; Table 12 in Appendix E gives the complete ladder.

LINK achieves the lowest error on elasticity at a common budget. On the elasticity benchmark of Li et al. (2023a), at a common 30,000 steps over four seeds, LINK reaches 0.0114 ± 0.0002 against 0.0150 ± 0.0011 for GNOT, the best-performing evaluated baseline, a relative improvement of 24% (Table 19 in Appendix I.2).

5.3 LANDMARK ROUTING SCALES TO LARGE POINT SETS

Table 6 shows that LINK-Landmark lowers the time and memory of each optimizer step at scale: in the Poisson systems probe, it takes 249.8 ms and 3.12 GB per step at $n = 16{,}384$, against 853.3 ms and 34.52 GB for the Transformer. Exact query blocking reduces representative Dense pair activations from 67.6 GB to 2.2 GB with unchanged outputs, as detailed in Appendix K.

On large Darcy, the test error of LINK-Landmark stays between 0.116 and 0.119 from 2,048 to 65,536 points, as reported in Table 7.

Table 6: **LINK-Landmark needs less memory than the Transformer, and less time at $n = 16{,}384$.** Per-step cost on Poisson (batch sizes in Appendix F). The top three results are marked with **First**, **Second**, **Third**.

	$n = 4096$		$n = 16{,}384$	
Method ↓	ms/step↓	GB↓	ms/step↓	GB↓
Transformer	**107.3**	**4.39**	**853.3**	34.52
GNOT	**14.2**	**1.99**	**24.9**	**3.94**
LINK-Dense (Ours)	3138.9	5.12	24701.0	**10.23**
LINK-Landmark (Ours)	**124.2**	**2.21**	**249.8**	**3.12**

Discussion. On Poisson, the routing aligns with the Green operator, adapts to each sample in its early layers, and is used by the prediction, without relying on the locality prior. On Darcy and elasticity, LINK achieves lower error than every evaluated baseline, and the learned pair network and row normalization each lower the error at matched capacity.

Table 7: **LINK-Landmark keeps its error flat up to 65,536 points.** Large Darcy, $m = 32$, mean of three seeds.

$n \rightarrow$	2,048	8,192	32,768	65,536
test rel-L^2	0.117	0.116	0.118	0.119
peak GB	10.7	21.3	42.6	42.6

6 CONCLUSION

We asked whether a neural operator can learn the instance-dependent interaction kernel of a PDE directly as its attention routing, and when this is useful. With a joint pair score over two hidden states and their geometry, LINK learns Green-like routing that adapts to each sample in its early layers. It achieves lower error than all evaluated baselines on Darcy and elasticity and transfers across discretizations, and on Poisson its Jacobian-correlation gain is largest for nonlocal interaction.

AI USE STATEMENT

Generative AI tools were used solely to support language editing, including correcting grammar and improving the readability, wording, and clarity of the manuscript. They were not used to formulate scientific ideas, develop the methodology, design experiments, interpret results, or construct scientific claims. All AI-assisted revisions were carefully reviewed to ensure that the original technical meaning was retained. The authors take full responsibility for the final manuscript and for all text, claims, and other materials produced with the assistance of generative AI.

ETHICS STATEMENT

The authors have read and adhered to the ICLR Code of Ethics. This work uses synthetically generated PDE data and a public benchmark and involves no human subjects or personal data, and we are not aware of ethical issues specific to it.

REPRODUCIBILITY STATEMENT

Appendix F specifies data, splits, configurations, the hyperparameter selection protocol, seeds, and budgets, and Appendix G reports the selection results; Appendices A, J and K provide proofs, mechanism diagnostics, and systems details. Exact query blocking is checked against direct dense evaluation, including gradient agreement. The code will be made publicly available upon acceptance.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015.

Nicolas Boullé, Christopher J. Earls, and Alex Townsend. Data-driven discovery of Green's functions with human-understandable deep learning. *Scientific Reports*, 12:4824, 2022. doi: 10.1038/s41598-022-08745-5.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations (ICLR)*, 2022.

Edoardo Calvello, Nikola B. Kovachki, Matthew E. Levine, and Andrew M. Stuart. Continuum attention for neural operators. *Journal of Machine Learning Research*, 26(300):1–52, 2025.

Shuhao Cao. Choose a transformer: Fourier or Galerkin. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with Performers. In *International Conference on Learning Representations (ICLR)*, 2021.

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 (ICML)*, 2023.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020.

Georgios Kissas, Jacob H. Seidman, Leonardo Ferreira Guilhoto, Victor M. Preciado, George J. Pappas, and Paris Perdikaris. Learning operators with coupled attention. *Journal of Machine Learning Research*, 23(215):1–63, 2022.

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.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv:2003.03485*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

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, 2023a.

Zongyi Li, Nikola Borislavov Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Prakash Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3D PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

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. doi: 10.1038/s42256-021-00302-5.

Jieyuan Pei, Zhuoxuan Li, Wei Li, Haobo Zhang, Jiawei Jiang, and Jianwei Zheng. Hyperbolic neural operator. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *International Conference on Machine Learning (ICML)*, 2024.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A Nyström-based algorithm for approximating self-attention. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Seungwoo Yoo, Kyeongmin Yeo, Jisung Hwang, and Minhyuk Sung. Neural Green's functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yue Yu, Ning Liu, Fei Lu, Tian Gao, Siavash Jafarzadeh, and Stewart Silling. Nonlocal attention operator: Materializing hidden knowledge towards interpretable physics discovery. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

ORGANIZATION OF THE APPENDIX

Table 8 maps each appendix to the part of the paper it supports.

Table 8: **Organization of the appendix.**

Appendix	Content	Supports
A	Proofs and assumptions of both propositions	Section 4
B	Routing intervention	Section 5.1
C	Matched-capacity comparison with GKN	Section 5.2
D	Score-family control	Section 5
E	Fixed-operator Darcy control	Section 5.2
F	Experimental setup: data, models, training, and hyperparameter selection	Section 5
G	Hyperparameter selection	Section 5.2
H	Convergence ladders at every measured budget	Section 5.2
I	Matched-capacity Transformer and elasticity	Section 5.2
J	Green-function diagnostics	Section 5.1
K	Cost, memory, and scale	Sections 3 and 5.3

A PROOFS AND SCOPE OF THE LINK PROPERTIES

Table 9 lists the assumptions behind both propositions and what each one excludes.

Table 9: Assumptions used by the main-text propositions. "Idealization" means that the assumption belongs to the analytical model and is not enforced by the complete trained model.

