PRINCIPAL-PART DECOMPOSITION FOR NEURAL OPERATOR LEARNING OF DIRICHLET-TO-NEUMANN MAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Dirichlet-to-Neumann (DtN) maps encode PDE responses on the boundary by mapping Dirichlet data to Neumann data. Learning these operators across varying geometries is challenging because DtN maps are nonlocal first-order operators with singular high-frequency behavior. For smooth planar boundaries, a boundary integral formulation decomposes the DtN map into a universal Fourier multiplier and a smoother geometry-dependent remainder. We introduce Principal-Part Decomposed Neural Operators (PPDNO), a hybrid analytic-neural framework that computes the principal part exactly via FFT and learns only the remainder using a geometry-conditioned low-rank network that preserves linearity in the boundary data. We prove smoothing and separated-approximation properties of the remainder and derive error bounds for the reconstructed map. Experiments on Laplace and Helmholtz problems show that PPDNO improves accuracy and generalization over standard neural operators and residualized baselines while maintaining high inference efficiency.

1 INTRODUCTION

Dirichlet-to-Neumann (DtN) maps send Dirichlet data into Neumann data and play a central role in inverse problems, domain decomposition, and non-reflecting boundary conditions for wave propagation (Calderón, 2006; Sylvester & Uhlmann, 1987; Toselli & Widlund, 2004; Keller & Givoli, 1989). Given a geometry and a Dirichlet datum, the DtN map provides the corresponding boundary response without requiring access to the full interior solution. Classical boundary integral methods can evaluate these operators accurately, but each new geometry typically requires assembling and solving a new dense system. A learned surrogate could amortize this cost over families of geometries while preserving the boundary-only formulation. In this work, we study the problem of learning DtN maps from sampled boundary geometries and Dirichlet data.

Neural networks can solve individual PDE instances (Raissi et al., 2019; E & Yu, 2018; Sirignano & Spiliopoulos, 2018; Yang & Zhu, 2022; Wu et al., 2026), while neural operators learn maps between function spaces. Representative architectures include DeepONet (Lu et al., 2021), FNO (Li et al., 2021), graph-based models (Anandkumar et al., 2020; Lötzsch et al., 2022; Kovachki et al., 2023), and transformers (Cao, 2021; Li et al., 2023a). Recent extensions also accommodate irregular geometries (Li et al., 2023b; Wang et al., 2024; Fang et al., 2024). Most approaches approximate the target operator directly. However, DtN maps present a fundamental challenge for data-driven learning: they are nonlocal and first-order operators that amplify small-scale high-frequency components, making approximation and resolution-independent generalization difficult.

The key observation of this work is that the leading singular structure of DtN maps is analytically known. For smooth planar boundaries, a boundary integral representation yields a decomposition

$$\breve{\Lambda}_{\boldsymbol{X}} f = 2\mathcal{W}_0 f + \mathcal{R}_{\boldsymbol{X}} f. \tag{1}$$

where the universal principal part $\mathcal{W}_0$ is a Fourier multiplier $|k|/2$ and $\mathcal{R}_{\boldsymbol{X}}$ is a smoother remainder depending on the geometry. Rather than asking a neural network to approximate the full singular operator, we compute the principal part exactly and learn only the remainder. This converts the original learning problem into the approximation of a compact, geometry-dependent operator.

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.

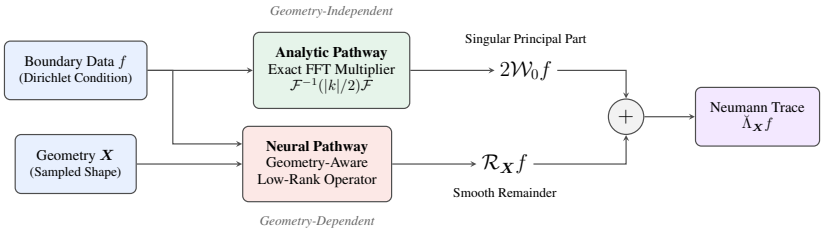


Figure 1: Overview of PPDNO: computing the principal part and learning the smooth remainder.

Based on this observation, we introduce **Principal-Part Decomposed Neural Operators (PPDNO)**. PPDNO combines an analytic FFT pathway for the universal principal part with a geometry-conditioned low-rank neural operator for the remainder. The low-rank structure is not imposed only for efficiency; it reflects the separated representation of the smooth remainder kernel and preserves the intrinsic linear dependence of the DtN map on the boundary data.

Our contributions are threefold. First, we turn the principal-part decomposition into a hybrid operator architecture that combines an exact FFT pathway with a geometry-conditioned low-rank remainder. Second, we establish smoothing and separated-approximation properties of the remainder and derive finite-node and training-error bounds for the reconstructed map. Third, we evaluate interior Laplace and exterior Helmholtz problems against direct neural operators and ablations. Residualized DeepONet and FNO controls further test whether principal-part subtraction alone explains the performance gain, while held-out and analytic out-of-distribution cases assess generalization.

2 Related Work

Neural Operators for PDEs. Neural operators include DeepONet (Lu et al., 2021; Jin et al., 2022; Cai et al., 2021), graph, low-rank, and spectral models (Anandkumar et al., 2020; Kovachki et al., 2023; Li et al., 2021; Tran et al., 2023; Rahman et al., 2023), and attention-based architectures (Cao, 2021; Li et al., 2023a). SVD-NO similarly uses a low-rank kernel factorization (Koren et al., 2026), whereas PPDNO applies it only to the analytically isolated smooth remainder.

Operator Learning on Complex Geometries. Neural operators for varying domains (Hao et al., 2023; Serrano et al., 2023; Yin et al., 2024) commonly use latent-grid coordinate maps (Li et al., 2023b) or graph and point-cloud encoders (Lötzsch et al., 2022; Wang et al., 2024; Brandstetter et al., 2022). Whereas these frameworks typically predict volumetric fields, PPDNO learns a boundary response map whose geometry affects only the smooth remainder.

Physics-Informed and Boundary Integral Neural Networks. Physics-informed operator learning incorporates governing laws into the model or objective (Wang et al., 2021; Gin et al., 2021; Li et al., 2024). More closely related methods reformulate boundary-value problems through boundary integral equations (Fang et al., 2024; Gu et al., 2026; Lin et al., 2023b;a; Melchers et al., 2026). Building on classical analysis of logarithmic and hypersingular DtN kernels (Kress, 1989; McLean, 2000; Hsiao & Wendland, 2022; Sauter & Schwab, 2010), PPDNO uses the boundary integral structure as an architectural decomposition rather than only as a solver or regularizer.

3 Background

We introduce the notation for the interior Laplace equation; extensions to other PDEs are discussed in Section 4.4.

3.1 Dirichlet-to-Neumann Maps

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain with smooth boundary $\Gamma = \partial\Omega$, and let $\boldsymbol{n}$ denote the outward unit normal. Given Dirichlet data $g \in H^{1/2}(\Gamma)$, consider

$$\begin{cases} \Delta u = 0, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega. \end{cases} \tag{2}$$

The corresponding Dirichlet-to-Neumann (DtN) map $\Lambda : H^{1/2}(\Gamma) \to H^{-1/2}(\Gamma)$ is defined by

$$\Lambda[g] = \partial_{\boldsymbol{n}} u. \tag{3}$$

3.2 Boundary Integral Equations

The DtN map can also be expressed through boundary integral operators. Let $G(\boldsymbol{x})$ be the two-dimensional Green's function, whose explicit form is given in Appendix A.1. Using an indirect double-layer representation with density φ,

$$u(\boldsymbol{x}) = \int_\Gamma \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial \boldsymbol{n}_{\boldsymbol{y}}} \varphi(\boldsymbol{y})\, ds_{\boldsymbol{y}}, \qquad \boldsymbol{x} \in \Omega. \tag{4}$$

The jump relation gives the boundary integral equation (BIE)

$$\frac{1}{2}\varphi(\boldsymbol{x}) + \int_\Gamma \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial \boldsymbol{n}_{\boldsymbol{y}}} \varphi(\boldsymbol{y})\, ds_{\boldsymbol{y}} = g(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Gamma. \tag{5}$$

The Neumann trace is given by the hypersingular integral operator

$$\partial_{\boldsymbol{n}} u(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{n}_{\boldsymbol{x}}} \int_\Gamma \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial \boldsymbol{n}_{\boldsymbol{y}}} \varphi(\boldsymbol{y})\, ds_{\boldsymbol{y}} = \frac{\partial}{\partial \boldsymbol{\tau}_{\boldsymbol{x}}} \int_\Gamma G(\boldsymbol{x}-\boldsymbol{y}) \frac{\partial \varphi}{\partial \boldsymbol{\tau}_{\boldsymbol{y}}}(\boldsymbol{y})\, ds_{\boldsymbol{y}}, \qquad \boldsymbol{x} \in \Gamma. \tag{6}$$

The second identity is the standard regularized form of the hypersingular boundary integral operator Kress (1989); Sauter & Schwab (2010). Equation 5 and equation 6 provide the boundary-integral representation of the DtN map used below.

4 Proposed Method

PPDNO is built around a simple principle: isolate the universal singular part of the DtN map analytically, and learn only the smoother component that depends on the geometry. We first derive this decomposition and then use the resulting remainder structure to construct the neural architecture.

4.1 Decomposition of the DtN Maps

Let the boundary be parameterized by a smooth closed curve $\boldsymbol{X} : \mathbb{T} \to \mathbb{R}^2$, where $\mathbb{T} = \mathbb{R}/(2\pi\mathbb{Z})$. We write $\breve{g} = g \circ \boldsymbol{X}$, $\breve{\varphi} = \varphi \circ \boldsymbol{X}$ and $\breve{\psi} = (\partial_{\boldsymbol{n}} u \circ \boldsymbol{X})|\boldsymbol{X}'|$ as the parameterized Dirichlet data, density, and Neumann trace. The BIE equation 5 gives

$$(\mathcal{I} + 2\mathcal{K})\breve{\varphi} = 2\breve{g}, \qquad \breve{\psi} = \mathcal{W}\breve{\varphi} := \breve{\Lambda}_{\boldsymbol{X}} \breve{g}, \tag{7}$$

where $\mathcal{I}$ denotes the identity and $\mathcal{K}$ is the parameterized double-layer operator. A direct calculation, given in Appendix B, yields

$$\mathcal{W} f(s) = \frac{1}{2\pi} \partial_s \int_{\mathbb{T}} \log|\boldsymbol{X}(s) - \boldsymbol{X}(t)|\, \partial_t f(t)\, dt. \tag{8}$$

The logarithmic kernel admits a canonical splitting into a universal singular part and a smoother geometry-dependent remainder,

$$\mathcal{W} = \mathcal{W}_0 + \mathcal{W}_1, \qquad \mathcal{W}_0 f(s) = \frac{1}{2\pi} \partial_s \int_{\mathbb{T}} \log\left|2\sin\frac{s-t}{2}\right| \partial_t f(t)\, dt. \tag{9}$$

The operator $\mathcal{W}_0$ is independent of the geometry and is diagonal in Fourier space:

$$\mathcal{W}_0 f = \mathcal{F}^{-1} \widehat{W}_0(k) \mathcal{F} f, \qquad \widehat{W}_0(k) = \frac{|k|}{2}, \quad k \in \mathbb{Z}. \tag{10}$$

Using equation 7, we obtain

$$\breve{\psi} = \mathcal{W}\breve{\varphi} = 2\mathcal{W}(\breve{g} - \mathcal{K}\breve{\varphi}) = 2\mathcal{W}_0\breve{g} + 2\mathcal{W}_1\breve{g} - 4\mathcal{W}\mathcal{K}(\mathcal{I} + 2\mathcal{K})^{-1}\breve{g}. \tag{11}$$

For smooth boundaries, both $\mathcal{K}$ and $\mathcal{W}_1$ have smooth kernels, so the last two terms define a smoothing remainder. Hence the DTN map can be written as

$$\breve{\Lambda}_{\boldsymbol{X}} f = 2\mathcal{W}_0 f + \mathcal{R}_{\boldsymbol{X}} f. \tag{12}$$

The singular part $\mathcal{W}_0$ is geometry-independent and is computed exactly via FFT, while the neural network only learns the smooth remainder $\mathcal{R}_{\boldsymbol{X}}$ (Theorem 5.1; proof in Appendix C).

4.2 Neural Approximation of the Remainder

Since $\mathcal{R}_{\boldsymbol{X}}$ has a smooth kernel, we approximate it by a separated low-rank representation,

$$\mathcal{R}_{\boldsymbol{X}} f(s) = \int_{\mathbb{T}} R(s,t;\boldsymbol{X}) f(t)\, dt, \qquad R(s,t;\boldsymbol{X}) \approx \sum_{k=1}^{r} a_k(s) b_k(t;\boldsymbol{X}). \tag{13}$$

Let $\{t_j\}_{j=1}^{N}$ be sensor points, $\boldsymbol{f} = (f(t_1), \ldots, f(t_N))^\top \in \mathbb{R}^N$, and $z_{\boldsymbol{X}} = (\boldsymbol{X}(t_1); \ldots; \boldsymbol{X}(t_N)) \in \mathbb{R}^{N\times 2}$. Applying quadrature to equation 13 gives

$$\mathcal{R}_{\boldsymbol{X}} f(s) \approx \sum_{k=1}^{r} a_k(s) \sum_{j=1}^{N} \omega_j b_k(t_j; z_{\boldsymbol{X}}) f(t_j). \tag{14}$$

The factor depending on the integration variable is discretized as a geometry-dependent vector

$$\boldsymbol{\beta}_k(z_{\boldsymbol{X}}) := \big(\omega_1 b_k(t_1; z_{\boldsymbol{X}}), \ldots, \omega_N b_k(t_N; z_{\boldsymbol{X}})\big)^\top \in \mathbb{R}^N, \tag{15}$$

so that its action on the Dirichlet data is $\boldsymbol{f}^\top \boldsymbol{\beta}_k(z_{\boldsymbol{X}})$. Then

$$\mathcal{R}_{\boldsymbol{X}} f(s) \approx \sum_{k=1}^{r} a_k(s)\, \boldsymbol{f}^\top \boldsymbol{\beta}_k(z_{\boldsymbol{X}}). \tag{16}$$

The network is chosen to mirror equation 16. An evaluation-point network $\mathrm{MLP}_s : \mathbb{R} \to \mathbb{R}^r$ outputs $a(s) = (a_1(s), \ldots, a_r(s))^\top$, while a geometry network $\mathrm{MLP}_{\boldsymbol{X}} : \mathbb{R}^{2N} \to \mathbb{R}^{rN}$ maps the sampled boundary to a matrix in $\mathbb{R}^{r\times N}$ whose kth row is $\boldsymbol{\beta}_k(z_{\boldsymbol{X}})^\top$. The remainder predictor is therefore

$$\mathcal{N}_r(z_{\boldsymbol{X}}, \boldsymbol{f}, s; \theta) = \big(\mathrm{MLP}_s(s)\big)^\top \big(\mathrm{MLP}_{\boldsymbol{X}}(z_{\boldsymbol{X}})\boldsymbol{f}\big), \tag{17}$$

where θ denotes the trainable parameters. The architecture explicitly encodes the linearity and low-rank separable structure of the smooth remainder, yielding an efficient approximation of the boundary integral operator. In practice, batches of geometries, boundary data, and evaluation points are processed simultaneously by vectorized matrix operations, which is equivalent to repeated application of equation 17. Figure 2 illustrates the architecture.

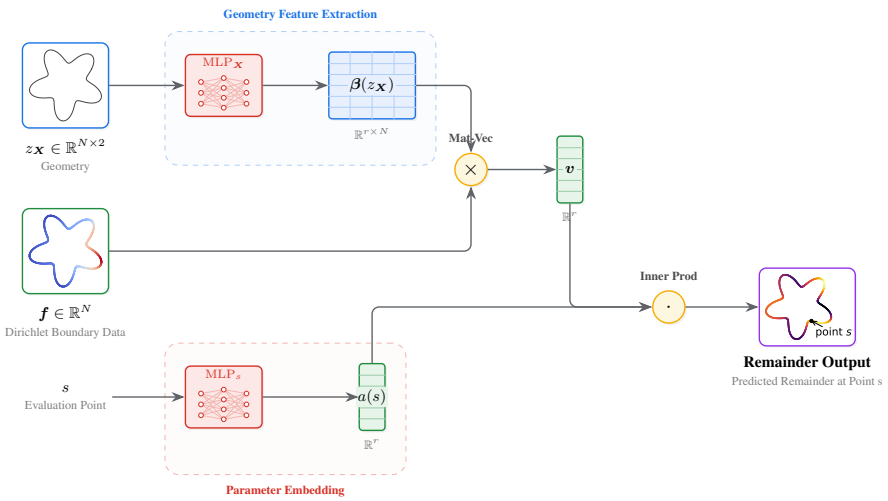


Figure 2: Schematic illustration of the proposed network architecture equation 17.