Assumption	Status	Consequence / boundary
Fixed continuous hidden field, giving a continuous pair score ℓ and value field v	analytical idealization	Proposition 1 is a one-head refinement result, not a convergence theorem for the complete trained model.
Empirical measures $\nu_n \Rightarrow \mu$	property of the point-set sequence	Refinement represents the same limiting sampling measure, or nonnegative quadrature weights that sum to one, added to the scores as $\log w_t$, recover it.
Strictly positive continuous target routing $P^\star$	theorem assumption	The row-centered log-kernel is well defined and $P^\star$ has a positive minimum; exact-zero kernels and the singular continuum Green kernel are outside the statement, while its discretization is covered (see Discrete kernels below).
Width-r bilinear head	baseline-family definition	Proposition 2 characterizes one bilinear head; more width, heads, depth, or nonlinear metrics enlarge the family.

Proof of Proposition 1. The proof has two steps: weak convergence gives convergence for each continuous test function, and compactness of the query-indexed test-function family upgrades it to uniform convergence in x. For one component of v, define

$$\psi_x(y) = e^{\ell(x,y)} v(y), \qquad \chi_x(y) = e^{\ell(x,y)}.$$

Because ℓ and v are continuous on compact domains, the maps $x \mapsto \psi_x$ and $x \mapsto \chi_x$ are continuous from Ω into $C(\Omega)$ equipped with the sup norm. Their images are therefore compact subsets of $C(\Omega)$. Weak convergence $\nu_n \Rightarrow \mu$ gives $\int w \, \mathrm{d}\nu_n \to \int w \, \mathrm{d}\mu$ for every $w \in C(\Omega)$. The integration functionals have operator norm one, so pointwise convergence of these functionals is uniform on each compact family: for any $\varepsilon > 0$, cover the family by finitely many sup-norm balls of radius ε, use weak convergence at the centers, and bound the two approximation terms by ε each, which gives $\sup_w |\int w \, \mathrm{d}\nu_n - \int w \, \mathrm{d}\mu| \le 3\varepsilon$ over the family for all large n. Hence the numerator and denominator

in Equation (3) converge uniformly in x. Because ℓ is bounded,

$$\inf_x \int e^{\ell(x,y)}\,\mathrm{d}\mu(y) \geq e^{-\|\ell\|_\infty} > 0,$$

and the same bound holds for every ν_n, because each ν_n is a probability measure. The numerators are bounded by $e^{\|\ell\|_\infty}\|v\|_\infty$ and the denominators are bounded below by $e^{-\|\ell\|_\infty}$ uniformly in x and n, so the quotients converge uniformly, which gives Equation (4). Vector-valued v follows componentwise. □

Quadrature weights. With nonnegative quadrature weights that sum to one and whose weighted empirical measures converge weakly to μ, Proposition 1 extends to point sets drawn from a different sampling measure. The weights then enter the softmax as $\log w_t$ added to the scores. The layer in Equation (2) uses uniform weights, which suffices when the point sets sample the same measure.

Proof of Proposition 2. For every $P \in \mathcal{P}_r$, Equation (6) writes $\widetilde{R}_P$ as a sum of at most r separable functions, so the corresponding Hilbert–Schmidt operator has rank at most r. Let $\widetilde{R}^\star = \widetilde{R}_{P^\star}$. Because $P^\star$ is continuous and strictly positive on the compact set $\Omega \times \Omega$, $\widetilde{R}^\star$ is bounded, so the integral operator $\mathcal{K}_{\widetilde{R}^\star}$ on $L^2(\mu)$ with kernel $\widetilde{R}^\star$ is Hilbert–Schmidt with $\|\mathcal{K}_{\widetilde{R}^\star}\|_{\mathrm{HS}} = \|\widetilde{R}^\star\|_{L^2(\mu\otimes\mu)}$; let $s_{r+1} > 0$ be its $(r+1)$-st singular value. The Eckart–Young–Mirsky theorem for the singular expansion gives

$$\|\widetilde{R}^\star - \widetilde{R}_P\|_{L^2(\mu\otimes\mu)} \geq s_{r+1}$$

for every $P \in \mathcal{P}_r$. Because μ is a probability measure, the sup norm dominates this L^2 norm, so $\|\widetilde{R}^\star - \widetilde{R}_P\|_\infty \geq s_{r+1}$. Let $\eta = \min P^\star > 0$. If $\|P - P^\star\|_\infty \geq \eta/2$, the bound in Equation (7) holds through its first term. Otherwise $P > P^\star - \eta/2 \geq \eta/2$ pointwise, and $\log$ is $(2/\eta)$-Lipschitz on $[\eta/2, \infty)$, which contains the values of both P and $P^\star$. Writing $\zeta = \log P - \log P^\star$,

$$\widetilde{R}_P(x,y) - \widetilde{R}^\star(x,y) = \zeta(x,y) - \int \zeta(x,y')\,\mathrm{d}\mu(y'),$$

so

$$\|\widetilde{R}_P - \widetilde{R}^\star\|_\infty \leq 2\|\zeta\|_\infty \leq \frac{4}{\eta}\|P - P^\star\|_\infty.$$

Combining the inequalities yields $\|P - P^\star\|_\infty \geq \eta\, s_{r+1}/4$, proving Equation (7). □

Discrete kernels. The proof uses only that μ is a probability measure, so Proposition 2 holds verbatim for the empirical measure ν_n, with $\Omega \times \Omega$ replaced by $T \times T$ and $\|\cdot\|_\infty$ by the entrywise maximum. This is the case that applies to our reference kernels. The continuum Green kernel of Poisson is singular on the diagonal and vanishes on a Dirichlet boundary, whereas the discrete Green matrix $G = A^{-1}$ of Appendix F is entrywise positive, so its row normalization satisfies the assumptions.

Approximation by joint pair scores. For a strictly positive continuous normalized target $P^\star$, $\ell^\star = \log P^\star$ is continuous on the compact product domain and its row-normalization is exactly $P^\star$. The locality prior $g_h(x,y) = -\|x-y\|^2/(2\sigma_h^2)$ in Equation (2) is continuous for every fixed $\sigma_h > 0$, and the inputs of the pair network contain the pair (x,y) and range over a compact set. By the universal approximation theorem for MLPs with a continuous non-polynomial activation, the pair network can therefore approximate $\ell^\star - g_h$ uniformly, and the full LINK score approximates $\ell^\star$. If scores $\ell_m \to \ell^\star$ uniformly, the kernels obtained by exponentiation and normalization with respect to μ converge uniformly to $P^\star$, by the quotient argument in the proof of Proposition 1. The approximation assumes sufficient MLP capacity.

Gaussian locality prior. The locality prior expands as

$$-\frac{\|x-y\|^2}{2\sigma_h^2} = -\frac{\|x\|^2}{2\sigma_h^2} - \frac{\|y\|^2}{2\sigma_h^2} + \frac{x^\top y}{\sigma_h^2}.$$

The first term depends only on the query location and disappears under row normalization. The remaining normalized routing is therefore generated by c coordinate products plus one key-only term, i.e. by a bilinear score of width $c+1$. With $q(x) = (x/\sigma_h, 1)$ and $k(y) = (y/\sigma_h, -\|y\|^2/(2\sigma_h^2))$, the locality prior alone therefore gives a kernel in $\mathcal{P}_{c+1}$, and in every $\mathcal{P}_r$ with $r \geq c+1$ ($c+1 = 3$ on our two-dimensional domains); the larger routing family comes from the learned joint pair score.

Scope. Proposition 1 concerns a fixed continuous hidden field and point-set measures that converge to the same limit; the transfer experiments of Section 5.2 measure the complete trained model. Proposition 2 concerns a single width-r bilinear head and characterizes the routing kernels that the two score families can represent; wider, multi-head, and deeper Transformers, and nonlinear factorized scores such as HNO's hyperbolic score, form larger families. The accuracy gains are established empirically in Section 5.

B ROUTING INTERVENTION

This section details the routing intervention of Section 5.1. For a test sample i with routing P_i at one layer and a donor sample j, we replace P_i at that layer with

$$(1-\alpha)P_i + \alpha P_j, \qquad \alpha \in \{0, 0.25, 0.5, 0.75, 1\}.$$

Both samples share the same 20×20 grid, so P_i and P_j have the same shape, and the mixture stays row-stochastic. The values, hidden states, model weights, and all other layers remain those of sample i. We first record the donor routing in a separate forward pass and then substitute it during the forward pass of sample i.

We use two controls. First, replacing the routing with the sample's own routing must leave the output unchanged; this provides a direct check of the intervention. Second, we randomly permute the columns of P_i with the same permutation for all rows. This preserves the row sums and the overall distribution of routing weights, but breaks the correspondence between source points. It therefore tests whether the spatial assignment of the weights matters, rather than just their distribution.

Donors are sampled randomly from the test set, with 50 donor pairs per seed. Repeating the experiment with 100 pairs changed the reported values by about 3%.

Table 10: Routing intervention on Poisson with per-sample coefficients ($n = 400$, three seeds, 50 donor pairs). Entries are the change in test rel-L^2 relative to the same donor pairs at $\alpha = 0$, whose mean error is 0.0089 ± 0.0009. The identity control reproduced the unmodified output with zero difference in all twelve layer–seed cells.

Layer ↓ / intervention →	$\alpha = 0.5$	$\alpha = 1$	permuted	identity
1	$+0.0240 \pm 0.0076$	$+0.0545 \pm 0.0146$	$+0.2072 \pm 0.0185$	0.0
2	$+0.0092 \pm 0.0060$	$+0.0232 \pm 0.0135$	$+0.2450 \pm 0.0911$	0.0
3	$+0.0027 \pm 0.0019$	$+0.0085 \pm 0.0049$	$+0.0388 \pm 0.0252$	0.0
4	$+0.0144 \pm 0.0028$	$+0.0341 \pm 0.0058$	$+0.0368 \pm 0.0018$	0.0

Table 10 shows that the error increases with α at every layer, so the substituted routing affects the prediction. At layer 1, replacing the routing with that of another sample increases the error to about seven times the base error. The permutation control has a larger effect at every layer, consistent with the fact that it breaks the spatial correspondence. Layer 3 is relatively insensitive to both interventions, while the layer 2 permutation result has the largest variation across seeds. The prediction therefore depends on the routing, including the spatial assignment of its weights.

Stratification by Green similarity. For each donor pair, we measure the Green dissimilarity $1 - \mathrm{corr}(G_i, G_j)$ between the exact Green operators of the recipient and donor, using the flattened, row-centered operators. The Spearman correlation between Green dissimilarity and the error increase at $\alpha = 1$ is $+0.518 \pm 0.050$ at layer 1 and $+0.436 \pm 0.087$ at layer 2, over three seeds, and falls to $+0.220 \pm 0.169$ and $+0.099 \pm 0.252$ at layers 3 and 4. Splitting the pairs at the median, the more dissimilar half causes a larger error increase in every seed at layers 1 and 2. The dependence on the donor is therefore carried by the early layers, where the swap also has the largest effect.

C MATCHED-CAPACITY PAIR-KERNEL COMPARISON

The last row of Table 2 compares LINK and GKN on scattered Darcy at nearly identical parameter counts. LINK uses $161{,}995$ parameters and GKN $161{,}697$, a difference of 0.18%. Depth and head

count are fixed, with width adjusted to match the parameter count. Both models use three seeds and the same budget, $E = 24{,}000$. Following the convergence-ladder protocol of Section 5, Table 2 reports the GKN value from its ladder at this budget; in the matched-capacity tuning itself, GKN reaches 0.1708 with its learning rate at the top of the grid. The per-seed test errors are 0.0469, 0.0448, and 0.0462 for LINK, and 0.1316, 0.1320, and 0.1307 for GKN.

GKN reaches 0.1314 at $E = 24{,}000$ and 0.1304 by $E = 48{,}000$, with slightly higher error at larger budgets. The matched-width LINK model is still improving: validation error decreases throughout the rate sweep, and its lowest measured error occurs at the largest rate and budget tested, giving 0.0460 as its lowest measured error. At this budget, LINK takes $15{,}291$ s per run against $7{,}312$ s for GKN.

D SCORE-FAMILY CONTROL

This control supports the score-family comparison in Section 5. It compares the joint pair score of LINK with a factorized hyperbolic score following HNO (Pei et al., 2026) within the same backbone. The hyperbolic arm projects each endpoint separately into Lorentz space and routes with a Gibbs kernel based on their hyperbolic distance. The backbone, data, split, metric, optimizer, schedule, budget, and learning-rate grid are shared; the locality prior is off in both arms, and both arms have $664{,}641$ parameters ($291{,}488$ in the score module). On Darcy, where both arms select an interior learning rate, the joint pair score is 14% more accurate, with no overlap between the seed ranges. On Poisson, every run of both arms selects the largest learning rate of the shared grid, so those values are upper bounds on the error each arm can reach; both arms achieve low error there.

Table 11: **Same-backbone score-family control.** Test rel-L^2, mean$\pm$std over three seeds, at matched capacity. Poisson runs select the largest learning rate of the shared grid. The better score in each block is marked with **First**.

Task	Score	test rel-L^2 $\downarrow$	selected lr
Darcy	Hyperbolic score, as in HNO	0.0574$\pm$0.0049	1×10^{-3}
	LINK joint score (Ours)	**0.0491$\pm$0.0003**	1×10^{-3}
Poisson	Hyperbolic score, as in HNO	**0.0114$\pm$0.0010**	4×10^{-3}
	LINK joint score (Ours)	0.0162$\pm$0.0025	4×10^{-3}

E FIXED-OPERATOR DARCY CONTROL

The control behind Section 5.2 uses one Darcy coefficient field $a^\star$ for the whole dataset while the forcing varies per sample. Table 12 reports three seeds for each cell, with the same budget ladder for both models.