4.3 Training Strategy

Given reference DtN data $\{\boldsymbol{X}^l, f^l, \widetilde{\Lambda}_{\boldsymbol{X}^l} f^l\}$, we define the residual

$$y^l_{rem} := \widetilde{\Lambda}_{\boldsymbol{X}^l} f^l - 2\mathcal{W}_0 f^l. \tag{18}$$

The network is trained by minimizing the mean squared residual error over evaluation points $\{s_\ell\}_{\ell=1}^{N_s}$

$$\mathcal{L}_{rem}(\theta) := \frac{1}{N_{batch}} \sum_{l=1}^{N_{batch}} \frac{1}{N_s} \sum_{\ell=1}^{N_s} \left| \mathcal{N}_r(z_{\boldsymbol{X}^l}, f^l, s_\ell; \theta) - y^l_{rem}(s_\ell) \right|^2. \tag{19}$$

At inference time, the full DtN prediction is recovered by adding back the exact principal part:

$$\widehat{\Lambda}_{\boldsymbol{X}} f(s) := 2\mathcal{W}_0 f(s) + \mathcal{N}_r(z_{\boldsymbol{X}}, f, s; \theta). \tag{20}$$

4.4 Extension to Other PDEs

The same construction applies to two-dimensional PDEs whose Green's functions share the leading logarithmic singularity of the Laplace kernel. For the interior Helmholtz equation $(\Delta + k^2)u = 0$, this gives

$$\breve{\Lambda}_{\boldsymbol{X},k} f = 2\mathcal{W}_0 f + \mathcal{R}_{\boldsymbol{X},k} f, \tag{21}$$

where the remainder contains the geometry- and wavenumber-dependent contribution. For the exterior Helmholtz problem, the orientation of the normal derivative reverses the sign of the principal part:

$$\breve{\Lambda}^{\text{ext}}_{\boldsymbol{X},k} f = -2\mathcal{W}_0 f + \mathcal{R}^{\text{ext}}_{\boldsymbol{X},k} f. \tag{22}$$

The corresponding derivations are given in Appendix D.

5 Theoretical Analysis

We next justify the two structural properties built into PPDNO: after removing the principal part, the geometry-dependent remainder is smooth, and this smooth remainder admits an efficient separated low-rank approximation. We then propagate these approximation properties through boundary discretization and training to obtain an error bound for the reconstructed DtN map. Detailed assumptions and proofs are deferred to Appendix C.

Theorem 5.1 (Mapping property of the remainder operator)**.** *Let Γ be a C^∞ closed curve for which the interior Dirichlet problem is uniquely solvable. With the density convention in Section 4.1,*

$$\mathcal{R}_{\boldsymbol{X}} := \breve{\Lambda}_{\boldsymbol{X}} - 2\mathcal{W}_0 \tag{23}$$

is a smoothing operator on $\mathbb{T}$: for all Sobolev indices s and all integers $N \geq 0$,

$$\mathcal{R}_{\boldsymbol{X}} : H^s(\mathbb{T}) \to H^{s+N}(\mathbb{T}) \tag{24}$$

is bounded. For finite C^n curves, the same argument gives a finite-smoothness kernel whose regularity is limited by n.

Theorem 5.1 shows that principal-part subtraction changes the approximation problem qualitatively. Rather than learning the order-one operator $\breve{\Lambda}_{\boldsymbol{X}}$ directly, the neural component only needs to approximate the compact, geometry-dependent operator $\mathcal{R}_{\boldsymbol{X}}$, whose kernel has no diagonal singularity.

Theorem 5.2 (Low-rank approximation of the remainder operator)**.** *Assume the admissible remainder family of Definition C.3. For each rank budget $r \geq m$, one can choose trunk functions $a_1, \ldots, a_r$ independently of $\boldsymbol{X}$ such that, for every $\boldsymbol{X} \in \mathcal{A}_n$, there are geometry-dependent functions $b_k(\cdot; \boldsymbol{X})$ defining*

$$\mathcal{R}_{\boldsymbol{X},r} f(s) = \sum_{k=1}^{r} a_k(s) c_k^{\boldsymbol{X}}(f), \qquad c_k^{\boldsymbol{X}}(f) = \int_{\mathbb{T}} b_k(t; \boldsymbol{X}) f(t)\, dt, \tag{25}$$

such that

$$\|\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X},r}\|_{L^2(\mathbb{T}) \to L^\infty(\mathbb{T})} + \|\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X},r}\|_{L^2(\mathbb{T}) \to L^2(\mathbb{T})} \leq C r^{-m} \tag{26}$$

uniformly over the admissible geometry class.

This is the structural statement matched by the architecture in equation 17. The remainder kernel has the fixed-trunk form

$$R(s, t; \boldsymbol{X}) \approx \sum_{k=1}^{r} a_k(s) b_k(t; \boldsymbol{X}). \tag{27}$$

where $a_k(s)$ is chosen once and the geometry dependence is carried by $b_k(t; \boldsymbol{X})$, equivalently by the branch functionals $c_k^{\boldsymbol{X}}$, rather than through an unconstrained two-variable kernel for each boundary.

Theorem 5.3 (Discretized remainder learning and full-map error)**.** *Let $\Theta_{r,J}$ be a class of low-rank DeepONets and let μ be the distribution of boundary data and geometries. For a remainder predictor $\mathcal{N}_\theta$, set*

$$\mathcal{L}_{\mathrm{pop}}(\theta) := \mathbb{E}_{(f,\boldsymbol{X})\sim\mu} \|\mathcal{N}_\theta(f,\boldsymbol{X}) - \mathcal{R}_{\boldsymbol{X}} f\|^2_{L^2(\mathbb{T})}\,,\ \mathcal{L}_{\mathrm{emp}}(\theta) := \frac{1}{N}\sum_{i=1}^{N} \|\mathcal{N}_\theta(f_i,\boldsymbol{X}_i) - \mathcal{R}_{\boldsymbol{X}_i} f_i\|^2_{L^2(\mathbb{T})}\,. \tag{28}$$

Define

$$\Delta_N := \sup_{\theta\in\Theta_{r,J}} |\mathcal{L}_{\mathrm{pop}}(\theta) - \mathcal{L}_{\mathrm{emp}}(\theta)|\,. \tag{29}$$

Suppose the trained parameter $\widehat{\theta}$ satisfies $\mathcal{L}_{\mathrm{emp}}(\widehat{\theta}) \le \inf_{\theta\in\Theta_{r,J}} \mathcal{L}_{\mathrm{emp}}(\theta) + \eta_{\mathrm{opt}}$, and suppose there is a deterministic witness in $\Theta_{r,J}$ with remainder error $E_{\mathrm{det}}(r,J)$ on $\operatorname{supp}\mu$. Then

$$\mathcal{L}_{\mathrm{pop}}(\widehat{\theta})^{1/2} \le E_{\mathrm{det}}(r,J) + \sqrt{2\Delta_N} + \sqrt{\eta_{\mathrm{opt}}}. \tag{30}$$

If the predictor evaluates $2\mathcal{W}_0$ exactly, the same bound holds for the reconstructed DtN map. If $2\mathcal{W}_0$ is truncated to $2\mathcal{W}_0^{(K)}$, one adds the corresponding root-mean-square Fourier truncation error.

Under the finite-node assumptions in Appendix C, the deterministic remainder term separates into the expected numerical and approximation errors:

$$E_{\mathrm{det}}(r,J) \lesssim E_{\mathrm{rank}}(r) + E_{\mathrm{geom}}(J) + E_{\mathrm{quad}}(r,J) + E_{\mathrm{net}}(r,J), \tag{31}$$

where $E_{\mathrm{rank}}(r) = O(r^{-m})$ is the fixed-trunk rank error, $E_{\mathrm{geom}}(J)$ is the geometry reconstruction error, $E_{\mathrm{quad}}(r,J)$ is the input quadrature error, and $E_{\mathrm{net}}(r,J)$ is the neural realization error for the trunk and geometry-conditioned branch factors.

6 Numerical Experiments

Protocol. We evaluate PPDNO on two interior Laplace problems and two exterior Helmholtz problems, with each boundary sampled at $N = 256$ equispaced points. For every task, we consider three fixed test regimes: *ID-fixed*, in which geometries and boundary data are drawn from the training distributions; *Functional Dirichlet-OOD*, which keeps the geometry distribution fixed but changes the family of boundary data; and *Joint-OOD*, which shifts both the boundary data and geometry. Here, *ID* means "in-distribution": test geometries are independently drawn from the training geometry distribution and need not appear in the training set. Each profile contains $10{,}000$ independently generated geometries with one Dirichlet datum per geometry, and all methods are evaluated on the same samples. Stored data are split 80%/20% into fixed training and validation sets. We report the mean relative L^2 error $|\widehat{\partial_n u} - \partial_n u|_2/|\partial_n u|_2$ of the physical Neumann trace; sample-wise dispersion and relative maximum errors are reported in Appendix G.3. Training functions are random combinations of fundamental solutions with randomly sampled sources. Complete geometry, source, coefficient, and test-function distributions are given in Appendices G.1 and G.2. Independent retraining on the same datasets reproduces the main trends below (Appendix H).

Baselines and ablations. DeepONet (Lu et al., 2021), 1D FNO (Li et al., 2021), and a standard encoder–decoder operator Transformer (OT) adapted from (Li et al., 2023a) learn the full DtN map directly. Their PPD variants use the same trainable backbones and parameter counts, but learn the remainder after subtracting the principal part and add it back at inference time. PPDNO-Direct uses the same low-rank architecture as PPDNO while learning the full DtN map; FFT-only retains only the exact principal part. These comparisons separate the effect of principal-part subtraction from that of the low-rank remainder architecture. For Helmholtz, we compare FNO, PPD+FNO, FFT-only, PPDNO-Direct, and PPDNO. Architecture and training details are given in Appendix E.

For GPU-evaluated methods, throughput is measured at batch size 2,048 on one NVIDIA GeForce RTX 3080. After 10 warmup batches, we average over three synchronized passes through 10,000 device-resident samples, excluding disk I/O and host–device transfer. Timings for principal-part-decomposed methods include the neural forward pass, FFT evaluation of the principal part, and reconstruction. Figure 3 complements the mean errors in Tables 1, 2, and 3 with sample-wise error distributions for the primary learned architectures. PPDNO generally reduces both the median error and the interquartile range. *The code will be released publicly upon acceptance.*

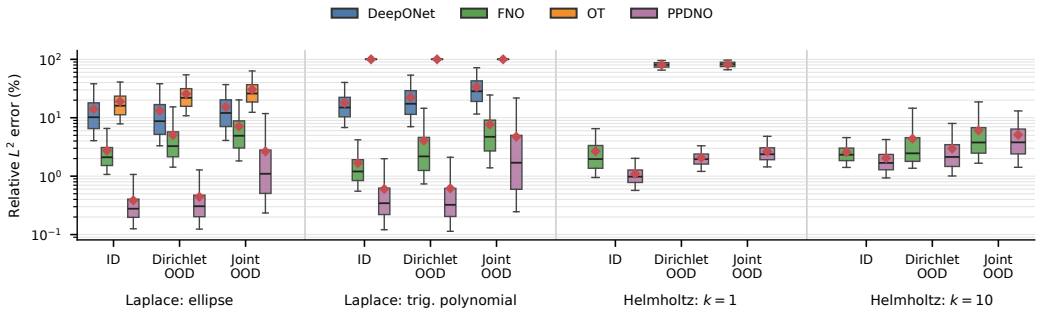


Figure 3: Sample-wise relative L^2 errors for the four tasks on a logarithmic scale. Each box summarizes 10,000 fixed samples: the box spans the interquartile range, the center line is the median, whiskers mark the 5th and 95th percentiles, and red diamonds denote means.

6.1 Laplace Equation on Elliptical Domains

Ellipses are parameterized by $\boldsymbol{X}(t) = (a\cos t, b\sin t)$ with $a, b \sim \mathcal{U}(0.5, 1.5)$. We train on 5,000 geometries with ten harmonic point-source fields per geometry. Functional Dirichlet-OOD uses an exactly balanced mixture of exterior harmonic multipoles and entire harmonic exponentials; Joint-OOD uses the same function family with $(a, b) \in [0.4, 1.6]^2 \setminus [0.5, 1.5]^2$. Details are given in Appendices G.1 and G.2.

Table 1: Mean relative L^2 error (%) and throughput on ellipses. GPU throughput uses batch 2,048, and each profile has 10,000 fixed samples. Boldface marks the best learned error. BIE is an eight-thread CPU cold-start reference and its ID-fixed throughput is not a same-hardware comparison.

Method	ID-fixed	Functional Dirichlet-OOD	Joint-OOD	Throughput (samples/s)
BIE (CPU cold-start)	0.0129	0.0120	0.0164	430
DeepONet	14.062	13.152	15.290	1,951,288
PPD+DeepONet	1.168	1.564	4.888	1,696,880
FNO	2.738	5.039	7.128	54,982
PPD+FNO	1.128	1.592	4.785	58,057
OT	18.994	25.673	30.459	2,766
PPD+OT	7.765	9.517	19.211	2,750
FFT-only	28.422	25.143	47.584	5,043,218
PPDNO-Direct	9.321	6.397	6.431	2,416,179
PPDNO	**0.385**	**0.443**	**2.662**	1,544,479

PPDNO achieves the lowest error among learned methods in all three test regimes. On Functional Dirichlet-OOD, it reduces the mean error from 6.397% for PPDNO-Direct to 0.443%, showing the benefit of principal-part decomposition within the same architecture. The paired PPD baselines also consistently outperform their direct counterparts. On Joint-OOD, PPDNO attains 2.662% error, compared with 4.785% for the strongest paired baseline. At batch size 2,048, PPDNO is $26.6\times$ faster than PPD+FNO. The BIE reference is more accurate, but requires geometry-dependent assembly and a dense solve for every query.

6.2 Laplace Equation on Trigonometric-Polynomial Boundaries

The second Laplace family is the analytic boundary

$$\boldsymbol{Z}(t) = c + Re^{i\phi}\big(e^{it} + \beta e^{-it} + \alpha_2 e^{2it} + \alpha_3 e^{3it}\big), \qquad q = |\beta| + 2|\alpha_2| + 3|\alpha_3| < 1. \tag{32}$$

Training and ID geometries use $q \in [0.10, 0.50]$, whereas Joint-OOD uses $q \in [0.60, 0.75]$. We train on 10,000 geometries with ten point-source fields per geometry; the Functional Dirichlet-OOD family is the same as in the ellipse task. The function generator, random coefficient allocation, and other details are given in Appendices G.1 and G.2.

Table 2: Mean relative L^2 error (%) and batch-2,048 throughput on trig. polynomial boundaries.

Method	ID-fixed	Functional Dirichlet-OOD	Joint-OOD	Throughput (samples/s)
DeepONet	17.925	22.248	33.682	2,411,359
PPD+DeepONet	2.084	2.508	11.928	1,581,406
FNO	1.673	4.061	7.646	39,044
PPD+FNO	0.672	0.863	5.079	38,309
OT	100.008	100.003	100.005	2,404
PPD+OT	16.629	13.885	38.831	2,434
FFT-only	16.627	13.884	38.830	2,448,838
PPDNO-Direct	10.049	10.832	19.441	2,131,037
PPDNO	**0.603**	**0.618**	**4.746**	1,401,662

Table 2 shows that PPDNO again achieves the lowest mean error on every profile, although PPD+FNO is close on Joint-OOD (5.079% versus 4.746%). Principal-part decomposition improves all paired architectures, while the near equality of PPD+OT and FFT-only indicates that OT learned little of the remainder in this task. At batch 2,048, PPDNO is $36.6\times$ faster than PPD+FNO.

6.3 Exterior Helmholtz Equation on Periodic-RBF Boundaries

We consider wave numbers $k = 1$ and $k = 10$ on a common family of analytic closed boundaries. For each boundary, 5–8 control points are sampled from disjoint angular sectors around the unit circle, with radii $1 + au_i$, $u_i \sim \mathcal{U}(-1, 1)$. Their Cartesian coordinates are interpolated with the periodic Gaussian kernel $K(t, \tau_i) = \exp\{2.5[\cos(t-\tau_i)-1]\}$, followed by random scaling, rotation, and translation. Training and ID geometries use $a \in [0.10, 0.25]$, whereas Joint-OOD uses $a \in [0.30, 0.35]$, leaving $(0.25, 0.30)$ unused. For each wave number, we train on $5{,}000$ geometries with ten fields formed from one to five outgoing $H_0^{(1)}$ monopoles per geometry. Complete interpolation, certification, and data-generation rules are given in Appendices G.1 and G.2.