Table 12: Fixed-operator Darcy control at the common measured budgets. Test rel-L^2 $\downarrow$, three seeds per cell. The better result in each row is marked with **First**.

$E \downarrow$ / model $\rightarrow$	Transformer $\downarrow$	LINK (Ours) $\downarrow$	margin $\uparrow$
12,000	0.0358$\pm$0.0025	**0.0249$\pm$0.0008**	+30.4%
24,000	0.0301$\pm$0.0019	**0.0211$\pm$0.0012**	+29.9%
48,000	0.0266$\pm$0.0017	**0.0195$\pm$0.0012**	+26.7%
96,000	0.0252$\pm$0.0013	**0.0192$\pm$0.0016**	+23.8%

The Transformer was also continued to $192{,}000$ steps, where it reaches its lowest measured error of 0.0250 ± 0.0015 in $2{,}315$ s; the lowest measured LINK error is 0.0192 ± 0.0016 at $96{,}000$ steps in $24{,}871$ s. Table 12 uses the fixed-operator configurations of Table 13, with LINK-Dense at width 256 and 3 layers versus a Transformer at width 64 and 3 layers. The fixed-operator LINK uses two additional architectural options, multi-scale locality and a mixture-of-experts feed-forward block (Appendix F). The control shows that the advantage of LINK does not depend on an interaction that changes between samples.

F EXPERIMENTAL SETUP

This section collects the data, models, training, and hyperparameter selection settings used in the experiments. Throughout, E denotes the number of optimizer steps, and every $\pm$ is a sample standard deviation over independently seeded runs.

Poisson. We solve $(-\Delta+\kappa^2)u = f$ with Dirichlet boundary conditions on the $N \times N$ interior grid of the unit square, with grid spacing $1/(N+1)$ and the five-point stencil; $N = 20$ gives $n = 400$ points on a regular grid. Forcings f are Gaussian random fields with spectral slope 3. The Green operator $G = A^{-1}$ of the discrete operator A is formed by dense inversion, so the target $u = Gf$ is exact. The interaction range is set by $\kappa \in \{1, 4, 8, 16, 32\}$, with $\kappa = 8$ as the reference setting, and each task uses 512 training and 64 test samples. The landmark-count study of Appendix K uses $\kappa = 20$ and $E = 2000$. For the instance-dependent experiments of Figure 1b and Tables 10 and 20, sample i solves $(-\Delta + a^{(i)}(x))u = f$ with its own coefficient field $a^{(i)}$, a Gaussian random field mapped through a logistic function into $[1, 400]$, and into $[1, 8000]$ for Table 20; the model receives f and the standardized $\log a^{(i)}$, and $G_i = G_{a^{(i)}}$ is again an exact dense inverse. The routing-intervention runs of Table 10 keep these counts, and the input-dependence study of Table 20 uses 768 training and 64 test samples. The refinement study of Table 23 retrains both models from scratch at $N = 20$, 64, and 128 ($n = 400$, 4096, and 16384), with 256 training samples, learning rate 10^{-3}, and pair-network width $d_\phi = 128$.

Green-function diagnostics. The Jacobian $J = \partial\hat{u}/\partial f$ is evaluated on a subset of probe rows, 24 in the reference runs and 64 in the refinement study, by automatic differentiation, one backward pass per probe row at a single input sample. $\mathrm{corr}(J, G)$ is the Pearson correlation between J and G on those rows, and the routing–Green correlation compares G with the routing matrix of each layer after averaging over heads and without symmetrizing P. Layers are numbered from 1. The peak-layer correlation of a seed is its trained correlation at its peak layer, the layer where it is highest, and the improvement from training subtracts the untrained correlation at that same layer; the peak layer can differ across seeds. The correlation length is the distance at which the mean absolute kernel value, binned by source–target distance over 24 bins out to 0.75 and excluding the self-pair, decays to $1/e$ of its first bin, and the far/near ratio in Table 24 is the last bin of that profile divided by the first. The input-dependence study uses 12 test samples as probes; each probe's Jacobian or routing is compared with its own Green operator and with that of a different sample, and p-values come from a two-sided sign test over the 12 probes.

Variable-coefficient Darcy. We solve $-\nabla \cdot (a\nabla u) = f$ on an $N \times N$ grid with $N = 32$ and Dirichlet boundary conditions $u = 0$. Coefficient fields are log-normal, $a = \exp(g \log 4)$ with g a Gaussian random field of spectral slope 3, and the forcing is fixed across samples; reference solutions come from conjugate gradients with 400 iterations and tolerance 10^{-8}. The model receives the log-permeability $\log a$, and the target is standardized. The scattered discretization uses 1024 points per sample drawn by taking, for each sample, an independent uniform random subset of the 4096 points of a finer 64×64 solve. Dataset sizes: 1000 training and 200 test samples, with a validation holdout of 20% of the training split used for hyperparameter selection. The frozen-transfer study of Table 5 trains on 512 points and evaluates at 1024, 2048, and 4096 points and on an independently drawn off-grid set of 1024 points; the on-grid sets are independent uniform random subsets of the same solve, drawn with a seed distinct from the training subset, and the off-grid set is drawn at arbitrary continuous locations with values interpolated bilinearly off the grid. The fixed-operator control shares one coefficient field $a^\star$ across the dataset, drawn once from the same log-normal distribution as the coefficients of the varying-coefficient task, so that the shared operator is a typical member of that family rather than a special case, and varies the forcing per sample, Gaussian random fields of spectral slope 3 drawn from a seed stream disjoint from the one that produced $a^\star$, with the grid, solver, coefficient statistics, and sample counts of the main task. The large-Darcy runs of Table 7 use Darcy solves generated on a 256×256 grid under unit forcing, with a log-normal permeability field and conjugate gradients at 600 iterations, from which $n \in \{2048, 8192, 32768, 65536\}$ points are drawn uniformly at random in the unit square and the coefficient and solution interpolated bilinearly off that grid; 128 training and 32 test samples, and a separate model trained from scratch at each n for 800 steps over three seeds, at width 96, 4 layers, 8 heads, and $d_\phi = 128$.

Elasticity. We use the elasticity benchmark of Li et al. (2023a), with 972 points per sample and the fixed 1000/200 train/test split of that benchmark. Each model is tuned on a validation split, retrained on the full training set, and evaluated on the untouched test set at $E = 30{,}000$ over four seeds.

Models. Table 13 lists the configurations selected for the main comparisons. The Poisson LINK uses width 96, 8 heads (head width 12), 4 routing layers, pair-network width $d_\phi = 64$, and learning rate 2×10^{-3}; the Poisson Transformer uses width 96, 8 heads, and 4 layers. The pair-network control uses width 384, 8 heads, 8 layers, $d_\phi = 64$, batch size 16, learning rate 3×10^{-4}, and query blocks of 64. The normalization control has 406,273 parameters in both arms, at width 64, 8 layers, and 4 heads, with $d_\phi = 64$, batch size 4, and query blocks of 64. The matched GKN comparison fixes depth and head count and solves for the width, giving LINK width 24 with $d_\phi = 108$ and 4 heads, and GKN width 64 with kernel width 32 and 32 neighbours, both at 6 layers. The fixed-operator models are LINK-Dense with width 256, 3 layers, 4 heads, $d_\phi = 64$, the locality prior off, weight decay 10^{-3}, and batch size 16, and a Transformer with width 64, 3 layers, 8 heads, weight decay 10^{-5}, and batch size 16. The fixed-operator LINK also uses two additional architectural options. Multi-scale locality initializes the per-head bandwidths σ_h on a ladder instead of a single shared value, so that some heads start with a sharply local prior. The mixture-of-experts feed-forward block replaces the position-wise MLP by three expert MLPs behind a linear gate on the hidden state. Both are hyperparameters selected on validation data rather than fixed choices. LINK-Landmark uses $m = 32$ landmarks on large Darcy; the gather matrix Γ and the redistribution matrix Λ of Section 3 share the pair network, and each is normalized over its own source index. The reported runs use static landmark locations, learned as free parameters. On elasticity, hyperparameter selection chooses LINK at width 192 and 4 layers (5,242,253 parameters, learning rate 1×10^{-3}), Transolver at 192 and 6 (2,286,265, 2×10^{-3}), Galerkin at 192 and 6 (2,670,337, 1×10^{-3}), GNOT at 192 and 6 (4,448,461, 2×10^{-3}), and the Transformer at 128 and 6 (1,190,017, 6×10^{-4}). The score-family control of Appendix D uses the scattered discretization at $E = 24{,}000$.

Table 13: **Selected configurations.** Width d, depth L, parameter count, selected learning rate, and selection budget E. Learning rates marked * sit at an edge of the learning-rate grid; depth ext. marks the LINK-only depth extensions, and the grid LINK (depth ext.) row is the configuration of Table 3.

Task	Method	d	L	params	lr	E
Grid Darcy	Transformer	96	6	671,329	$3 \times 10^{-4}{}^*$	96,000
	Transolver	128	2	334,705	4×10^{-3}	96,000
	Galerkin	192	6	2,670,913	1×10^{-3}	96,000
	GNOT	384	3	8,872,135	2×10^{-3}	48,000
	LINK (depth ext.)	96	24	2,870,497	6×10^{-4}	48,000
	LINK (shared space)	96	6	1,977,811	2×10^{-3}	12,000
	LINK-Shared	96	24	2,870,497	6×10^{-4}	48,000
Scattered Darcy	Transformer	96	4	447,649	$3 \times 10^{-4}{}^*$	96,000
	Transolver	192	6	2,286,265	4×10^{-3}	48,000
	Galerkin	256	3	2,370,433	$3 \times 10^{-4}{}^*$	24,000
	GNOT	384	3	12,417,994	4×10^{-3}	48,000
	GKN	64	6	161,697	2×10^{-3}	12,000
	LINK	384	8	40,600,857	$3 \times 10^{-4}{}^*$	6,000
	LINK-Shared, $L = 8$	384	8	40,600,857	$3 \times 10^{-4}{}^*$	6,000
	LINK-Shared, $L = 24$ (depth ext.)	64	24	1,218,433	$8 \times 10^{-3}{}^*$	96,000
Fixed operator	Transformer	64	3	150,145	1×10^{-3}	96,000
	LINK-Dense	256	3	6,808,150	$3 \times 10^{-4}{}^*$	12,000

Training. All models are trained with AdamW, a cosine learning-rate schedule over the full budget, and gradient-norm clipping at 1.0. Each optimizer step draws one random batch, so E counts optimizer steps; the Poisson runs use $E = 4000$. Weight decay is selected for each family from $\{0, 10^{-5}, 10^{-4}, 10^{-3}\}$, and from $\{0, 10^{-5}, 10^{-4}\}$ in the low-cost space; on scattered Darcy the selected values are 10^{-5} for LINK and 10^{-3} for the Transformer, and the capacity sweep of Appendix I.1 fixes it at 10^{-5} for every arm. Batch size is 16 for the fixed-operator and pair-network

controls, 8 for Poisson, and selected from $\{4, 8, 16\}$ elsewhere, with $\{8, 16\}$ in the low-cost space. Training loss: the same relative L^2 error used as the metric, computed per sample and averaged over the batch. No reported run uses learning-rate warmup; the cosine schedule decays from the first step. Each budget of a convergence ladder is a separate run retrained from scratch with the selected configuration.

Hyperparameter selection. Each model family draws its configurations from one shared randomized space, with width in $\{64, 96, 128, 192, 256, 384\}$, 4 or 8 heads, learning rate in $\{3, 6\} \times 10^{-4}$ and $\{1, 2, 4, 8\} \times 10^{-3}$, extended to 3.2×10^{-2} for GKN, the weight-decay and batch-size grids above, $d_\phi \in \{64, 128, 256\}$, family-specific parameters in place of width and depth where a family does not use them, and depth in $\{2, 3, 4, 6, 8\}$, which stops at depth $L = 8$; LINK additionally receives the depth extensions described in Appendix G. Each configuration is trained once on the training split minus a validation holdout of 20% and selected by validation error. Configurations whose projected training time exceeds six hours are excluded before full training, and runs that collapse to predicting the mean (rel-$L^2 > 0.9$) are excluded. Table 14 gives the number of configurations per family.

Table 14: **Size of the hyperparameter selection.** Usable / excluded for projected runtime / collapsed (predicting the mean) configurations per family and hyperparameter space. Depth extensions are LINK-only.

Space	Transformer	Transolver	Galerkin	GNOT	GKN	LINK
Grid, shared	34 / 0 / 12	40 / 0 / 0	40 / 0 / 0	47 / 0 / 0	–	37 / 13 / 0
Grid, depth ext.	–	–	–	–	–	47 / 1 / 0
Scattered, shared	67 / 0 / 36	95 / 0 / 0	98 / 0 / 0	104 / 0 / 0	51 / 66 / 0	53 / 32 / 0
Scattered, depth ext. 1	–	–	–	–	–	47 / 1 / 0
Scattered, depth ext. 2	–	–	–	–	–	12 / 6 / 0
Scattered, shared kernel	–	–	–	–	–	54 / 6 / 0
Fixed operator	14 / 0 / 6	20 / 0 / 0	20 / 0 / 0	20 / 0 / 0	–	18 / 2 / 0
Elasticity, low-cost	29 / 0 / 11	40 / 0 / 0	40 / 0 / 0	40 / 0 / 0	–	40 / 0 / 0

Metrics and timing. Test error is the relative L^2 error $\|\hat{u} - u\|_2 / \|u\|_2$, computed per sample and averaged over the test set. Training time (s/run) is the measured time of one complete training run on a single NVIDIA RTX 6000 Ada Generation GPU in full single precision, under PyTorch 2.0.1 with CUDA 11.7. Runs compared by training time within a table share the same hardware and software setup. Training times were measured separately for each table, so wall-clock comparisons should be made only within the same table. The systems probe of Table 6 reports time per optimizer step and peak memory at width 96, 4 layers, batch size 8 at $n = 4096$ and 4 at $n = 16{,}384$, and single precision.