Functional Dirichlet-OOD replaces the training family with a single moving outgoing mode $H_2^{(1)}(k\rho)e^{2i\theta}$ centered at a randomly sampled interior source point. The $k = 1$ and $k = 10$ tasks share geometry, source, and coefficient draws; only the wave number used for evaluation changes. PPDNO and FNO use the same task-specific harmonic coordinate encoding within each paired comparison (Appendix E).

Table 3: Mean relative L^2 error (%) and batch-2,048 throughput for exterior Helmholtz.

k	Method	ID-fixed	Functional Dirichlet-OOD	Joint-OOD	Throughput (samples/s)
1	FNO	2.669	80.637	82.049	35,240
	PPD+FNO	2.352	24.703	26.000	37,946
	FFT-only	96.378	14.598	15.097	5,240,412
	PPDNO-Direct	6.899	45.678	49.836	1,461,050
	PPDNO	**1.100**	**2.065**	**2.646**	1,318,471
10	FNO	2.575	4.372	6.072	24,370
	PPD+FNO	2.215	3.207	**4.514**	24,032
	FFT-only	108.351	110.582	112.473	5,927,733
	PPDNO-Direct	2.057	3.766	5.798	996,577
	PPDNO	**2.031**	**2.962**	5.095	964,404

Table 3 shows that at $k = 1$, PPDNO leads on all profiles and avoids the severe extrapolation errors of the baselines. At $k = 10$, PPDNO remains close to PPDNO-Direct on ID data while improving both OOD results, although PPD+FNO performs best on Joint-OOD. At batch 2,048, PPDNO is $34.7\times$ and $40.1\times$ faster than PPD+FNO for $k = 1$ and $k = 10$, respectively. The smaller advantage at larger k is consistent with the operator structure: $\mathcal{W}_0$ extracts the k-independent logarithmic singularity, while derivatives of the oscillatory factor make the remainder increasingly important, shifting the approximation bottleneck from the singularity to the oscillatory remainder. Overall,

the gains of PPD+FNO over FNO validate principal-part extraction, while PPDNO's robustness supports the low-rank remainder architecture.

The Laplace Functional Dirichlet-OOD samples are normalized by an arc-length-weighted boundary L^2 norm, whereas the Helmholtz order-two mode has a fixed unit coefficient and no per-sample amplitude normalization. The strong separation at $k = 1$ is therefore consistent with a practical advantage of preserving exact linearity in the Dirichlet data. Figure 4 shows a representative Functional Dirichlet-OOD case at $k = 10$, including the boundary, exterior field, and predicted Neumann trace.

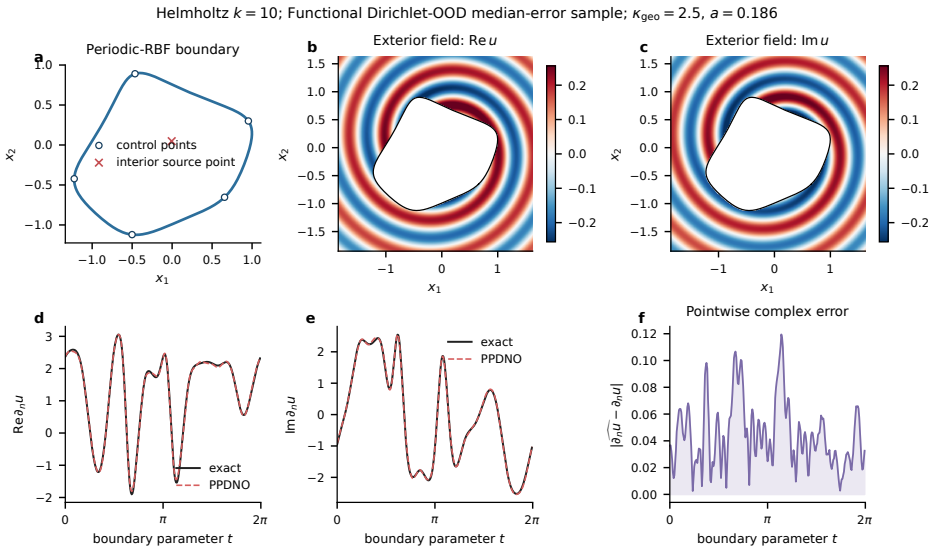


Figure 4: A representative Functional Dirichlet-OOD example for the periodic-RBF Helmholtz task at $k = 10$. Panel (a) shows the boundary, control points, and interior source point; (b,c) show the real and imaginary parts of the exterior field; (d,e) compare the exact and PPDNO Neumann traces; and (f) shows the pointwise complex error. This sample has relative L^2 error 2.13%.

6.4 Zero-Shot Output-Grid Refinement

We test generalization to finer *output evaluation grids*, while keeping the geometry and Dirichlet inputs fixed at $N_{\text{in}} = 256$. On 100 fixed ellipse Functional Dirichlet-OOD samples, the same trained model is queried at $N_{\text{out}} = 256, 512, 1024, 2048$ boundary coordinates without retraining. The analytic reference Neumann trace is reevaluated at each output grid rather than interpolated. PPDNO's mean relative L^2 error changes only from 0.4161% to 0.4164%, showing that its predictions are not tied to the 256 evaluation points used during training. Complete results are given in Appendix G.5.

7 Conclusion

PPDNO incorporates the principal-part decomposition of Dirichlet-to-Neumann maps directly into a hybrid analytic-neural operator. The universal singular component is evaluated exactly as a Fourier multiplier, while only the smooth, geometry-dependent remainder is learned through a low-rank architecture. Our analysis characterizes the regularity and approximability of this remainder and provides error bounds for the reconstructed DtN map. Across interior Laplace and exterior Helmholtz problems, PPDNO generally improves accuracy and out-of-distribution generalization while retaining high inference throughput. These results support a broader principle: when a PDE operator contains an analytically identifiable singular component, separating it from the learned component can substantially simplify the operator-learning problem. The present study focuses on smooth planar boundaries and moderate-frequency regimes; extensions to three dimensions, less regular geometries, evolving boundaries, and higher frequencies remain open.

AI USE STATEMENT

In this work, we used generative AI tools for formulating mathematical claims, assisting in the writing of proofs, providing critical ingredients for proving mathematical claims, implementing methods, interpreting results, designing or providing feedback on research methodology or experiments. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses. Assisting with translation, cleaning and reformatting dataset, supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for summarizing or analysing existing literature, editing a research paper to improve readability. We have reviewed all AI-assisted work. Specifically, all AI-assisted code was reviewed, modified as necessary, and tested for correctness by the authors, and all AI-assisted text was checked and revised by the authors for factual and technical accuracy and consistency with the reported methods and results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Architectures, hyperparameters, data-generation procedures, evaluation protocols, and independent training runs are documented in the appendices. Anonymous supplementary code provides fixed seeds and scripts for reproducing the reported experiments, tables, and figures. The code will be released publicly upon acceptance.

REFERENCES

Anima Anandkumar, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Nikola Kovachki, Zongyi Li, Burigede Liu, and Andrew Stuart. Neural operator: Graph kernel network for partial differential equations. In *ICLR 2020 workshop on integration of deep neural models and differential equations*, 2020.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vSix3HPYKSU.

Shengze Cai, Zhicheng Wang, Lu Lu, Tamer A. Zaki, and George Em Karniadakis. DeepM&Mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *Journal of Computational Physics*, 436:110296, 2021. ISSN 0021-9991. doi: 10.1016/j.jcp.2021.110296. URL http://dx.doi.org/10.1016/j.jcp.2021.110296.

Alberto P Calderón. On an inverse boundary value problem. *Computational & Applied Mathematics*, 25(2-3):133–138, 2006.

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

Weinan E and Bing Yu. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, March 2018. ISSN 2194-671X. doi: 10.1007/s40304-018-0127-z. URL https://doi.org/10.1007/s40304-018-0127-z.

Zhiwei Fang, Sifan Wang, and Paris Perdikaris. Learning only on boundaries: A physics-informed neural operator for solving parametric partial differential equations in complex geometries. *Neural Computation*, 36(3):475–498, 2024. doi: 10.1162/neco_a_01647.

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.

Yan Gu, Chuanzeng Zhang, Chaofeng Lv, Yang Yang, and Vladimir Babeshko. A novel deep learning-enhanced boundary element framework for three-dimensional elasticity. *Computer Methods in Applied Mechanics and Engineering*, 458:119058, 2026. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2026.119058. URL https://www.sciencedirect.com/science/article/pii/S0045782526003312.

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 *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

G.C. Hsiao and W.L. Wendland. *Boundary Integral Equations*. Applied Mathematical Sciences. Springer International Publishing, 2022. ISBN 9783030711290. URL https://books.google.com.hk/books?id=xIzwzgEACAAJ.

Pengzhan Jin, Shuai Meng, and Lu Lu. Mionet: Learning multiple-input operators via tensor product. *SIAM Journal on Scientific Computing*, 44(6):A3490–A3514, 2022. doi: 10.1137/22M1477751. URL https://doi.org/10.1137/22M1477751.

Joseph B Keller and Dan Givoli. Exact non-reflecting boundary conditions. *Journal of computational physics*, 82(1):172–192, 1989.

Noam Koren, Ralf JJ Mackenbach, Ruud JG van Sloun, Kira Radinsky, and Daniel Freedman. Svdno: Learning pde solution operators with svd integral kernels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 525–533, 2026.

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.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023a. ISSN 2835-8856. URL https://openreview.net/forum?id=EPPqt3uERT.

Zongyi Li, Nikola Borislavov 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*, 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, 2023b.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS J. Data Sci.*, 1(3), May 2024. doi: 10.1145/3648506. URL https://doi.org/10.1145/3648506.

Guochang Lin, Fukai Chen, Pipi Hu, Xiang Chen, Junqing Chen, Jun Wang, and Zuoqiang Shi. Bi-greennet: learning green's functions by boundary integral network. *Communications in Mathematics and Statistics*, 11(1):103–129, 2023a.

Guochang Lin, Pipi Hu, Fukai Chen, Xiang Chen, Junqing Chen, Jun Wang, and Zuoqiang Shi. Binet: Learn to solve partial differential equations with boundary integral networks. *CSIAM Transactions on Applied Mathematics*, 4(2):275–305, 2023b. ISSN 2708-0579.

Winfried Lötzsch, Simon Ohler, and Johannes Otterbach. Learning the solution operator of boundary value problems using graph neural networks. In *ICML 2022 2nd AI for Science Workshop*, 2022. URL https://openreview.net/forum?id=4vx9FQA7wiC.

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.

W.C.H. McLean. *Strongly Elliptic Systems and Boundary Integral Equations*. Cambridge University Press, 2000. ISBN 9780521663755. URL https://books.google.com.hk/books?id=RILqjEeMfK0C.

H.A. Melchers, J.H.M. Prins, and M.R.A. Abdelmalik. Neural green's operators for parametric partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 455: 118893, 2026. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2026.118893. URL https://www.sciencedirect.com/science/article/pii/S0045782526001660.

Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-NO: U-shaped neural operators. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=j3oQF9coJd.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

S.A. Sauter and C. Schwab. *Boundary Element Methods*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2010. ISBN 9783540680932. URL https://books.google.com.hk/books?id=yEFu7sVW3LEC.

Louis Serrano, Lise Le Boudec, Armand Kassaï Koupaï, Thomas X Wang, Yuan Yin, Jean-Noël Vittaut, and patrick gallinari. Operator learning with neural fields: Tackling PDEs on general geometries. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=4jEjq5nhg1.

Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

John Sylvester and Gunther Uhlmann. A global uniqueness theorem for an inverse boundary value problem. *Annals of mathematics*, pp. 153–169, 1987.

A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2004. ISBN 9783540206965. URL https://books.google.com.hk/books?id=tpSPx68R3KwC.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tmIiMPl4IPa.

Haixin Wang, Jiaxin LI, Anubhav Dwivedi, Kentaro Hara, and Tailin Wu. BENO: Boundary-embedded neural operators for elliptic PDEs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZZTkLDRmkg.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science advances*, 7(40):eabi8605, 2021.

Yulun Wu, Miguel Aguiar, Karl Henrik Johansson, and Matthieu Barreau. Iterative training of physics-informed neural networks with fourier-enhanced features. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ybffyf7LE7.

Jiang Yang and Quanhui Zhu. A local deep learning method for solving high order partial differential equations. *Numerical Mathematics: Theory, Methods and Applications*, 15(1):42–67, 2022. doi: 10.4208/nmtma.OA-2021-0035. URL https://doi.org/10.4208/nmtma.OA-2021-0035.

Minglang Yin, Nicolas Charon, Ryan Brody, Lu Lu, Natalia Trayanova, and Mauro Maggioni. A scalable framework for learning the geometry-dependent solution operators of partial differential equations. *Nature computational science*, 4(12):928–940, 2024.

A Green's Function in the Free Space

This appendix provides the definitions and explicit forms of the free-space Green's functions for the Laplace and Helmholtz equations. By definition, the Green's function $G(\boldsymbol{x})$ satisfies the governing differential equation with a Dirac delta function $\delta(\boldsymbol{x})$ as the source term.

A.1 Green's Function for the Laplace Equation

The free-space Green's function for the 2D Laplace equation satisfies:

$$\Delta G(\boldsymbol{x}) = \delta(\boldsymbol{x}) \tag{33}$$

where Δ is the Laplacian operator. The explicit forms is given by:

$$G(\boldsymbol{x}) = \frac{1}{2\pi} \ln |\boldsymbol{x}|. \tag{34}$$

A.2 Green's Function for the Helmholtz Equation

The free-space Green's function for the 2D Helmholtz equation satisfies:

$$\left(\Delta + k^2\right) G_k(\boldsymbol{x}) = \delta(\boldsymbol{x}) \tag{35}$$

where k is the wavenumber. Under the outward radiation condition at infinity, the explicit form is expressed as:

$$G_k(\boldsymbol{x}) = -\frac{i}{4} H_0^{(1)}(k|\boldsymbol{x}|), \tag{36}$$

where $H_0^{(1)}$ is the Hankel function of the first kind and zeroth order.

B Details of the DtN Principal-Part Decomposition

This appendix records the boundary-integral calculation used in Section 4.1.