G HYPERPARAMETER SELECTION ON DARCY

The results of hyperparameter selection are kept separate from the three-seed retraining tables in the main paper. Each configuration is selected on validation data and its test error is reported alongside; the selection tables below, Tables 15 and 16, are single-seed measurements, whereas the main-text comparisons in Table 3 retrain selected configurations on the full training set.

All model families share one hyperparameter space, which stops at depth $L = 8$. LINK additionally received three depth extensions up to $L = 24$, and the grid LINK configuration in Table 3 comes from one of them. Several selected learning rates lie at the lower limit 3×10^{-4} of the shared grid: the Transformer on both discretizations, Galerkin and LINK on the scattered discretization, and LINK in the fixed-operator control. For the 0.45M scattered Transformer, the extended grid of the capacity sweep, which reaches 6×10^{-5}, again selects 3×10^{-4}, as we show in Appendix I.1.

Table 15: **Grid Darcy hyperparameter selection.** Validation-selected test error and training time for the model families used in the comparative study, single seed. The top three results are marked by **First**, **Second**, **Third**.

Method ↓ / metric →	val ↓	test ↓	steps ↓	s/run ↓
Transformer	**0.0275**	**0.0286**	96,000	2,371
Transolver	0.0744	0.0833	96,000	575
Galerkin	0.0884	0.0993	96,000	1,189
GNOT	0.0935	0.0971	48,000	383
LINK (Ours) (selected)	**0.0213**	**0.0230**	12,000	7,521
LINK (Ours) (nearer matched time)	**0.0214**	**0.0232**	12,000	3,000

Table 16: **Scattered Darcy hyperparameter selection.** Validation-selected test error and training time, single seed. The top three results are marked by **First**, **Second**, **Third**.

Method ↓ / metric →	val ↓	test ↓	steps ↓	s/run ↓
Transformer	**0.0573**	**0.0559**	96,000	630
Transolver	0.0761	0.0796	48,000	625
Galerkin	0.1079	0.1084	24,000	148
GNOT	0.1060	0.1049	48,000	410
LINK (Ours) (selected)	**0.0432**	**0.0430**	6,000	5,127
LINK (Ours) (nearer matched time)	**0.0566**	**0.0537**	6,000	1,505

Closest pair-kernel predecessor. Under the scattered-Darcy sampler, split, metric, selection rule, and time budget used for LINK, GKN reaches 0.1304 ± 0.0020 by $48{,}000$ steps. The feasible GKN models are much smaller than the selected LINK configuration, so this comparison locates the method relative to the closest pair-kernel predecessor; Appendix C gives the matched-capacity comparison.

Pair-network and normalization controls. At width $d = 384$ and $12{,}000$ steps, the learned pair network reaches 0.0522 ± 0.0039 versus 0.1110 ± 0.0022 for the learned-bandwidth Gaussian-only replacement, with $12{,}233{,}985$ parameters in both arms. At twice the budget, $E = 24{,}000$, the contrast persists: 0.0509 ± 0.0025 against 0.0952 ± 0.0022. In the routing-normalization control, all inputs, pair-network parameters, value maps, data, seeds, and training budget ($E = 6{,}000$) are held fixed, and each arm's learning rate is tuned separately: row-softmax routing gives 0.0669 ± 0.0021 and the positive unnormalized rule $e^{\ell(t,t')}/n$ gives 0.0721 ± 0.0022, with $406{,}273$ parameters in both models.

Grid-Darcy training-time frontier. Figure 3 shows the complete measured frontier used to contextualize Table 4. The Transformer occupies the lowest-cost region, while the LINK configurations move below the Transformer error at larger budgets.

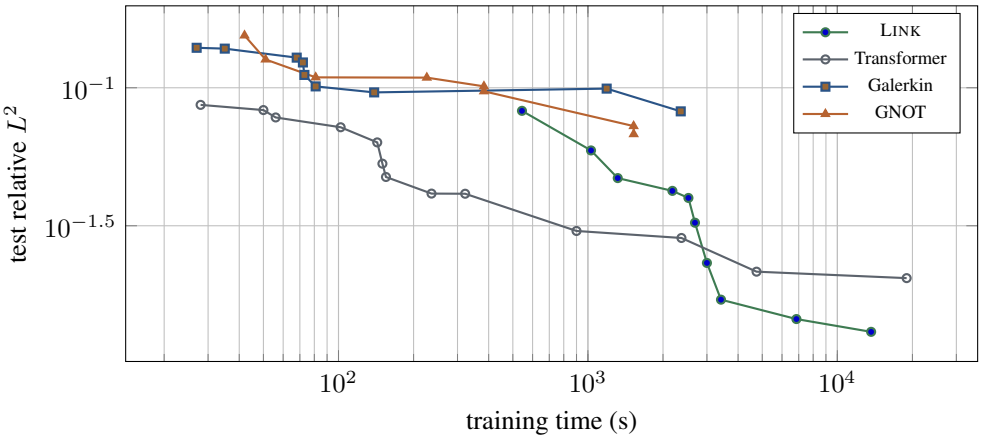

Figure 3: **Accuracy–training-time frontier on grid Darcy.** Lower-left is better. The frontier is built from the single-seed hyperparameter selection runs: points are selected on validation error and shown with their test error, so a configuration with higher validation error and lower test error can lie below the value selected in Table 15. The three-seed operating points quoted in the main paper are reported in Tables 4 and 17.

H CONVERGENCE LADDERS AT EVERY MEASURED BUDGET

Final error depends on the number of optimizer steps, and the models reach their lowest errors at very different budgets, so we report two views. Table 17 lists every budget with three completed seeds. Table 3 reports each model at the budget with its lowest measured error, except the scattered LINK row, which uses the 6,000 steps chosen during hyperparameter selection. The 192,000-step rows give a common-budget comparison: LINK-Shared reaches 0.0157 on the grid against 0.0224 for the Transformer and at least 0.0735 for the other baselines, and 0.0313 on the scattered discretization against 0.0503 and at least 0.0829. The per-layer LINK ladders stop earlier, and their lowest errors, at 24,000 and 12,000 steps, are already below every baseline at 192,000 steps. Both Transformer ladders are still improving at their largest budget (marked †), so their lowest measured errors may overstate the error they would reach.

Table 17: **Darcy convergence ladders.** Test rel-L^2, mean±std over three seeds, and training time per run at every budget with three completed seeds. LINK rows are ours. The 192,000-step rows give the common-budget comparison. †Still improving at the largest budget. The top three results in each panel are marked by **First**, **Second**, **Third**.