To make the integral operator more explicit, we parameterize the closed curve Γ by $\boldsymbol{X} : \mathbb{T} \to \mathbb{R}^2$ with $\mathbb{T} = \mathbb{R}/(2\pi\mathbb{Z})$. The solution in the form of a double-layer potential is given by

$$u(\boldsymbol{x}) = \frac{1}{2\pi} \int_{\mathbb{T}} \frac{-(\boldsymbol{x} - \boldsymbol{X}(t)) \cdot R\boldsymbol{X}'(t)}{|\boldsymbol{x} - \boldsymbol{X}(t)|^2} \varphi(\boldsymbol{X}(t))\, dt, \tag{37}$$

where

$$R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{38}$$

The boundary integral equation is given by

$$\frac{1}{2}\varphi(\boldsymbol{X}(s)) + \frac{1}{2\pi} \int_{\mathbb{T}} \frac{-\Delta\boldsymbol{X} \cdot R\boldsymbol{X}'(t)}{|\Delta\boldsymbol{X}|^2} \varphi(\boldsymbol{X}(t))\, dt = g(\boldsymbol{X}(s)), \qquad s \in \mathbb{T}, \tag{39}$$

where $\Delta\boldsymbol{X} = \boldsymbol{X}(s) - \boldsymbol{X}(t)$. The normal derivative is given by

$$\partial_{\boldsymbol{n}} u(\boldsymbol{X}(s)) = -\frac{1}{2\pi} \frac{R\boldsymbol{X}'(s)}{|\boldsymbol{X}'(s)|} \cdot \int_{\mathbb{T}} \left(\frac{1}{|\Delta\boldsymbol{X}|^2} I - 2\frac{\Delta\boldsymbol{X} \otimes \Delta\boldsymbol{X}}{|\Delta\boldsymbol{X}|^4} \right) R\boldsymbol{X}'(t) \varphi(\boldsymbol{X}(t))\, dt, \tag{40}$$

where I denotes the 2×2 identity matrix and $\otimes$ denotes the outer product. Let $\breve{g} = g \circ \boldsymbol{X}$, $\breve{\varphi} = \varphi \circ \boldsymbol{X}$, $\breve{\psi} = (\partial_{\boldsymbol{n}} u \circ \boldsymbol{X})|\boldsymbol{X}'|$, and $\mathcal{I}$ denote the identity mapping. We have the relations

$$(\mathcal{I} + 2\mathcal{K})\breve{\varphi} = 2\breve{g}, \qquad \breve{\psi} = \mathcal{W}\breve{\varphi} := \breve{\Lambda}_{\boldsymbol{X}} \breve{g}, \tag{41}$$

where

$$\mathcal{K} f(s) = \frac{1}{2\pi} \int_{\mathbb{T}} \frac{-\Delta\boldsymbol{X} \cdot R\boldsymbol{X}'(t)}{|\Delta\boldsymbol{X}|^2} f(t)\, dt, \tag{42}$$

and

$$\mathcal{W}f(s) = -\frac{1}{2\pi} R\boldsymbol{X}'(s) \cdot \int_{\mathbb{T}} \left(\frac{1}{|\Delta \boldsymbol{X}|^2} I - 2\frac{\Delta \boldsymbol{X} \otimes \Delta \boldsymbol{X}}{|\Delta \boldsymbol{X}|^4} \right) R\boldsymbol{X}'(t) f(t)\, dt \tag{43}$$

$$= \frac{1}{2\pi} \boldsymbol{X}'(s) \cdot \int_{\mathbb{T}} \left(\frac{1}{|\Delta \boldsymbol{X}|^2} I - 2\frac{\Delta \boldsymbol{X} \otimes \Delta \boldsymbol{X}}{|\Delta \boldsymbol{X}|^4} \right) \boldsymbol{X}'(t) f(t)\, dt \tag{44}$$

$$= -\frac{1}{2\pi} \int_{\mathbb{T}} \partial_s \partial_t \log |\Delta \boldsymbol{X}| f(t)\, dt, \tag{45}$$

$$= \frac{1}{2\pi} \partial_s \int_{\mathbb{T}} \log |\Delta \boldsymbol{X}| \partial_t f(t)\, dt, \tag{46}$$

$$= \frac{1}{2\pi} \partial_s \int_{\mathbb{T}} \log \left| 2\sin \frac{s-t}{2} \right| \partial_t f(t)\, dt + \frac{1}{2\pi} \partial_s \int_{\mathbb{T}} \log \left| \frac{\Delta \boldsymbol{X}}{2\sin \frac{s-t}{2}} \right| \partial_t f(t)\, dt \tag{47}$$

$$:= \mathcal{W}_0 f(s) + \mathcal{W}_1 f(s). \tag{48}$$

The operator $\mathcal{W}_0$ is independent of $\boldsymbol{X}$ and can be characterized and efficiently computed using the Fourier transform:

$$\mathcal{W}_0 f = \mathcal{F}^{-1} \widehat{W}_0(\hat{k}) \mathcal{F} f, \qquad \widehat{W}_0(\hat{k}) = \frac{|\hat{k}|}{2}, \quad \hat{k} \in \mathbb{Z}, \tag{49}$$

where $\mathcal{F}$ is the Fourier transform on the torus $\mathbb{T}$. The operator $\mathcal{W}_1$ is a smoothing operator.

C Proofs for the Theoretical Analysis

This appendix gives the assumptions, constants, and proofs used in Section 5. We provide the boundary-integral mapping argument, the low-rank approximation of the remainder, and the error estimates for the discretized learning problem.

C.1 Mapping Property of the Remainder Operator

We first justify the decomposition $\mathcal{W} = \mathcal{W}_0 + \mathcal{W}_1$ from Section 4.1. Here $\mathcal{W}_0$ is defined in equation 10, while $\mathcal{W}_1$ is smooth. The periodic logarithmic kernel has the Fourier series

$$\log \left| 2\sin \frac{\theta}{2} \right| = -\sum_{k=1}^{\infty} \frac{\cos(k\theta)}{k} = -\sum_{k \in \mathbb{Z}\setminus\{0\}} \frac{e^{ik\theta}}{2|k|} \tag{50}$$

in the distributional sense. Applying $\mathcal{W}_0$ to e^{iks} and using equation 50 gives

$$\mathcal{W}_0 e^{iks} = \frac{1}{2\pi} \partial_s \int_{\mathbb{T}} \log \left| 2\sin \frac{s-t}{2} \right| \partial_t e^{ikt}\, dt = \frac{|k|}{2} e^{iks}, \tag{51}$$

with the zero mode mapped to zero. Thus $2\mathcal{W}_0$ is exactly the Fourier multiplier $|k|$.

Thus the only non-smooth part of the pulled-back hypersingular operator is the universal circular kernel in $\mathcal{W}_0$.

Lemma C.1 (Cancellation of the diagonal singularity)**.** *Let $\boldsymbol{X} \in C^n(\mathbb{T}; \mathbb{R}^2)$, $n \geq 2$, satisfy the chord-arc lower bound*

$$|\boldsymbol{X}(s) - \boldsymbol{X}(t)| \geq c_1 \left| 2\sin \frac{s-t}{2} \right|. \tag{52}$$

Then the double-layer kernel

$$K_{\boldsymbol{X}}(s,t) := -\frac{(\boldsymbol{X}(s) - \boldsymbol{X}(t)) \cdot R\boldsymbol{X}'(t)}{|\boldsymbol{X}(s) - \boldsymbol{X}(t)|^2} \tag{53}$$

extends continuously across $s = t$; if $n \geq 3$ it extends with C^{n-2} regularity. Moreover

$$L_{\boldsymbol{X}}(s,t) := \log \left| \frac{\boldsymbol{X}(s) - \boldsymbol{X}(t)}{2\sin \frac{s-t}{2}} \right| \tag{54}$$

extends to a C^{n-1} function on $\mathbb{T}^2$. Thus $\mathcal{K}$ has a regular kernel, and, for $n \geq 3$,

$$\mathcal{W}_1 f(s) = -\frac{1}{2\pi} \int_{\mathbb{T}} \partial_s \partial_t L_{\boldsymbol{X}}(s,t) f(t)\, dt \tag{55}$$

for smooth periodic f. Its kernel is C^{n-3} because $L_{\boldsymbol{X}} \in C^{n-1}(\mathbb{T}^2)$.

Proof. It remains to examine the diagonal $s = t$. Work in a local lift of the torus and set $h = s - t$. For $h \neq 0$,

$$\boldsymbol{X}(t+h) - \boldsymbol{X}(t) = hA(t,h), \qquad A(t,h) := \int_0^1 \boldsymbol{X}'(t+\theta h)\, d\theta. \tag{56}$$

The function A extends to $h = 0$, with $A(t,0) = \boldsymbol{X}'(t)$, and has one fewer derivative than $\boldsymbol{X}$. Similarly,

$$2\sin\frac{h}{2} = hB(h), \qquad B(h) := \frac{2\sin(h/2)}{h}, \qquad B(0) = 1, \tag{57}$$

where B is smooth and nonzero near $h = 0$. Thus

$$\frac{|\boldsymbol{X}(t+h) - \boldsymbol{X}(t)|^2}{|2\sin(h/2)|^2} = \frac{|A(t,h)|^2}{B(h)^2} \tag{58}$$

extends regularly to $h = 0$ with diagonal value $|\boldsymbol{X}'(t)|^2$. The chord-arc bound keeps this ratio uniformly away from zero. Taking the logarithm therefore gives the stated regular extension of $L_{\boldsymbol{X}}$.

For the double-layer kernel, the apparent first-order singularity cancels since $\boldsymbol{X}'(t) \cdot R\boldsymbol{X}'(t) = 0$. Indeed,

$$(\boldsymbol{X}(t+h) - \boldsymbol{X}(t)) \cdot R\boldsymbol{X}'(t) = h \int_0^1 (\boldsymbol{X}'(t+\theta h) - \boldsymbol{X}'(t)) \cdot R\boldsymbol{X}'(t)\, d\theta. \tag{59}$$

For $n \geq 2$ the difference inside the integral may be written as

$$\boldsymbol{X}'(t+\theta h) - \boldsymbol{X}'(t) = h \int_0^\theta \boldsymbol{X}''(t+\eta h)\, d\eta. \tag{60}$$

Thus

$$(\boldsymbol{X}(t+h) - \boldsymbol{X}(t)) \cdot R\boldsymbol{X}'(t) = h^2 C(t,h) \cdot R\boldsymbol{X}'(t), \qquad C(t,h) := \int_0^1 \int_0^\theta \boldsymbol{X}''(t+\eta h)\, d\eta\, d\theta. \tag{61}$$

The denominator equals $h^2|A(t,h)|^2$, and $|A(t,0)| = |\boldsymbol{X}'(t)| \geq c_0$. Hence, near the diagonal,

$$K_{\boldsymbol{X}}(t+h,t) = \frac{C(t,h) \cdot R\boldsymbol{X}'(t)}{|A(t,h)|^2} \tag{62}$$

for $h \neq 0$, and the right-hand side defines its continuous extension. Its diagonal value is

$$K_{\boldsymbol{X}}(t,t) = \frac{1}{2} \frac{\boldsymbol{X}''(t) \cdot R\boldsymbol{X}'(t)}{|\boldsymbol{X}'(t)|^2} \tag{63}$$

because $C(t,0) = \frac{1}{2}\boldsymbol{X}''(t)$. Since C is C^{n-2} and the denominator is bounded away from zero, the quotient is C^{n-2}.

Finally,

$$\mathcal{W}_1 f(s) = \frac{1}{2\pi} \partial_s \int_{\mathbb{T}} L_{\boldsymbol{X}}(s,t) \partial_t f(t)\, dt. \tag{64}$$

For smooth periodic f, integration by parts in t has no boundary term, so

$$\mathcal{W}_1 f(s) = -\frac{1}{2\pi} \int_{\mathbb{T}} \partial_s \partial_t L_{\boldsymbol{X}}(s,t) f(t)\, dt. \tag{65}$$

The stated kernel regularity follows by differentiating the C^{n-1} function $L_{\boldsymbol{X}}$ once in each variable. □

For the Laplace equation, Lemma C.1 shows that the double-layer operator $\mathcal{K}$ has a regular kernel. The composition $\mathcal{W}\mathcal{K}(\mathcal{I}+2\mathcal{K})^{-1}$ in equation 11 is also smoothing: $\mathcal{K}$ first smooths the input, after which $\mathcal{W}$ acts on a smooth boundary function. Definition C.3 records the corresponding uniform assumptions over the admissible geometry class. This gives the decomposition equation 12.

Theorem C.2 (Mapping property of the remainder operator)**.** *Let Γ be a C^∞ closed curve for which the interior Dirichlet problem is uniquely solvable. With the density convention above, the operator*

$$\mathcal{R}_{\boldsymbol{X}} := \breve{\Lambda}_{\boldsymbol{X}} - 2\mathcal{W}_0 \tag{66}$$

is smoothing on $\mathbb{T}$: for all Sobolev indices s and all integers $N \geq 0$,

$$\mathcal{R}_{\boldsymbol{X}} : H^s(\mathbb{T}) \to H^{s+N}(\mathbb{T}) \tag{67}$$

is bounded. For finite C^n curves, the same argument gives a finite-smoothness kernel whose regularity is limited by n.

Proof. The jump relations and invertibility of the second-kind double-layer equation on smooth curves are standard consequences of Fredholm boundary-integral theory (McLean, 2000; Kress, 1989). Hence $(\mathcal{I}+2\mathcal{K})^{-1}$ is bounded on the Sobolev and Hölder scales used here. By Lemma C.1, $\mathcal{K}$ has a smooth kernel when $\boldsymbol{X}$ is C^∞, and the non-circular part $\mathcal{W}_1$ of the hypersingular operator also has a smooth kernel.

We first recall a standard consequence of a smooth kernel. If

$$Af(s) = \int_{\mathbb{T}} A(s,t) f(t)\, dt \tag{68}$$

and $A \in C^\infty(\mathbb{T}^2)$, then $A : H^s(\mathbb{T}) \to H^q(\mathbb{T})$ is bounded for all s, q. To see this, write the two-variable Fourier coefficients as $A_{\ell j}$. Smoothness gives, for every M_1, M_2,

$$|A_{\ell j}| \leq C_{M_1,M_2} \langle \ell \rangle^{-M_1} \langle j \rangle^{-M_2}, \qquad \langle k \rangle := (1+k^2)^{1/2}. \tag{69}$$

The Fourier coefficients of Af are finite linear combinations of $\sum_j A_{\ell,-j} \widehat{f}_j$. Choosing $M_1 > q + \frac{1}{2}$ and $M_2 > \frac{1}{2} - s$ and applying Cauchy–Schwarz gives

$$\begin{aligned} \|Af\|_{H^q}^2 &\leq C \sum_{\ell \in \mathbb{Z}} \langle \ell \rangle^{2q-2M_1} \left(\sum_{j \in \mathbb{Z}} \langle j \rangle^{-M_2} |\widehat{f}_j| \right)^2 \\ &\leq C \sum_{\ell \in \mathbb{Z}} \langle \ell \rangle^{2q-2M_1} \sum_{j \in \mathbb{Z}} \langle j \rangle^{-2(M_2+s)} \|f\|_{H^s}^2 \leq C \|f\|_{H^s}^2 . \end{aligned} \tag{70}$$

Hence both $\mathcal{K}$ and $\mathcal{W}_1$ gain arbitrarily many Sobolev derivatives.

Using the identity displayed before the theorem,

$$\breve{\Lambda}_{\boldsymbol{X}} = 2\mathcal{W}_0 + 2\mathcal{W}_1 - 4\mathcal{W}\mathcal{K}(\mathcal{I}+2\mathcal{K})^{-1}. \tag{71}$$

The term $2\mathcal{W}_1$ is smoothing by the preceding paragraph. For the last term, fix $s \in \mathbb{R}$ and an integer $N \geq 0$. Given $f \in H^s(\mathbb{T})$, set

$$g := (\mathcal{I}+2\mathcal{K})^{-1} f. \tag{72}$$

Bounded invertibility gives $\|g\|_{H^s} \leq C \|f\|_{H^s}$. The smooth-kernel estimate for $\mathcal{K}$ with target index $s+N+1$ gives

$$\|\mathcal{K}g\|_{H^{s+N+1}} \leq C \|g\|_{H^s} \leq C \|f\|_{H^s} . \tag{73}$$

The hypersingular operator $\mathcal{W} = \mathcal{W}_0 + \mathcal{W}_1$ is bounded $H^a(\mathbb{T}) \to H^{a-1}(\mathbb{T})$: $\mathcal{W}_0$ has Fourier symbol $|k|/2$, and $\mathcal{W}_1$ is smoothing. Therefore

$$\left\| \mathcal{W}\mathcal{K}(\mathcal{I}+2\mathcal{K})^{-1} f \right\|_{H^{s+N}} \leq C \|f\|_{H^s} . \tag{74}$$

Combining this estimate with the smoothing bound for $2\mathcal{W}_1$ proves $\mathcal{R}_{\boldsymbol{X}} : H^s(\mathbb{T}) \to H^{s+N}(\mathbb{T})$. If $\boldsymbol{X}$ has only C^n regularity, the same calculation applies only up to the number of kernel derivatives supplied by Lemma C.1; this is the finite-regularity situation encoded in Definition C.3. □

Theorem 5.1 is the reason for subtracting $2\mathcal{W}_0$ before training. After the principal part is removed, the neural approximation is applied only to the compact geometry-dependent remainder, not to the order-one Fourier multiplier.

C.2 Remainder Class and Low-Rank Approximation

We formulate the approximation theory for the family of remainder operators $\{\mathcal{R}_{\boldsymbol{X}}\}_{\boldsymbol{X}\in\mathcal{A}_n}$ in equation 12. Since $2\mathcal{W}_0$ is known, the unknown object to approximate is $\mathcal{R}_{\boldsymbol{X}}$, uniformly over the geometry class.

Let $n \geq 2$ and fix positive constants M, c_0, c_1. We define the admissible class

$$\mathcal{A}_n(M, c_0, c_1) := \left\{ \begin{array}{ll} \boldsymbol{X} \in C^n(\mathbb{T}; \mathbb{R}^2) : & \|\boldsymbol{X}\|_{C^n(\mathbb{T})} \leq M, \\ & |\boldsymbol{X}'(s)| \geq c_0, \\ & |\boldsymbol{X}(s) - \boldsymbol{X}(t)| \geq c_1 \left| 2 \sin \dfrac{s-t}{2} \right| \end{array} \right\}. \tag{75}$$

The two lower bounds exclude vanishing parameterization speed and self-intersections. Thus the class consists of smooth embedded curves with uniform geometric bounds.