(a) Grid

Method ↓	steps	test rel-L^2 ↓	s/run
Transformer	192,000	0.0224±0.0013	4,744
	768,000	0.0214±0.0012†	18,968
Transolver	192,000	0.0839±0.0189	1,098
Galerkin	192,000	0.0903±0.0086	2,356
GNOT	192,000	0.0735±0.0060	1,526
LINK (Ours)	12,000	0.0214±0.0001	7,497
	24,000	0.0184±0.0009	14,997
	48,000	0.0189±0.0002	29,996
LINK-Shared	24,000	**0.0174±0.0003**	3,417
	48,000	**0.0157±0.0023**	6,845
	96,000	**0.0143±0.0013**	13,677
	192,000	**0.0157±0.0017**	27,560

(b) Scattered

Method ↓	steps	test rel-L^2 ↓	s/run
Transformer	192,000	0.0503±0.0011	1,280
	384,000	0.0496±0.0004†	2,550
Transolver	192,000	0.0829±0.0020	2,505
Galerkin	192,000	0.1034±0.0045	1,126
GNOT	192,000	0.1069±0.0083	3,134
LINK (Ours)	6,000	0.0376±0.0009	5,104
	12,000	0.0365±0.0006	10,218
	24,000	0.0385±0.0015	20,440
LINK-Shared, $L = 8$	6,000	0.0420±0.0007	1,077
	12,000	0.0407±0.0012	2,154
	24,000	0.0431±0.0027	4,308
LINK-Shared, $L = 24$	48,000	0.0363±0.0015	4,931
	96,000	**0.0337±0.0009**	9,909
	192,000	**0.0313±0.0016**	20,014
	384,000	**0.0329±0.0009**	40,093

I MATCHED-CAPACITY AND CROSS-GEOMETRY CONTROLS

This section supports Section 5.2: a Transformer grown to the capacity of LINK, and a second task family with a varying geometry.

I.1 MATCHED-CAPACITY TRANSFORMER ON SCATTERED DARCY

The scattered-Darcy configurations of Table 3 differ sharply in size, so we ran a separate capacity-control sweep in which Transformer width is increased and its learning rate is retuned. Table 18 reports the lowest measured error on each ladder and, for a common budget, the error at 24,000 steps. The sweep fixes the weight decay at 10^{-5} for every arm, LINK included, so all comparisons within the sweep share one setting; the runs of Table 3 select weight decay on validation data. Each ladder point is a single seed, and these values belong to the capacity-control protocol.

Table 18: **Capacity control on scattered Darcy.** Lowest measured test relative L^2 on each capacity ladder, with its budget, and test relative L^2 at the common budget of 24,000 steps, single seed per ladder point. The matched pair is the 39.39M Transformer and 40.60M LINK configuration. The top three results are marked by **First**, **Second**, **Third**.

Method ↓	params	selected lr	lowest test rel-L^2 ↓	steps ↓	at 24k ↓
Transformer	0.45M	3×10^{-4}	0.0531	384k	0.0686
Transformer	3.56M	2×10^{-4}	**0.0483**	384k	**0.0576**
Transformer	7.10M	1×10^{-4}	0.0526	384k	0.0642
Transformer	14.20M	1×10^{-4}	**0.0454**	96k	**0.0548**
Transformer	39.39M	6×10^{-5}	0.0508	384k	0.0677
LINK (Ours)	40.60M	1×10^{-4}	**0.0334**	24k	**0.0334**

The matched pair differs by 3.1% in parameter count, while LINK has 34.3% lower test error (0.0334 versus 0.0508). Both matched models reach their lowest measured error at the largest budget of their ladders. At the common 24,000 steps, every Transformer is at 0.0548 or above, against 0.0334 for LINK.

I.2 EQUAL-STEP ELASTICITY BREADTH CHECK

For a geometry outside the Darcy experiments, each model is tuned on a validation split, retrained on the full training set, and evaluated on the untouched test set for a common 30,000 optimizer steps. Table 19 reports the four-seed means with sample standard deviations, matching the uncertainty convention used elsewhere in the paper.

Table 19: **Elasticity at a common 30k-step budget.** Test relative L^2 over four seeds; lower is better. The top three methods are marked by **First**, **Second**, **Third**.

Method ↓	test rel-L^2 ↓
GNOT	**0.0150±0.0011**
Galerkin	**0.0160±0.0003**
Transolver	0.0178±0.0016
Transformer	0.0248±0.0004
LINK (Ours)	**0.0114±0.0002**

At the common optimizer-step budget, LINK is the lowest-error method in this comparison, with 24% lower mean error than the closest baseline, GNOT.

J GREEN-FUNCTION ALIGNMENT: ADDITIONAL MEASUREMENTS

The main text reports the Poisson test error, Jacobian–Green correlation, locality-prior control, sample dependence, and the κ sweep. This section gives additional routing measurements behind those claims.

The $\kappa = 8$, $n = 400$ reference point was repeated with a second seed. Relative L^2 is 0.0044 versus 0.0045 for the first seed, and Jacobian–Green correlation is 0.986 versus 0.983. The corresponding layerwise routing correlations are 0.58/0.81/0.80/0.33 versus 0.50/0.82/0.77/0.30, and Table 21 lists those of the first seed. The middle-layer pattern and the physical decay scale are therefore stable across the two measured seeds.

Table 20: **Sample dependence of the learned interaction.** Correlation with the Green operator from the same sample versus a mismatched sample, for one trained model (seed 0) and 12 test samples as probes; the coefficient field ranges over $[1, 8000]$. The table reports the two early routing layers, where the sample dependence is concentrated. The intrinsic mean correlation between different samples' Green operators is 0.573. Test rel-L^2 on this task: LINK 0.0073, Transformer 0.0341.

Quantity ↓ / metric →	matched ↑	mismatched ↑	gap ↑	matched higher / p
Transformer Jacobian	0.757	0.345	0.412	12/12, $< .001$
LINK (Ours) Jacobian	**0.966**	0.527	**0.440**	12/12, $< .001$
LINK (Ours) routing, trained, layer 1	0.677	0.475	0.202	12/12, $< .001$
LINK (Ours) routing, trained, layer 2	0.662	0.600	0.062	11/12, .006
LINK (Ours) routing, untrained, layer 1	0.147	0.116	0.031	12/12, $< .001$
LINK (Ours) routing, untrained, layer 2	0.160	0.127	0.033	12/12, $< .001$

Table 21: **Layerwise routing at the Poisson reference point.** Correlation between LINK routing and the exact Green operator for $\kappa = 8$, $n = 400$, seed 0. The untrained row isolates the spatial structure present at initialization.