Definition C.3 (Admissible remainder family)**.** *Fix integers $m \geq 1$, $0 \leq \mu \leq m$, and $0 \leq \nu < n$. We assume that for every $\boldsymbol{X} \in \mathcal{A}_n(M, c_0, c_1)$ the remainder operator has the representation*

$$\mathcal{R}_{\boldsymbol{X}} f(s) = \int_{\mathbb{T}} R(s, t; \boldsymbol{X}) f(t)\, dt \tag{76}$$

with

$$\sup_{\boldsymbol{X}\in\mathcal{A}_n} \|R(\cdot, \cdot; \boldsymbol{X})\|_{C^m(\mathbb{T}^2)} \leq M_R, \tag{77}$$

and

$$\|R(\cdot, \cdot; \boldsymbol{X}) - R(\cdot, \cdot; \boldsymbol{Y})\|_{C^\mu(\mathbb{T}^2)} \leq L_R \|\boldsymbol{X} - \boldsymbol{Y}\|_{C^\nu(\mathbb{T})}. \tag{78}$$

This is the finite-regularity counterpart of Theorem 5.1. It keeps only the uniform kernel bounds needed below for approximation and discretization.

Because $R(\cdot, \cdot; \boldsymbol{X})$ is continuous on the compact torus, $\mathcal{R}_{\boldsymbol{X}}$ is Hilbert–Schmidt and therefore compact as an operator $L^2(\mathbb{T}) \to L^2(\mathbb{T})$. We repeatedly use the following estimates: if $(\mathcal{A}f)(s) = \int_{\mathbb{T}} K(s, t) f(t)\, dt$ with $K \in L^\infty(\mathbb{T}^2)$, then Hölder's inequality gives

$$\|\mathcal{A}\|_{L^2(\mathbb{T})\to L^\infty(\mathbb{T})} \leq |\mathbb{T}|^{1/2} \|K\|_{L^\infty(\mathbb{T}^2)}, \tag{79}$$

$$\|\mathcal{A}\|_{L^2(\mathbb{T})\to L^2(\mathbb{T})} \leq |\mathbb{T}| \|K\|_{L^\infty(\mathbb{T}^2)}, \tag{80}$$

$$\|\mathcal{A}f\|_{L^\infty(\mathbb{T})} \leq |\mathbb{T}| \|K\|_{L^\infty(\mathbb{T}^2)} \|f\|_{C^0(\mathbb{T})}. \tag{81}$$

The approximation class is a geometry-conditioned low-rank DeepONet with a fixed trunk

$$\mathcal{N}_{r,J}(f, \boldsymbol{X})(s) := \sum_{k=1}^{r} \tau_k(s)\, \rho_k(f; Z_J(\boldsymbol{X})), \tag{82}$$

where $Z_J(\boldsymbol{X})$ is built from J boundary nodes, τ_k are fixed trunk outputs, and ρ_k are branch outputs conditioned on the geometry. In the exact representation results, $f \mapsto \rho_k(f; Z_J(\boldsymbol{X}))$ is allowed to be a bounded linear functional for each fixed geometry code. This is the mathematical idealization of the branch component before finite-node and neural approximation errors are added.

Finite-rank kernels fit this form. If

$$R(s, t; \boldsymbol{X}) = \sum_{k=1}^{r} a_k(s) b_k(t; \boldsymbol{X}) \tag{83}$$

then

$$\mathcal{R}_{\boldsymbol{X}} f(s) = \sum_{k=1}^{r} a_k(s) c_k^{\boldsymbol{X}}(f), \qquad c_k^{\boldsymbol{X}}(f) := \int_{\mathbb{T}} b_k(t; \boldsymbol{X}) f(t)\, dt. \tag{84}$$

If $b_k(\cdot; \boldsymbol{X}) \in L^2(\mathbb{T})$, then

$$|c_k^{\boldsymbol{X}}(f)| \leq \|b_k(\cdot; \boldsymbol{X})\|_{L^2(\mathbb{T})} \|f\|_{L^2(\mathbb{T})} \tag{85}$$

by Cauchy–Schwarz, so the branch map is bounded.

C.3 Low-Rank Approximation of the Remainder Operator

The next estimate gives a constructive rate in the rank. The key point for the architecture is that the trunk functions are chosen once for a given rank budget, while the factors acting on the input may depend on the geometry.

Theorem C.4 (Low-rank approximation of the remainder operator)**.** *Assume Definition C.3. Then for every $\boldsymbol{X} \in \mathcal{A}_n$ and every integer $p \geq 2$ there exists a separated kernel*

$$R_p(s,t;\boldsymbol{X}) = \sum_{k=1}^{r_p} a_k(s) b_k(t;\boldsymbol{X}) \tag{86}$$

with rank

$$r_p \leq m(p+1) \tag{87}$$

such that

$$\|R(\cdot,\cdot;\boldsymbol{X}) - R_p(\cdot,\cdot;\boldsymbol{X})\|_{L^\infty(\mathbb{T}^2)} \leq C_{\mathrm{ker}} p^{-m}, \tag{88}$$

where C_{ker} depends only on m, $|\mathbb{T}|$, and M_R, uniformly for $\boldsymbol{X} \in \mathcal{A}_n$. The functions a_k may depend on the rank budget but are chosen independently of $\boldsymbol{X}$. Consequently, for every $r \geq m$ there exists a separated operator of rank at most r,

$$\mathcal{R}_{\boldsymbol{X},r} f(s) := \sum_{k=1}^{r} a_k(s) c_k^{\boldsymbol{X}}(f), \qquad c_k^{\boldsymbol{X}}(f) := \int_{\mathbb{T}} b_k(t;\boldsymbol{X}) f(t)\, dt, \tag{89}$$

such that

$$\|\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X},r}\|_{L^2(\mathbb{T})\to L^\infty(\mathbb{T})} \leq C_{\mathrm{rank},\infty} r^{-m}, \tag{90}$$

$$\|\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X},r}\|_{L^2(\mathbb{T})\to L^2(\mathbb{T})} \leq C_{\mathrm{rank},2} r^{-m}, \tag{91}$$

uniformly in $\boldsymbol{X} \in \mathcal{A}_n$.

Proof. Fix $p \geq 2$ and set

$$h := \frac{2\pi}{p+1}, \qquad s_j := jh, \qquad j = 0,\dots,p. \tag{92}$$

Let $\{\chi_j\}_{j=0}^p$ be the periodic piecewise-linear hat partition of unity on $\mathbb{T}$ associated with these nodes: $\chi_j \geq 0$, $\sum_j \chi_j = 1$, and $\chi_j(s)$ is supported where the signed periodic distance $d_j(s)$ from s_j to s satisfies $|d_j(s)| \leq h$. Since $p \geq 2$, these supports lie in coordinate arcs where d_j is a smooth local coordinate.

For fixed t and $\boldsymbol{X}$, Taylor expand the function $s \mapsto R(s,t;\boldsymbol{X})$ at each node s_j and glue the local polynomials with the partition of unity:

$$P_j(s,t;\boldsymbol{X}) := \sum_{\ell=0}^{m-1} \frac{\partial_s^\ell R(s_j,t;\boldsymbol{X})}{\ell!} d_j(s)^\ell, \tag{93}$$

$$R_p(s,t;\boldsymbol{X}) := \sum_{j=0}^{p} \chi_j(s) P_j(s,t;\boldsymbol{X}). \tag{94}$$

The representation is separated in (s,t):

$$R_p(s,t;\boldsymbol{X}) = \sum_{j=0}^{p} \sum_{\ell=0}^{m-1} a_{j,\ell}(s) b_{j,\ell}(t;\boldsymbol{X}), \tag{95}$$

where

$$a_{j,\ell}(s) := \chi_j(s) \frac{d_j(s)^\ell}{\ell!}, \qquad b_{j,\ell}(t;\boldsymbol{X}) := \partial_s^\ell R(s_j,t;\boldsymbol{X}). \tag{96}$$

There are $m(p+1)$ pairs (j,ℓ), so the separated rank is at most $m(p+1)$.

The kernel error follows from Taylor's theorem. If $\chi_j(s) \neq 0$, the integral form of Taylor's theorem gives

$$\begin{aligned} |R(s,t;\boldsymbol{X}) - P_j(s,t;\boldsymbol{X})| &\leq \frac{|d_j(s)|^m}{(m-1)!} \int_0^1 (1-\theta)^{m-1} \left|\partial_s^m R(s_j + \theta d_j(s), t; \boldsymbol{X})\right| \, d\theta \\ &\leq \frac{M_R}{m!} h^m. \end{aligned} \tag{97}$$

Since the χ_j form a nonnegative partition of unity,

$$|R(s,t;\boldsymbol{X}) - R_p(s,t;\boldsymbol{X})| \leq \sum_{j=0}^{p} \chi_j(s) \frac{M_R}{m!} h^m = \frac{M_R}{m!} \left(\frac{2\pi}{p+1}\right)^m. \tag{98}$$

After enlarging the constant and using $(p+1)^{-m} \leq p^{-m}$, this gives the stated kernel bound.

Let $\mathcal{R}_{\boldsymbol{X},p}$ denote the operator with kernel R_p. By equation 79–equation 80,

$$\|\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X},p}\|_{L^2(\mathbb{T}) \to L^\infty(\mathbb{T})} \leq |\mathbb{T}|^{1/2} C_{\mathrm{ker}} p^{-m}, \tag{99}$$

$$\|\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X},p}\|_{L^2(\mathbb{T}) \to L^2(\mathbb{T})} \leq |\mathbb{T}| C_{\mathrm{ker}} p^{-m}. \tag{100}$$

It remains to express the estimate in terms of an arbitrary rank budget r. First suppose $r \geq 3m$ and choose

$$p := \left\lfloor \frac{r}{m} \right\rfloor - 1. \tag{101}$$

Then $p \geq 2$ and $m(p+1) \leq r$, while $p \geq r/(3m)$. Hence $p^{-m} \leq (3m)^m r^{-m}$, giving equation 90 and equation 91. For the finitely many ranks $m \leq r < 3m$, take the zero operator and enlarge the constants using

$$\|\mathcal{R}_{\boldsymbol{X}}\|_{L^2 \to L^\infty} \leq |\mathbb{T}|^{1/2} M_R, \qquad \|\mathcal{R}_{\boldsymbol{X}}\|_{L^2 \to L^2} \leq |\mathbb{T}| M_R. \tag{102}$$

The constants remain independent of $\boldsymbol{X}$. □

C.4 Exact-Geometry Neural Approximation

Theorem C.5 (Exact-geometry approximation error)**.** *Assume Definition C.3. Fix $\boldsymbol{X} \in \mathcal{A}_n$ and $r \geq m$. Let*

$$\mathcal{R}_{\boldsymbol{X},r} f(s) = \sum_{k=1}^{r} a_k(s) c_k^{\boldsymbol{X}}(f) \tag{103}$$

be a rank-r approximation satisfying equation 90. Assume a rank-r DeepONet

$$\mathcal{N}_r(f, \boldsymbol{X})(s) = \sum_{k=1}^{r} \tau_k(s) \rho_k(f; \boldsymbol{X}) \tag{104}$$

satisfies

$$\sup_{s \in \mathbb{T}} |a_k(s) - \tau_k(s)| \leq \varepsilon_k^{\mathrm{tr}}, \qquad \sup_{\|f\|_{\mathcal{U}} \leq 1} \left|c_k^{\boldsymbol{X}}(f) - \rho_k(f; \boldsymbol{X})\right| \leq \varepsilon_k^{\mathrm{br}}, \tag{105}$$

for some Banach space $\mathcal{U} \hookrightarrow L^2(\mathbb{T})$. Set

$$C_{\mathcal{U}} := \sup_{\|f\|_{\mathcal{U}} \leq 1} \|f\|_{L^2(\mathbb{T})}, \qquad A_k := \|a_k\|_{L^\infty(\mathbb{T})}, \qquad B_k^{\boldsymbol{X}} := \left\|c_k^{\boldsymbol{X}}\right\|_{\mathcal{U}^*}. \tag{106}$$

Then

$$\|\mathcal{R}_{\boldsymbol{X}} - \mathcal{N}_r(\cdot, \boldsymbol{X})\|_{\mathcal{U} \to L^\infty(\mathbb{T})} \leq E_{\mathrm{rank}}(r) + E_{\mathrm{net}}^{\mathrm{exact}}(r, \boldsymbol{X}), \tag{107}$$

where

$$E_{\mathrm{rank}}(r) := C_{\mathcal{U}} C_{\mathrm{rank},\infty} r^{-m}, \tag{108}$$

$$E_{\mathrm{net}}^{\mathrm{exact}}(r, \boldsymbol{X}) := \sum_{k=1}^{r} \left(B_k^{\boldsymbol{X}} \varepsilon_k^{\mathrm{tr}} + A_k \varepsilon_k^{\mathrm{br}} + \varepsilon_k^{\mathrm{tr}} \varepsilon_k^{\mathrm{br}}\right). \tag{109}$$

The same algebraic rate holds for the operator norm $\mathcal{U} \to L^2(\mathbb{T})$, after changing the constant.

Proof. Split the deterministic error as

$$\mathcal{R}_{\boldsymbol{X}} - \mathcal{N}_r = (\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X},r}) + (\mathcal{R}_{\boldsymbol{X},r} - \mathcal{N}_r). \tag{110}$$

The first term is bounded by Theorem 5.2:

$$\|\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X},r}\|_{\mathcal{U}\to L^\infty(\mathbb{T})} \le C_{\mathcal{U}} \|\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X},r}\|_{L^2(\mathbb{T})\to L^\infty(\mathbb{T})} \le C_{\mathcal{U}} C_{\mathrm{rank},\infty} r^{-m}. \tag{111}$$

For the second term, take $\|f\|_{\mathcal{U}} \le 1$ and expand

$$a_k c_k^{\boldsymbol{X}}(f) - \tau_k \rho_k = (a_k - \tau_k) c_k^{\boldsymbol{X}}(f) + \tau_k (c_k^{\boldsymbol{X}}(f) - \rho_k), \tag{112}$$

where $\rho_k = \rho_k(f; \boldsymbol{X})$. Using $\left|c_k^{\boldsymbol{X}}(f)\right| \le B_k^{\boldsymbol{X}}$ and

$$|\tau_k(s)| \le |a_k(s)| + |a_k(s) - \tau_k(s)| \le A_k + \varepsilon_k^{\mathrm{tr}}, \tag{113}$$

we obtain

$$\left|a_k c_k^{\boldsymbol{X}}(f) - \tau_k \rho_k\right| \le B_k^{\boldsymbol{X}} \varepsilon_k^{\mathrm{tr}} + (A_k + \varepsilon_k^{\mathrm{tr}}) \varepsilon_k^{\mathrm{br}}. \tag{114}$$

Summing over k and taking the supremum in s proves equation 107. The corresponding $\mathcal{U} \to L^2(\mathbb{T})$ estimate follows from $\|g\|_{L^2(\mathbb{T})} \le |\mathbb{T}|^{1/2} \|g\|_{L^\infty(\mathbb{T})}$. □

C.5 Finite-Node Deterministic Error

Let $s_j = \frac{2\pi j}{J}$, $j = 0, \dots, J-1$, and define

$$Z_J(\boldsymbol{X}) := \big(\boldsymbol{X}(s_0), \dots, \boldsymbol{X}(s_{J-1})\big) \in (\mathbb{R}^2)^J. \tag{115}$$

Let Π_J be a reconstruction operator, for example trigonometric interpolation, and set

$$\boldsymbol{X}_J := \Pi_J Z_J(\boldsymbol{X}). \tag{116}$$

Assumption C.6 (Geometry reconstruction accuracy)**.** *There exist constants C_Π and J_0 such that for every $\boldsymbol{X} \in \mathcal{A}_n$ and every $J \ge J_0$, the reconstruction $\boldsymbol{X}_J$ remains admissible and satisfies*

$$\boldsymbol{X}_J \in \mathcal{A}_n(2M, c_0/2, c_1/2), \tag{117}$$

and

$$\|\boldsymbol{X} - \boldsymbol{X}_J\|_{C^\nu(\mathbb{T})} \le C_\Pi J^{-(n-\nu)} \|\boldsymbol{X}\|_{C^n(\mathbb{T})} \tag{118}$$

For the discrete estimates below, we apply Definition C.3 on a class containing both $\boldsymbol{X}$ and its reconstruction $\boldsymbol{X}_J$, for example $\mathcal{A}_n(2M, c_0/2, c_1/2)$. The constants M_R and L_R may change on this enlarged class; only their uniform finiteness is used.