State ↓ / layer →	1	2	3	4
trained	0.50	**0.82**	0.77	0.30
untrained	0.38	0.38	0.37	0.37

For the same seed, the correlation length of the exact Green function is 0.123 (analytic $1/\kappa = 0.125$). LINK's second and third routing layers have correlation lengths 0.199 and 0.178 after training, whereas the untrained routing stays above the $1/e$ decay threshold over the measured domain.

Table 22: **Correlation with the Green operator across interaction ranges.** Poisson, $n = 400$, $E = 4000$. The upper block is seed 0; the lower block repeats the main Jacobian trend at seed 1. Increasing κ makes the physical interaction more local.

Metric ↓ / κ →	1	4	8	16	32
Transformer corr(J, G)↑	0.819	0.853	0.871	0.921	0.966
Exact G length	0.188	0.148	0.123	0.100	0.077
LINK (Ours) rel-L^2 ↓	0.0071	0.0059	0.0045	0.0039	**0.0028**
corr(J, G)↑	0.963	0.971	0.983	0.990	**0.996**
Difference, seed 0	**0.144**	0.118	0.112	0.069	0.030
Difference, mean of two seeds	**0.168**	0.145	0.115	0.078	0.031
Peak corr(routing,G)↑	0.788	0.816	**0.817**	0.808	0.776
Untrained routing, same layer	0.476	0.455	0.375	0.246	0.145
Improvement from training, same layer	0.312	0.361	0.442	0.562	**0.631**
Learned length, layer 2	0.374	0.280	0.199	0.134	0.107
corr(J, G)↑, seed 1	0.970	0.977	0.986	0.995	**0.996**
Difference, seed 1	**0.191**	0.173	0.119	0.086	0.032

Table 23: **Routing–Green correlation under refinement.** Poisson at $\kappa = 8$, refinement protocol of Appendix F: both models are retrained at every n, single seed. Routing–Green correlation stays near 0.82 across the measured sizes, so the learned routing–Green alignment is similar across independently trained resolutions, and the LINK Jacobian stays closer to G than the Transformer's at every n.

Metric ↓ / n →	400	4,096	16,384
Transformer corr(J, G)↑	0.872	0.482	0.292
LINK (Ours) rel-L^2 ↓	0.0045	0.0051	0.0077
Peak corr(routing,G)↑	0.810	**0.839**	0.817
corr(J, G)↑	**0.983**	0.692	0.385

Table 24: **Removing the Gaussian locality prior.** Poisson, $\kappa = 8$, $n = 400$, $E = 4000$. Test error and Jacobian–Green correlation remain similar even though the routing starts with almost no routing–Green correlation when the prior is removed. Per seed; the peak layer can differ across seeds. Without the prior, the peak-layer correlation averages 0.717 over three seeds and the same-layer improvement from training averages 0.716.

	default		no locality prior		
Metric ↓ / seed →	0	1	0	1	2
rel-L^2 ↓	0.0045	0.0044	0.0046	0.0051	0.0053
corr(J, G)↑	0.983	0.986	0.971	0.982	0.970
Peak trained corr(routing,G)↑	0.817	0.815	0.496	**0.889**	0.767
Peak untrained corr(routing,G)	0.376	0.395	0.038	0.046	0.049
Improvement from training↑	0.442	0.420	0.458	**0.898**	0.791
Untrained far/near ratio	0.58–0.59	0.57–0.59	0.97–1.01	0.99–1.04	0.99–1.01
Learned correlation length, peak layer	0.199	0.203	0.235	0.164	0.228

K COST, MEMORY, AND SCALE

The main text reports the measured cost of LINK-Dense and LINK-Landmark at two point-set sizes. This section reports the large-Darcy runs, verifies that query blocking leaves the dense computation unchanged, and gives the depth-sharing measurements behind Table 4.

Large point sets. LINK-Landmark keeps the same pair network but replaces all-pairs routing by two stages through m learned landmarks. The large-Darcy measurement is reported in Table 7: with $m = 32$, test error remains between 0.116 and 0.119 from $2{,}048$ to $65{,}536$ points.

Exact query blocking. Dense routing can be evaluated without materializing the complete pair-activation tensor. We block the query dimension while retaining every key in each softmax row, so the normalization and output are unchanged. On a representative four-layer configuration with batch size 8, $n = 972$, $d = 96$, and $d_\phi = 128$, blocking with query blocks of 128 reduces peak pair activations from 67.6 GB to 2.2 GB. Forward outputs are bit-identical to direct dense evaluation on feasible sizes, and gradients agree to 2×10^{-8} relative to the global gradient scale, including ragged final blocks. Dense memory is therefore set by the block size, and the full pair-activation tensor is not stored.

Sharing the pair-network output across depth. LINK-Dense evaluates an independent pair network at every layer. LINK-Shared materializes the learned pair-network contribution once and reuses it across depth, while each layer retains its own learned bandwidth and value map. At $L = 4$ and $d_\phi = 64$, measured step time changes from 125.6 to 41.4 ms at $n = 1024$ and from 1664.5 to 628.6 ms at $n = 4096$. The Poisson reference measurements remain unchanged at the reported precision at the tested settings (0.0045 relative L^2 and 0.983 Jacobian–Green correlation at $\kappa = 8$).

The scattered-Darcy depth study is summarized in Table 25. The shallow shared model gives a lower error than the Transformer in less training time, while deeper shared models continue to reduce error

at larger budgets. The Shared rows of this table form one depth-sharing study; the Dense row is the 12,000-step point of the per-layer ladder in Table 17.

Table 25: **Depth sharing on scattered Darcy.** Mean±std over three seeds, full training set. The Transformer row is its lowest measured error, still improving at 384,000 steps. The top three results are marked by **First**, **Second**, **Third**.

Method ↓ / metric →	steps ↓	test rel-L^2 ↓	s/run ↓
Transformer	384,000	0.0496±0.0004	**2,550**
LINK-Shared (Ours), $L = 8$	12,000	0.0407±0.0012	**2,154**
LINK-Shared (Ours), $L = 24$	48,000	**0.0363±0.0015**	**4,931**
LINK-Shared (Ours), $L = 24$	192,000	**0.0313±0.0016**	20,014
LINK-Dense (Ours), $L = 8$	12,000	**0.0365±0.0006**	10,218

On grid Darcy, the corresponding three-seed comparison gives 0.0174 ± 0.0003 in 3,417 s for LINK-Shared and 0.0224 ± 0.0013 in 4,744 s for the Transformer. The complete measured grid frontier, including the low-budget region where the Transformer is cheaper per run, is shown in Figure 3.

Landmark fidelity. LINK-Landmark is its own normalized routing construction, designed for scale. In the landmark-count study on Poisson at $\kappa = 20$ ($E = 2000$, three seeds), Landmark with $m = n = 400$ reaches 0.0105 ± 0.0002 against 0.0051 ± 0.0004 for Dense, which confirms that the two variants define different routing operators and places them at different accuracy–scale operating points.