Assumption C.7 (Input quadrature accuracy)**.** *We use the same integer J for the boundary nodes and the input quadrature nodes. For some integer $q \ge 0$ and all $g \in C^q(\mathbb{T})$,*

$$\left| \int_{\mathbb{T}} g(t)\, dt - \sum_{j=0}^{J-1} \omega_j g(s_j) \right| \le C_Q J^{-q} \|g\|_{C^q(\mathbb{T})}. \tag{119}$$

Theorem C.8 (Discretized remainder approximation error)**.** *Assume Definition C.3, Assumption C.6, and Assumption C.7, and let $r \ge m$, $J \ge J_0$. Let the rank-r kernel approximation of $\mathcal{R}_{\boldsymbol{X}_J}$ be*

$$R_r(s, t; \boldsymbol{X}_J) = \sum_{k=1}^{r} a_k(s) b_k(t; \boldsymbol{X}_J), \tag{120}$$

and assume

$$\|R(\cdot, \cdot; \boldsymbol{X}_J) - R_r(\cdot, \cdot; \boldsymbol{X}_J)\|_{L^\infty(\mathbb{T}^2)} \le C_{\mathrm{ker},C^0} r^{-m}. \tag{121}$$

Also assume $b_k(\cdot; \boldsymbol{X}_J) \in C^q(\mathbb{T})$ for $k = 1, \dots, r$. Define the point-value DeepONet

$$\widehat{\mathcal{N}}_{r,J}(f, \boldsymbol{X})(s) := \sum_{k=1}^{r} \widehat{a}_k(s) \sum_{j=0}^{J-1} \omega_j \widehat{b}_k(s_j; Z_J(\boldsymbol{X})) f(s_j). \tag{122}$$

Assume

$$\sup_s |a_k(s) - \widehat{a}_k(s)| \le \delta_k^a, \qquad \sup_t \left| b_k(t; \boldsymbol{X}_J) - \widehat{b}_k(t; Z_J(\boldsymbol{X})) \right| \le \delta_k^b. \tag{123}$$

Then for every $f \in C^q(\mathbb{T})$,

$$\left\| \mathcal{R}_{\boldsymbol{X}} f - \widehat{\mathcal{N}}_{r,J}(f, \boldsymbol{X}) \right\|_{L^\infty(\mathbb{T})} \le \left(E_{\mathrm{geom}}(J) + E_{\mathrm{rank}}(r) + E_{\mathrm{quad}}(r, J) + E_{\mathrm{net}}^{\mathrm{disc}}(r, J) \right) \|f\|_{C^q(\mathbb{T})}, \tag{124}$$

where

$$E_{\mathrm{geom}}(J) := |\mathbb{T}| L_R C_\Pi M J^{-(n-\nu)}, \tag{125}$$

$$E_{\mathrm{rank}}(r) := C_{\mathrm{rank},C^0} r^{-m}, \tag{126}$$

$$C_{\mathrm{rank},C^0} := |\mathbb{T}| C_{\mathrm{ker},C^0}, \tag{127}$$

$$E_{\mathrm{quad}}(r, J) := C_Q C_{\mathrm{prod}}(q) J^{-q} \sum_{k=1}^{r} \|a_k\|_{L^\infty(\mathbb{T})} \|b_k(\cdot; \boldsymbol{X}_J)\|_{C^q(\mathbb{T})}, \tag{128}$$

$$E_{\mathrm{net}}^{\mathrm{disc}}(r, J) := \sum_{k=1}^{r} \left(C_\omega \|b_k(\cdot; \boldsymbol{X}_J)\|_{L^\infty} \delta_k^a + C_\omega \|a_k\|_{L^\infty} \delta_k^b + C_\omega \delta_k^a \delta_k^b \right), \tag{129}$$

where $C_\omega := \sum_{j=0}^{J-1} |\omega_j|$ and $C_{\mathrm{prod}}(q)$ is any constant such that

$$\|uv\|_{C^q(\mathbb{T})} \le C_{\mathrm{prod}}(q) \|u\|_{C^q(\mathbb{T})} \|v\|_{C^q(\mathbb{T})}. \tag{130}$$

Proof. Introduce the intermediate discrete separated operator

$$\mathcal{Q}_{r,J}^{\boldsymbol{X}} f(s) := \sum_{k=1}^{r} a_k(s) \sum_{j=0}^{J-1} \omega_j b_k(s_j; \boldsymbol{X}_J) f(s_j), \tag{131}$$

and decompose

$$\mathcal{R}_{\boldsymbol{X}} - \widehat{\mathcal{N}}_{r,J} = (\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X}_J}) + (\mathcal{R}_{\boldsymbol{X}_J} - \mathcal{R}_{\boldsymbol{X}_J,r}) + (\mathcal{R}_{\boldsymbol{X}_J,r} - \mathcal{Q}_{r,J}^{\boldsymbol{X}}) + (\mathcal{Q}_{r,J}^{\boldsymbol{X}} - \widehat{\mathcal{N}}_{r,J}). \tag{132}$$

The geometry term follows from equation 78 and equation 118:

$$\|R(\cdot, \cdot; \boldsymbol{X}) - R(\cdot, \cdot; \boldsymbol{X}_J)\|_{L^\infty(\mathbb{T}^2)} \le L_R C_\Pi M J^{-(n-\nu)}. \tag{133}$$

Hence, by equation 81,

$$\|(\mathcal{R}_{\boldsymbol{X}} - \mathcal{R}_{\boldsymbol{X}_J}) f\|_{L^\infty(\mathbb{T})} \le |\mathbb{T}| L_R C_\Pi M J^{-(n-\nu)} \|f\|_{C^0(\mathbb{T})} \le E_{\mathrm{geom}}(J) \|f\|_{C^q(\mathbb{T})}. \tag{134}$$

For the rank-truncation term, the kernel approximation assumption equation 121 and equation 81 give

$$\|(\mathcal{R}_{\boldsymbol{X}_J} - \mathcal{R}_{\boldsymbol{X}_J,r}) f\|_{L^\infty(\mathbb{T})} \le |\mathbb{T}| C_{\mathrm{ker},C^0} r^{-m} \|f\|_{C^0(\mathbb{T})} \le E_{\mathrm{rank}}(r) \|f\|_{C^q(\mathbb{T})}. \tag{135}$$

For the input quadrature term,

$$\begin{aligned} \left| \left(\mathcal{R}_{\boldsymbol{X}_J,r} f - \mathcal{Q}_{r,J}^{\boldsymbol{X}} f \right)(s) \right| \le \sum_{k=1}^{r} \|a_k\|_{L^\infty} \Bigg| \int_{\mathbb{T}} b_k(t; \boldsymbol{X}_J) f(t)\, dt \\ - \sum_{j=0}^{J-1} \omega_j b_k(s_j; \boldsymbol{X}_J) f(s_j) \Bigg|. \end{aligned} \tag{136}$$

Apply the quadrature assumption to

$$g_k(t) := b_k(t; \boldsymbol{X}_J) f(t). \tag{137}$$

Since $\|g_k\|_{C^q(\mathbb{T})} \le \|b_k(\cdot; \boldsymbol{X}_J)\|_{C^q(\mathbb{T})} \|f\|_{C^q(\mathbb{T})}$ up to a constant given by the Banach-algebra estimate for $C^q(\mathbb{T})$, this yields the bound $E_{\mathrm{quad}}(r, J) \|f\|_{C^q(\mathbb{T})}$.

For the neural realization term,

$$\left|\left(\mathcal{Q}^{\boldsymbol{X}}_{r,J} f - \widehat{\mathcal{N}}_{r,J}(f,\boldsymbol{X})\right)(s)\right| \le \sum_{k=1}^{r}\sum_{j=0}^{J-1} |\omega_j|\,|f(s_j)|\,|a_k b_k - \widehat{a}_k \widehat{b}_k|, \tag{138}$$

where

$$|a_k b_k - \widehat{a}_k \widehat{b}_k| \le |a_k|\delta_k^b + |b_k|\delta_k^a + \delta_k^a \delta_k^b. \tag{139}$$

Using $\max_j |f(s_j)| \le \|f\|_{C^q(\mathbb{T})}$ gives the bound

$$\left\|\mathcal{Q}^{\boldsymbol{X}}_{r,J} f - \widehat{\mathcal{N}}_{r,J}(f,\boldsymbol{X})\right\|_{L^\infty(\mathbb{T})} \le E^{\mathrm{disc}}_{\mathrm{net}}(r,J)\,\|f\|_{C^q(\mathbb{T})}\,. \tag{140}$$

Combining the four estimates proves equation 124. □

C.6 Training Residual and Full-Map Error

Let $\Theta_{r,J}$ be a class of low-rank DeepONets, and let μ describe the ensemble of boundary data and geometries used to measure error. Define the mean residual loss

$$\mathcal{L}_{\mathrm{pop}}(\theta) := \mathbb{E}_{(\boldsymbol{X},f)\sim\mu}\,\|\mathcal{R}_{\boldsymbol{X}} f - \mathcal{N}_\theta(f,\boldsymbol{X})\|^2_{L^2(\mathbb{T})}\,, \tag{141}$$

and its finite-sample counterpart

$$\mathcal{L}_{\mathrm{emp}}(\theta) := \frac{1}{N}\sum_{\ell=1}^{N} \left\|\mathcal{R}_{\boldsymbol{X}^\ell} f^\ell - \mathcal{N}_\theta(f^\ell,\boldsymbol{X}^\ell)\right\|^2_{L^2(\mathbb{T})}\,. \tag{142}$$

Theorem C.9 (Discretized remainder learning and full-map error)**.** *Assume $\widehat{\theta} \in \Theta_{r,J}$ satisfies*

$$\mathcal{L}_{\mathrm{emp}}(\widehat{\theta}) \le \inf_{\theta\in\Theta_{r,J}} \mathcal{L}_{\mathrm{emp}}(\theta) + \eta_{\mathrm{opt}}, \tag{143}$$

and set

$$\Delta_N := \sup_{\theta\in\Theta_{r,J}} |\mathcal{L}_{\mathrm{pop}}(\theta) - \mathcal{L}_{\mathrm{emp}}(\theta)|\,. \tag{144}$$

Suppose there exists $\theta^\star \in \Theta_{r,J}$ such that

$$\sup_{(\boldsymbol{X},f)\in\operatorname{supp}\mu} \|\mathcal{R}_{\boldsymbol{X}} f - \mathcal{N}_{\theta^\star}(f,\boldsymbol{X})\|_{L^2(\mathbb{T})} \le E_{\mathrm{det}}(r,J). \tag{145}$$

Then

$$\mathcal{L}_{\mathrm{pop}}(\widehat{\theta})^{1/2} \le E_{\mathrm{det}}(r,J) + \sqrt{2\Delta_N} + \sqrt{\eta_{\mathrm{opt}}}. \tag{146}$$

In the exact-geometry regime, E_{det} may be taken from Theorem C.5. In the fully discrete regime, it may be taken from Theorem C.8, after converting the $L^\infty(\mathbb{T})$ bound to $L^2(\mathbb{T})$ and imposing the corresponding bound on the input norm over $\operatorname{supp}\mu$.

If the final approximation uses the same exact evaluation of $2\mathcal{W}_0$ as the remainder target, then

$$\left(\mathbb{E}_{(\boldsymbol{X},f)\sim\mu}\left\|\breve{\Lambda}_{\boldsymbol{X}} f - \widehat{\Lambda}_{\boldsymbol{X}} f\right\|^2_{L^2(\mathbb{T})}\right)^{1/2} \le E_{\mathrm{det}}(r,J) + \sqrt{2\Delta_N} + \sqrt{\eta_{\mathrm{opt}}}. \tag{147}$$

If $2\mathcal{W}_0$ is truncated to $2\mathcal{W}_0^{(K)}$, add the root-mean-square Fourier term

$$\left(\mathbb{E}_{(\boldsymbol{X},f)\sim\mu}\left\|2\mathcal{W}_0 f - 2\mathcal{W}_0^{(K)} f\right\|^2_{L^2(\mathbb{T})}\right)^{1/2}. \tag{148}$$

Proof. Compare the finite-sample and mean residual losses. For every $\theta \in \Theta_{r,J}$,

$$\mathcal{L}_{\mathrm{pop}}(\widehat{\theta}) \le \mathcal{L}_{\mathrm{emp}}(\widehat{\theta}) + \Delta_N \le \mathcal{L}_{\mathrm{emp}}(\theta) + \eta_{\mathrm{opt}} + \Delta_N \le \mathcal{L}_{\mathrm{pop}}(\theta) + \eta_{\mathrm{opt}} + 2\Delta_N. \tag{149}$$

Taking $\theta = \theta^\star$ and using equation 145 gives

$$\mathcal{L}_{\mathrm{pop}}(\widehat{\theta}) \le E_{\mathrm{det}}(r,J)^2 + \eta_{\mathrm{opt}} + 2\Delta_N. \tag{150}$$

The square-root bound follows from $\sqrt{a+b+c} \le \sqrt{a} + \sqrt{b} + \sqrt{c}$ for nonnegative a, b, c.

For the full-map estimate, we use the exact splitting

$$\breve{\Lambda}_{\boldsymbol{X}} f - \widehat{\Lambda}_{\boldsymbol{X}} f = (2\mathcal{W}_0 f + \mathcal{R}_{\boldsymbol{X}} f) - (2\mathcal{W}_0 f + \mathcal{N}_{\widehat{\theta}}(f,\boldsymbol{X})) = \mathcal{R}_{\boldsymbol{X}} f - \mathcal{N}_{\widehat{\theta}}(f,\boldsymbol{X}). \tag{151}$$

Squaring, integrating over μ, and applying the residual bound proves equation 147. If $2\mathcal{W}_0$ is replaced by $2\mathcal{W}_0^{(K)}$, the same identity gains the additive term $2\mathcal{W}_0 f - 2\mathcal{W}_0^{(K)} f$, and Minkowski's inequality gives the stated Fourier contribution. □

D EXTENSION TO OTHER PDES

The decomposition equation 12 is not specific to the interior Laplace problem. The same idea extends to other elliptic equations and to exterior problems.

Helmholtz Equation. For the interior Helmholtz equation $(\Delta + k^2)u = 0$, the free-space Green's function is given in Appendix A.2. Near the origin, its leading singularity coincides with that of the two-dimensional Laplace Green's function:

$$G_k(\boldsymbol{x}) \sim \frac{1}{2\pi} \log r + \frac{1}{2\pi} \log\left(\frac{k}{2}\right) + \frac{\gamma}{2\pi} - \frac{i}{4}, \qquad r = |\boldsymbol{x}| \to 0^+. \tag{152}$$

Therefore, the same principal-part decomposition applies:

$$\breve{\Lambda}_{\boldsymbol{X},k} f = 2\mathcal{W}_0 f + \mathcal{R}_{\boldsymbol{X},k} f, \tag{153}$$

where $2\mathcal{W}_0$ is the universal singular operator in equation 12, while $\mathcal{R}_{\boldsymbol{X},k}$ is a smoother remainder.

Exterior Problems. The same argument applies to exterior formulations. For the exterior Helmholtz problem, the DtN map has the form

$$\breve{\Lambda}^{\mathrm{ext}}_{\boldsymbol{X},k} f = -2\mathcal{W}_0 f + \mathcal{R}^{\mathrm{ext}}_{\boldsymbol{X},k} f. \tag{154}$$

A detailed derivation is provided following.

Let $\Omega \subset \mathbb{R}^2$ be a bounded simply connected domain with smooth boundary $\Gamma = \partial\Omega$, and denote its exterior by

$$\Omega^c := \mathbb{R}^2 \setminus \overline{\Omega}. \tag{155}$$

We consider the exterior Helmholtz problem

$$(\Delta + k^2)u = 0 \quad \text{in } \Omega^c, \tag{156}$$

subject to the Dirichlet boundary condition

$$u = g \quad \text{on } \Gamma, \tag{157}$$

together with the Sommerfeld radiation condition

$$\lim_{r\to\infty} r^{1/2}\left(\frac{\partial u}{\partial r} - iku\right) = 0, \quad r = |x|. \tag{158}$$

This radiation condition selects the unique outgoing solution.

For the exterior problem, the Dirichlet-to-Neumann (DtN) mapping is defined by

$$\Lambda^{\mathrm{ext}}_k[g] := \partial_n u\big|_\Gamma, \tag{159}$$

where ∂_n denotes the *outward normal derivative with respect to* Ω (equivalently, the inward normal derivative with respect to Ω^c). As in the interior case, the operator

$$\Lambda^{\mathrm{ext}}_k : H^{1/2}(\Gamma;\mathbb{C}) \to H^{-1/2}(\Gamma;\mathbb{C}) \tag{160}$$

is linear and complex-valued.

We represent the exterior solution using a double-layer potential

$$u(x) = \mathcal{K}_k(\varphi)(x) = \int_\Gamma \frac{\partial G_k(x-y)}{\partial n_y}\,\varphi(y)\,ds_y, \quad x \in \Omega^c. \tag{161}$$

This representation satisfies equation 156 and the Sommerfeld radiation condition equation 158.

The double-layer potential admits the following jump relation across Γ:

$$\lim_{x\to\Gamma,\, x\in\Omega^c} u = \left(-\frac{1}{2}\mathcal{I} + \mathcal{K}_k\right)\varphi. \tag{162}$$

Imposing the Dirichlet boundary condition equation 157, we obtain the second-kind boundary integral equation

$$\left(-\frac{1}{2}\mathcal{I} + \mathcal{K}_k\right)\varphi = g \quad \text{on } \Gamma, \tag{163}$$

assuming that k avoids the corresponding internal resonant frequencies. Using the exact same derivation as in Section 4.1 and Appendix B, we can obtain equation 154.

E Detailed Network Architectures and Hyperparameters

Within each direct/PPD pair, the trainable backbone and parameter count are identical. The two models differ only in the training target and in the analytic subtraction/reconstruction step. PPDNO-Direct and PPDNO likewise share an architecture. Table 4 records the task-specific constructors. For FNO, the listed parameter count follows stored PyTorch parameters; its spectral weights are complex. The DeepONet branch and trunk networks, and the PPDNO geometry and trunk networks, each use four hidden layers with ReLU activations. Every FNO uses four spectral blocks, four pointwise blocks, ReLU activations, and a width-128 pointwise projection before the final output. OT uses ReLU activations, feed-forward width 256, and zero dropout; its layer counts below refer to encoder/decoder layers.

Table 4: Network specifications and trainable parameter counts. A PPD+baseline has the same entry as its direct counterpart.

Task	Model	Network specification	Parameters
Laplace ellipse	DeepONet	rank 256; hidden width 256	723,712
	FNO	9 Fourier modes; channel width 100	413,857
	OT	$d = 64$; 8 encoder/4 decoder layers; 8 heads	684,161
	PPDNO / PPDNO-Direct	rank 12; hidden width 144	646,620
Laplace trig. polynomial	DeepONet	rank 256; hidden width 420	1,600,292
	FNO	9 Fourier modes; channel width 150	920,657
	OT	$d = 128$; 8 encoder/4 decoder layers; 8 heads	1,889,025
	PPDNO / PPDNO-Direct	rank 20; hidden width 240	1,709,380
Helmholtz $k = 1$	FNO	10 Fourier modes; channel width 150; 5 harmonics	1,012,436
	PPDNO / PPDNO-Direct	rank 14; hidden width 196; 5 harmonics	1,751,988
Helmholtz $k = 10$	FNO	10 Fourier modes; channel width 240; 10 harmonics	2,572,466
	PPDNO / PPDNO-Direct	rank 32; hidden width 256; 10 harmonics	4,758,592

The Helmholtz FNO and low-rank models receive the same task-specific coordinate encoding

$$\gamma(s) = (\cos s, \sin s, \ldots, \cos(n_{\max} s), \sin(n_{\max} s)), \tag{164}$$

with $n_{\max} = 5$ for $k = 1$ and 10 for $k = 10$. The Laplace models retain their original scalar coordinate interfaces; no harmonic encoding is added. The rank-20/width-240 low-rank model in the trigonometric-polynomial task was chosen so that its real parameter storage remains below that of the corresponding FNO.

Table 5 summarizes the optimization schedules for all learned models.

F Training Records

All 24 primary training runs completed their prescribed schedules, and all recorded losses remained finite. Table 6 reports the best-checkpoint epoch, validation MSE, and training-loop wall-clock time from the saved training histories. The timings include the training and validation passes and checkpoint writes, but exclude data generation. All runs used the same NVIDIA GeForce RTX 3080. On the Laplace tasks, FNO and OT use mini-batches because larger batches sharply increase memory

Table 5: Approved optimization schedules. "Full" denotes the complete stored training split. All runs use Adam, MSE, and the best validation checkpoint. Primary results use run seed 2027; Appendix H reports additional training runs.

Task	Model family	Batch	Epochs	Initial LR	Plateau patience	Factor
Ellipse	DeepONet / PPD+DeepONet / PPDNO-Direct / PPDNO	Full	20,000	10^{-4}	1,000	0.8
Ellipse	FNO / PPD+FNO	1,000	500	10^{-4}	100	0.8
Ellipse	OT / PPD+OT	500	200	10^{-4}	100	0.8
Trig. polynomial	DeepONet / PPD+DeepONet / PPDNO-Direct / PPDNO	Full	20,000	10^{-4}	1,000	0.8
Trig. polynomial	FNO / PPD+FNO	1,000	300	10^{-4}	100	0.8
Trig. polynomial	OT / PPD+OT	250	100	10^{-4}	100	0.8
Helmholtz $k = 1, 10$	FNO / PPD+FNO	1,024	1,000	5×10^{-5}	100	0.6
Helmholtz $k = 1, 10$	PPDNO-Direct / PPDNO	1,024	5,000	5×10^{-5}	100	0.6

use and computation time, whereas DeepONet and PPDNO permit full-batch training. Epoch counts are therefore not directly comparable across architectures; in wall-clock time, FNO and OT were generally trained longer, as shown in Table 6. Full-map and remainder-target losses have different target scales and should not be compared as if they were the same objective.

G Additional Experimental Details

G.1 Datasets and Function Families

The training-data generator uses seed 42 for ellipses and seed 33 for the other tasks. The fixed test data and the primary training run both use seed 2027; Appendix H reports independent training runs. Training contains $5{,}000 \times 10$ samples for the ellipse and each Helmholtz wave number, and $10{,}000 \times 10$ for the trigonometric-polynomial task. Each post-training test profile contains $10{,}000$ samples, with one boundary datum per geometry. ID-fixed is generated independently from the same geometry and function distributions as training. Functional Dirichlet-OOD retains the ID geometry and changes only the function family, whereas Joint-OOD combines that function shift with the task-specific OOD geometry in Appendix G.2.

Both Laplace tasks use random linear combinations of logarithmic fundamental solutions for training and ID-fixed:

$$u(x) = \sum_{j=1}^{M} \frac{c_j}{2\pi} \log |x - y_j|, \qquad M \sim \mathrm{Unif}\{1, 2, 3, 4\}. \tag{165}$$

The raw coefficient vector is sampled componentwise from a centered uniform distribution and then rescaled to an independently sampled $\|c\|_2 \sim \mathcal{U}(0.1, 0.5)$.

Laplace on ellipses. For each training or ID-fixed ellipse, let $R_{\max} = \max_{x \in \Gamma} |x|$. Each source angle is uniform on $[0, 2\pi)$ and its radius is uniform on $[1.1R_{\max}, 2R_{\max}]$, placing every source

Table 6: Final training records, including the best checkpoint epoch, validation MSE, and training-loop wall-clock time.

Task	Model	Objective	Epoch		Best validation MSE	Time (min)
			Best	Total		
Ellipse	DeepONet	Full map	19,963	20,000	4.47×10^{-5}	5.4
	PPD+DeepONet	Remainder	19,783	20,000	1.02×10^{-7}	5.0
	FNO	Full map	500	500	1.08×10^{-6}	20.5
	PPD+FNO	Remainder	484	500	7.95×10^{-8}	20.6
	OT	Full map	195	200	3.96×10^{-5}	173.7
	PPD+OT	Remainder	200	200	3.11×10^{-6}	173.5
	PPDNO-Direct	Full map	19,901	20,000	2.97×10^{-5}	6.4
	PPDNO	Remainder	19,879	20,000	1.26×10^{-8}	6.4
Trig. polynomial	DeepONet	Full map	5,528	20,000	3.76×10^{-5}	15.7
	PPD+DeepONet	Remainder	18,070	20,000	4.50×10^{-7}	15.8
	FNO	Full map	273	300	4.20×10^{-7}	34.5
	PPD+FNO	Remainder	281	300	2.94×10^{-8}	34.5
	OT	Full map	43	100	7.19×10^{-4}	212.3
	PPD+OT	Remainder	89	100	2.40×10^{-5}	214.1
	PPDNO-Direct	Full map	20,000	20,000	1.89×10^{-5}	21.1
	PPDNO	Remainder	19,950	20,000	3.02×10^{-8}	21.1
Helmholtz $k = 1$	FNO	Full map	996	1,000	2.17×10^{-4}	60.0
	PPD+FNO	Remainder	989	1,000	1.56×10^{-4}	60.1
	PPDNO-Direct	Full map	4,989	5,000	2.60×10^{-3}	62.5
	PPDNO	Remainder	4,997	5,000	3.18×10^{-5}	61.8
Helmholtz $k = 10$	FNO	Full map	997	1,000	2.35×10^{-3}	87.6
	PPD+FNO	Remainder	978	1,000	1.66×10^{-3}	87.5
	PPDNO-Direct	Full map	4,944	5,000	1.60×10^{-3}	64.6
	PPDNO	Remainder	4,974	5,000	1.52×10^{-3}	64.9

outside the domain. Functional Dirichlet-OOD uses the same ID ellipse distribution, while Joint-OOD uses the ellipse shell specified in Appendix G.2; both use the common Laplace OOD family below.

Laplace on trigonometric-polynomial boundaries. For training and ID-fixed, one to four source points are sampled uniformly in area in the certified annulus $[1.1R_{\mathrm{out}}, 2R_{\mathrm{out}}]$ about the curve center, where R_{out} is the analytic outer-radius certificate. Functional Dirichlet-OOD retains the ID range $q \in [0.10, 0.50]$, whereas Joint-OOD uses $q \in [0.60, 0.75]$; their function family is again the common Laplace OOD family.

Laplace Functional Dirichlet-OOD and Joint-OOD functions. Each profile is an exactly balanced mixture of two real harmonic families. For a boundary with stored center c_X and $R_X = \max_{x\in\Gamma} |x - c_X|$, write $z = x_1 + ix_2$. The multipole subtype contains one to three terms

$$u(z) = \Re \sum_j c_j e^{i\varphi_j} \left(\frac{z - y_j}{R_X} \right)^{-m_j}, \qquad m_j \in \{1, 2\}, \tag{166}$$

where the poles are uniform in area in the annulus $[1.5R_X, 2.5R_X]$. The exponential subtype contains two or three terms

$$u(z) = \Re \sum_j c_j e^{i\varphi_j} \exp\left(\lambda_j e^{-i\vartheta_j} \frac{z - c_X}{R_X} \right), \tag{167}$$

with λ_j log-uniform on $[0.25, 1.5]$ and directions uniform on $[0, 2\pi)$. The phases φ_j are also uniform on $[0, 2\pi)$. In both subtypes the real coefficient vector is normalized to unit ℓ^2 norm before amplitude scaling. Each sample is assigned $A \sim \mathrm{LogUniform}[0.01, 0.08]$ and scaled so that

$$\|f\|_{L^2_{\Gamma},\mathrm{av}} = \left(\frac{\sum_\ell |f(t_\ell)|^2 |X'(t_\ell)|}{\sum_\ell |X'(t_\ell)|} \right)^{1/2} = A. \tag{168}$$

The same scalar multiplies the analytic field, Dirichlet trace, and exact Neumann trace. Each OOD profile contains exactly 5,000 multipoles and 5,000 exponentials, with zero rejected proposals and no model-based sample selection.

Helmholtz at $k = 1$ and $k = 10$. Training and ID-fixed use one to five outgoing monopoles,

$$u(x) = \sum_{j=1}^{M} c_j \left(-\frac{i}{4} \right) H_0^{(1)}(k|x - y_j|), \qquad M \sim \mathrm{Unif}\{1, \ldots, 5\}. \tag{169}$$

The raw real coefficient vector is sampled componentwise from $\mathcal{U}(-5, 5)$ and rescaled to an independent $\|c\|_2 \sim \mathcal{U}(1, 5)$. Source points are uniform in area in the disk of radius $0.8r_{\mathrm{in}}$ about the curve center, where r_{in} is the certified safe inner radius. The $k = 1$ and $k = 10$ tasks use the same geometry, source, and coefficient draws and differ only in the wave number used to evaluate the field.

For both wave numbers, Functional Dirichlet-OOD and Joint-OOD replace the monopole combination with one unit-coefficient order-two outgoing mode

$$u(x) = H_2^{(1)}(k\rho) e^{2i\theta}, \qquad \rho = |x - y|, \qquad e^{i\theta} = \frac{(x_1 - y_1) + i(x_2 - y_2)}{\rho}, \tag{170}$$

whose interior source point y is sampled by the same safe area-uniform disk rule. For outward normal n and $d = x - y$,

$$\partial_n u = e^{2i\theta} \left[k H_2^{(1)\prime}(k\rho) \frac{d \cdot n}{\rho} + 2i H_2^{(1)}(k\rho) \frac{-d_2 n_1 + d_1 n_2}{\rho^2} \right]. \tag{171}$$

We apply no per-sample normalization, rejection, component mixing, or model- or FFT-based sample selection.

G.2 Geometry Families

Ellipses. Training, ID-fixed, and Functional Dirichlet-OOD use $X(t) = (a\cos t, b\sin t)$ with independent $a, b \sim \mathcal{U}(0.5, 1.5)$. Joint-OOD samples (a, b) uniformly by rejection from $[0.4, 1.6]^2 \setminus [0.5, 1.5]^2$, so every Joint-OOD ellipse lies outside the closed training parameter box in at least one coordinate.

Trigonometric-polynomial boundaries. For equation 32, draw $(p_1, p_2, p_3) \sim \text{Dirichlet}(1, 1, 1)$ and independent $\theta_j \sim \mathcal{U}(0, 2\pi)$, then set

$$\beta = qp_1 e^{i\theta_1}, \qquad \alpha_2 = \frac{qp_2}{2} e^{i\theta_2}, \qquad \alpha_3 = \frac{qp_3}{3} e^{i\theta_3}. \tag{172}$$

Also draw $R \sim \mathcal{U}(0.8, 1.2)$, each component of c from $\mathcal{U}(-0.3, 0.3)$, and $\phi \sim \mathcal{U}(0, 2\pi)$. Training, ID-fixed, and Functional Dirichlet-OOD use $q \in [0.10, 0.50]$; Joint-OOD uses $q \in [0.60, 0.75]$, leaving $(0.50, 0.60)$ unused. For distinct unit-circle points z, w, $|Z(z) - Z(w)| \geq R(1-q)|z-w| > 0$ and $|Z'(t)| \geq R(1-q) > 0$. With $s = |\beta| + |\alpha_2| + |\alpha_3|$, the certified radial bounds are $R(1-s) \leq |Z(t) - c| \leq R(1+s)$. Paired ID and Joint draws share $(p_j, \theta_j, R, c, \phi)$ and differ only through the mapped q interval.

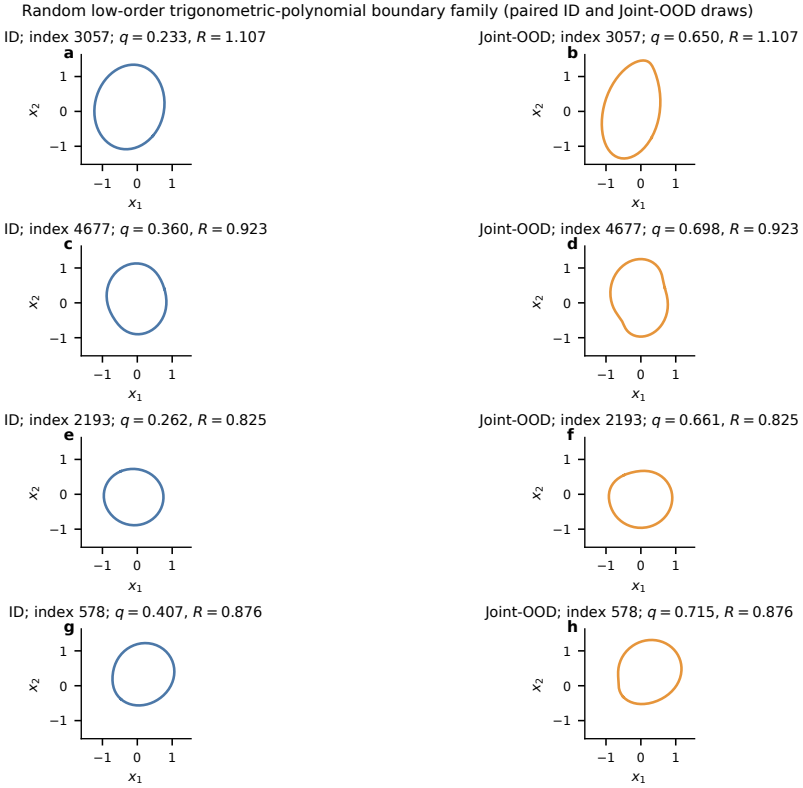


Figure 5: Fixed-seed paired examples from the trigonometric-polynomial family. Each row shares the coefficient allocation, phases, radius, center, and rotation; only q is mapped from the ID interval $[0.10, 0.50]$ to the Joint-OOD interval $[0.60, 0.75]$. Common axes expose the geometry shift without error-based sample selection.

Periodic-Gaussian-RBF boundaries. Draw n uniformly from $\{5, 6, 7, 8\}$ and define separated-sector control points

$$\theta_i = \frac{2\pi}{n}(i + \delta_i), \quad \delta_i \sim \mathcal{U}(-0.25, 0.25), \qquad P_i = r_i(\cos\theta_i, \sin\theta_i), \quad r_i = 1 + au_i, \quad u_i \sim \mathcal{U}(-1, 1). \tag{173}$$

Periodic nodes $\tau_i \in [0, 2\pi)$ are assigned by cumulative centripetal chord length, i.e., with increments proportional to $\sqrt{\|P_{i+1} - P_i\|_2}$. With

$$K(t, \tau) = \exp\{\kappa_{\text{geo}}[\cos(t - \tau) - 1]\}, \qquad \kappa_{\text{geo}} = 2.5, \tag{174}$$

the vector coefficients $C_j \in \mathbb{R}^2$ solve the interpolation system

$$\sum_{j=0}^{n-1} K(\tau_i, \tau_j) C_j = P_i, \qquad i = 0, \ldots, n-1, \tag{175}$$

and the physical boundary is

$$X(t) = c + RQ_\phi \sum_{j=0}^{n-1} C_j K(t, \tau_j). \tag{176}$$

Here Q_ϕ is the planar rotation through ϕ, with $R \sim \mathcal{U}(0.8, 1.2)$, $\phi \sim \mathcal{U}(0, 2\pi)$, and each component of c uniform on $[-0.3, 0.3]$. Training, ID-fixed, and Functional Dirichlet-OOD use $a \in [0.10, 0.25]$; Joint-OOD uses $a \in [0.30, 0.35]$, leaving $(0.25, 0.30)$ unused. The $k = 1$ and $k = 10$ tasks share all geometry draws. Paired ID and Joint curves share n, δ_i, u_i, R, c, and ϕ, and differ only through the amplitude mapping.

We accept a proposal only when both mapped curves have positive certified lower bounds for radius, speed, and the star-shaped cross product. The certification combines a 4,096-point grid with global analytic derivative bounds, ensuring a regular simple star-shaped boundary and a safe inner disk for source placement.

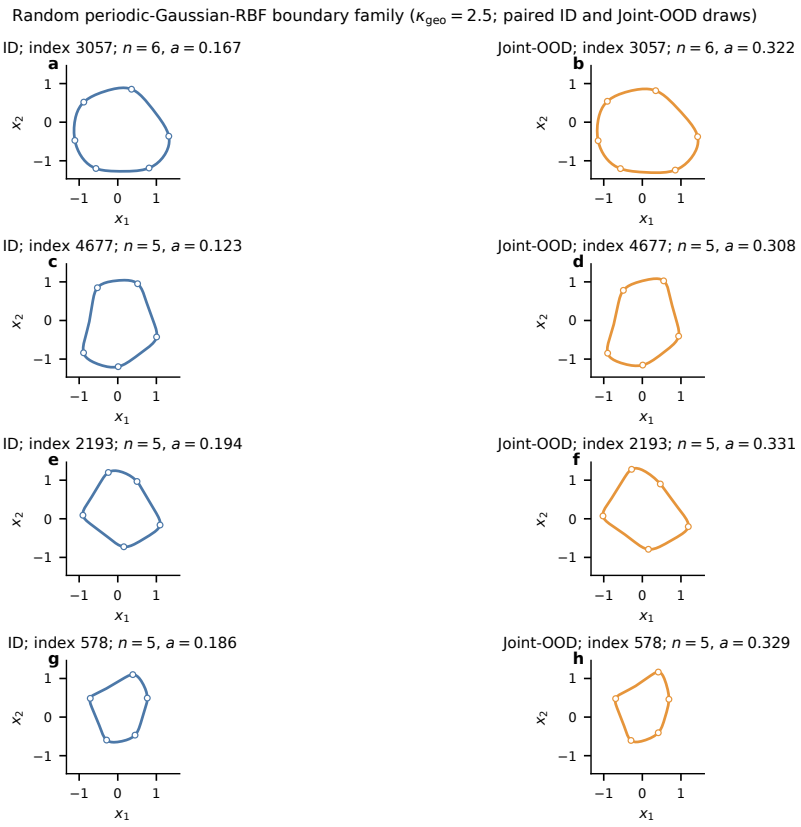


Figure 6: Fixed-seed paired examples from the periodic-Gaussian-RBF family. Control points are shown. Each row shares n, angular perturbations, radial directions, scale, center, and rotation; only a is mapped from ID $[0.10, 0.25]$ to Joint-OOD $[0.30, 0.35]$.

G.3 SAMPLE-WISE ERROR DISPERSION

Tables 7 and 8 report mean ± standard deviation across the 10,000 fixed samples. These standard deviations describe variation across test cases; they do not measure uncertainty across independent training runs.

Table 7: Laplace sample-wise errors (%), reported as mean ± standard deviation over the 10,000 fixed samples. Each method has separate rows for relative L^2 and relative maximum error.

Task	Method	Metric	ID-fixed	Functional Dirichlet-OOD	Joint-OOD
Ellipse	BIE	L^2	.0129±.0087	.0120±.0095	.0164±.0158
		Max	.0231±.0226	.0248±.0244	.0384±.0429
Ellipse	DeepONet	L^2	14.062±10.800	13.152±11.623	15.290±11.242
		Max	16.944±11.998	15.529±11.548	19.025±13.311
Ellipse	PPD+DeepONet	L^2	1.168±1.220	1.564±1.923	4.888±6.633
		Max	1.082±1.305	1.491±1.961	5.176±7.654
Ellipse	FNO	L^2	2.738±2.280	5.039±4.926	7.128±6.533
		Max	3.627±3.359	7.019±6.856	9.177±8.312
Ellipse	PPD+FNO	L^2	1.128±1.154	1.592±1.932	4.785±5.842
		Max	1.216±1.373	1.753±2.155	5.407±7.553
Ellipse	OT	L^2	18.994±10.809	25.673±14.131	30.459±18.797
		Max	23.790±14.363	33.198±18.710	41.370±30.196
Ellipse	PPD+OT	L^2	7.765±6.084	9.517±8.644	19.211±17.959
		Max	7.358±7.307	9.592±10.757	21.203±24.481
Ellipse	FFT-only	L^2	28.422±24.053	25.143±23.309	47.584±38.538
		Max	20.040±20.539	19.857±21.577	42.671±40.490
Ellipse	PPDNO-Direct	L^2	9.321±9.678	6.397±8.646	6.431±6.904
		Max	10.910±10.574	6.965±7.509	7.432±6.451
Ellipse	PPDNO	L^2	.385±.380	.443±.502	2.662±3.954
		Max	.407±.484	.482±.579	2.766±4.010
Trig.	DeepONet	L^2	17.925±10.613	22.248±14.876	33.682±21.294
		Max	19.989±12.179	23.565±15.016	37.906±27.046
Trig.	PPD+DeepONet	L^2	2.084±2.820	2.508±3.581	11.928±18.638
		Max	1.753±2.594	2.162±3.307	12.456±21.959
Trig.	FNO	L^2	1.673±1.626	4.061±4.792	7.646±8.238
		Max	2.269±2.445	5.712±6.554	9.960±10.513
Trig.	PPD+FNO	L^2	.672±.696	.863±1.135	5.079±8.767
		Max	.727±.867	.943±1.323	5.975±10.753
Trig.	OT	L^2	100.008±.313	100.003±.389	100.005±.522
		Max	100.001±.291	100.013±.410	100.019±.579
Trig.	PPD+OT	L^2	16.629±15.972	13.885±14.577	38.831±37.268
		Max	11.528±12.859	10.452±12.747	36.695±41.655
Trig.	FFT-only	L^2	16.627±15.973	13.884±14.579	38.830±37.271
		Max	11.525±12.860	10.449±12.749	36.694±41.659
Trig.	PPDNO-Direct	L^2	10.049±7.755	10.832±9.964	19.441±16.436
		Max	12.097±8.773	11.927±9.396	22.357±18.612
Trig.	PPDNO	L^2	.603±.807	.618±.914	4.746±7.880
		Max	.540±.821	.579±.924	5.037±8.797

Table 8: Helmholtz sample-wise errors (%), reported as mean ± standard deviation over the 10,000 fixed samples. Each method has separate rows for relative L^2 and relative maximum error.

k	Method	Metric	ID-fixed	Functional Dirichlet-OOD	Joint-OOD
1	FNO	L^2	2.669±2.076	80.637±9.263	82.049±9.449
		Max	3.780±3.038	86.273±6.771	86.748±7.182
1	PPD+FNO	L^2	2.352±1.635	24.703±4.528	26.000±5.288
		Max	2.942±1.881	21.416±6.991	22.589±8.233
1	FFT-only	L^2	96.378±10.565	14.598±9.336	15.097±9.713
		Max	84.535±26.487	8.638±6.544	8.975±6.816
1	PPDNO-Direct	L^2	6.899±6.448	45.678±23.705	49.836±23.568
		Max	10.159±8.322	41.753±19.072	45.369±19.066
1	PPDNO	L^2	1.100±.511	2.065±.662	2.646±1.103
		Max	1.330±.758	1.728±.802	2.248±1.387
10	FNO	L^2	2.575±1.063	4.372±5.099	6.072±6.537
		Max	3.989±1.819	7.514±7.887	10.103±9.486
10	PPD+FNO	L^2	2.215±.853	3.207±2.770	4.514±3.672
		Max	3.421±1.447	5.362±3.750	7.402±4.798
10	FFT-only	L^2	108.351±3.645	110.582±4.312	112.473±4.670
		Max	102.663±13.499	98.680±9.279	100.174±12.283
10	PPDNO-Direct	L^2	2.057±1.187	3.766±4.604	5.798±5.965
		Max	3.214±1.939	5.835±6.392	9.021±8.151
10	PPDNO	L^2	2.031±1.210	2.962±2.476	5.095±4.048
		Max	3.106±1.858	4.540±3.033	8.045±5.926

G.4 Inference and BIE Protocols

Neural inference is timed with PyTorch 2.7 under WSL2 on a single NVIDIA GeForce RTX 3080. Inputs are already on the GPU; disk I/O and host-device transfer are excluded. Each method uses batch 2,048 and 10 warmup batches. Throughput is averaged over three complete passes through the fixed 10,000-sample set, with CUDA synchronization after every batch. Principal-part-decomposed methods include the neural forward pass, FFT principal part, and reconstruction. At batch 2,048, peak allocated GPU memory for PPD+FNO versus PPDNO is respectively 1.23/0.08 GiB on ellipses, 1.82/0.10 GiB on trigonometric polynomials, 1.83/0.12 GiB at $k = 1$, and 2.90/0.20 GiB at $k = 10$.

The ellipse BIE reference uses a periodic Nyström discretization in double precision. Every geometry–datum pair is an independent cold start: geometry-dependent matrices are assembled, the dense system is solved, and the Neumann trace is recovered without caching. It runs in vectorized batches of eight with eight-thread Intel MKL on CPU; three complete repetitions are timed. ID-fixed throughput is 430.10 samples/s. Functional and Joint-OOD throughput is 415.88 and 415.61 samples/s, respectively. CPU BIE and GPU neural timings characterize their stated environments and are not a hardware-normalized speedup claim.

G.5 Zero-Shot Super-Resolution

The fixed set contains 100 ellipse Functional Dirichlet-OOD samples (50 multipoles and 50 exponentials). Geometry and input data remain at $N_{\text{in}} = 256$. See Table 9. For each output resolution, the same stored analytic function is reevaluated on the refined grid; the principal part uses spectral zero-padding and the low-rank trunk is queried at the new coordinates.

Table 9: Mean relative L^2 error (%) for zero-shot output refinement.

Method	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
FFT-only	23.5264	23.5264	23.5264	23.5264
PPDNO-Direct	6.9107	6.9243	6.9300	6.9327
PPDNO	**0.4161**	**0.4163**	**0.4164**	**0.4164**

H Independent Training Runs

Except for the additional runs below, all learned-model results, sample-wise analyses, visualizations, and ablations in the paper use checkpoints trained with seed 2027. This training-run seed is distinct from the fixed data-generation seeds specified in Appendix G.1. We repeated every learned method with run seeds 2028 and 2029, while keeping the stored training data, training/validation split, and three $10{,}000$-sample test profiles fixed. Thus only the network initialization and stochastic optimization trajectory change.

Tables 10 and 11 report the additional runs. Each entry before and after the slash corresponds to run seeds 2028 and 2029, respectively. The best-validation MSE is the value used for checkpoint selection. Full-map and remainder-target objectives have different target scales, so these validation losses should only be interpreted within a given model and objective. BIE and FFT-only are deterministic, untrained references and are therefore not repeated here.

Table 10: Additional Laplace training runs. Test columns give mean relative L^2 error (%). Each cell reports run seeds 2028/2029 in that order; boldface marks the lower learned error separately for each run.

Task	Method	Best validation MSE	ID-fixed	Functional Dirichlet-OOD	Joint-OOD
Ellipse	DeepONet	5.70×10^{-5} / 6.07×10^{-5}	16.295 / 17.286	14.792 / 16.015	17.402 / 17.261
	PPD+DeepONet	9.76×10^{-8} / 9.57×10^{-8}	1.124 / 1.114	1.514 / 1.515	4.617 / 4.707
	FNO	1.03×10^{-6} / 9.70×10^{-7}	2.671 / 2.535	5.037 / 4.841	7.012 / 6.734
	PPD+FNO	7.62×10^{-8} / 8.88×10^{-8}	1.135 / 1.213	1.602 / 1.700	4.912 / 5.022
	OT	4.01×10^{-5} / 4.14×10^{-5}	21.536 / 20.825	28.704 / 29.203	33.545 / 33.745
	PPD+OT	2.26×10^{-6} / 1.97×10^{-6}	5.616 / 5.358	7.101 / 6.983	16.139 / 16.032
	PPDNO-Direct	2.95×10^{-5} / 2.96×10^{-5}	9.205 / 9.228	6.181 / 6.217	6.480 / 6.183
	PPDNO	1.10×10^{-8} / 1.24×10^{-8}	**0.348 / 0.379**	**0.402 / 0.439**	**2.434 / 2.493**
Trig. polynomial	DeepONet	7.80×10^{-6} / 4.62×10^{-5}	8.205 / 18.556	11.700 / 21.870	19.775 / 34.225
	PPD+DeepONet	4.53×10^{-7} / 4.58×10^{-7}	2.119 / 2.147	2.555 / 2.569	11.912 / 11.854
	FNO	4.36×10^{-7} / 4.38×10^{-7}	1.806 / 1.899	4.236 / 4.308	7.948 / 7.925
	PPD+FNO	2.60×10^{-8} / 2.78×10^{-8}	**0.621 / 0.637**	0.800 / 0.836	4.956 / 5.008
	OT	7.19×10^{-4} / 7.19×10^{-4}	100.003 / 100.003	100.001 / 100.001	100.001 / 100.003
	PPD+OT	2.40×10^{-5} / 2.40×10^{-5}	16.628 / 16.627	13.884 / 13.884	38.831 / 38.831
	PPDNO-Direct	1.75×10^{-5} / 2.37×10^{-5}	9.709 / 11.780	10.430 / 12.935	19.206 / 21.292
	PPDNO	3.15×10^{-8} / 3.22×10^{-8}	0.624 / 0.639	**0.641 / 0.651**	**4.829 / 4.855**

Table 11: Additional exterior Helmholtz training runs. Test columns give mean relative L^2 error (%). Each cell reports run seeds 2028/2029 in that order; boldface marks the lower learned error separately for each run.

k	Method	Best validation MSE	ID-fixed	Functional Dirichlet-OOD	Joint-OOD
1	FNO	2.79×10^{-4} / 2.43×10^{-4}	3.052 / 2.833	82.163 / 79.313	83.497 / 80.364
	PPD+FNO	8.95×10^{-5} / 7.05×10^{-5}	1.842 / 1.609	18.920 / 22.596	19.516 / 23.752
	PPDNO-Direct	2.60×10^{-3} / 2.60×10^{-3}	6.904 / 6.906	47.056 / 46.260	51.259 / 50.433
	PPDNO	4.14×10^{-5} / 3.47×10^{-5}	**1.195 / 1.142**	**2.309 / 2.164**	**2.906 / 2.739**
10	FNO	2.31×10^{-3} / 1.99×10^{-3}	2.545 / 2.348	4.315 / 3.957	6.003 / 5.565
	PPD+FNO	1.70×10^{-3} / 1.71×10^{-3}	2.258 / 2.270	3.284 / 3.288	**4.661 / 4.658**
	PPDNO-Direct	1.60×10^{-3} / 1.59×10^{-3}	2.069 / 2.045	3.547 / 3.625	5.580 / 5.647
	PPDNO	1.54×10^{-3} / 1.52×10^{-3}	**2.043 / 2.028**	**2.798 / 3.052**	4.837 / 5.201