Basis-Adaptive Learned Asymptotic Nonlocal Inversion Neural Operator (BALANI-NO) for Elliptic PDE Surrogates and Gravitational Lensing

Anonymous authors
Paper under double-blind review

Abstract

Spectral neural operators parameterize each layer by a learned Fourier multiplier, or symbol, which is usually a lookup table over the lowest modes. For many elliptic operators the symbol is instead a rational function of frequency. We introduce BALANI-NO, a neural operator whose symbol is the sum of a learned pole (rational) branch in the isotropic magnitude $|k|$ and a small low-mode table. The pole branch specifies the symbol at all frequencies from a few coefficients, favors no coordinate axis, and does not depend on the grid. We compare BALANI-NO with ten baselines at matched parameter budgets, with seven seeds each, on Darcy flow, Navier–Stokes vorticity, and gravitational-lensing deflection fields, where models trained on warm dark matter are tested on cold and axion dark matter. BALANI-NO has the lowest zero-shot error on Darcy (0.1194 vs. 0.1300 for FNO), on Navier–Stokes, and on every zero-shot lensing split. It also has the lowest in-distribution error on Navier–Stokes and subhalo lensing, and ties with U-FNO on full-field lensing. Two smaller-scale additional experiments, on real (non-synthetic) cosmological density maps and real geomagnetic survey data, support the same pattern of best or near-best zero-shot transfer.

1 Introduction and Related Work

The microphysics of dark matter remains unknown. Direct-detection experiments (Drukier et al., 1986; Goodman & Witten, 1985; Akerib et al., 2017; Cui et al., 2017; Aprile et al., 2018) and collider searches (Aaboud et al., 2019) have placed increasingly stringent constraints on its properties, motivating complementary approaches that probe dark matter through its gravitational effects. One such approach is strong gravitational lensing, in which the gravity of a foreground galaxy bends and magnifies the light from a more distant background galaxy. The lensed images are sensitive to the presence of low-mass dark-matter subhalos within the foreground galaxy, allowing their abundance and mass distribution to be studied, which in turn can help to distinguishing dark-matter models, since warm dark matter (WDM) suppresses the formation of small-scale structure relative to cold dark matter (CDM) (Bode et al., 2001; Mahdi et al., 2016).

Machine learning is now widely used to extract such signals. Convolutional networks have been trained to detect substructure directly in strong-lens images (Diaz Rivero & Dvorkin, 2020; Varma et al., 2020). Subsequent work has classified the morphology of substructure (Alexander et al., 2020a), learned it without supervision (Alexander et al., 2020b), exploited astrometric signatures (Vattis et al., 2021), and used equivariant networks (Cheeramvelil et al., 2023) and self-supervised representations (Deshmukh et al., 2023). Physics-informed models have also been proposed, including a vision transformer (Veloso et al., 2023; Dosovitskiy et al., 2020) and a physics-informed neural network (Ojha et al., 2024).

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.

Neural operators learn mappings between function spaces. DeepONet (Lu et al., 2021) implements the operator approximation framework of Chen & Chen (1995) using branch and trunk networks to encode input functions and query coordinates, respectively. Extensions include multiple-input operators (Jin et al., 2022), POD-based bases (Lu et al., 2022), and learned bases for mesh-free settings (Zhang et al., 2022). Another class of neural operators parameterizes integral kernels, including graph-kernel operators (Li et al., 2020) and the Fourier neural operator (FNO) (Li et al., 2021), which represents the kernel in Fourier space. Kovachki et al. (2023) provide a general framework with universal approximation and discretization invariance, while convolutional neural operators (Raonić et al., 2023) extend convolutional networks to band-limited functions across resolutions. Standardized benchmarks are available through PDEBench (Takamoto et al., 2022), and the neuraloperator library (Kossaifi et al., 2026) provides implementations and the Darcy and Navier–Stokes datasets used here.

$W(k)$ specifies how a spectral operator acts on each Fourier mode k and determines its frequency response. Fourier-based neural operators mainly differ in how they parameterize $W(k)$ and incorporate it into the network architecture. FNO (Li et al., 2021) represents $W(k)$ with a learned matrix for each retained mode. This gives a translation-invariant kernel. Dynamic Schwartz–Fourier operator (Gao et al., 2025) extends this formulation by introducing interactions between Fourier modes. U-FNO (Wen et al., 2022) and UNO (Rahman et al., 2023) embed spectral blocks within encoder–decoder architectures with pooling and upsampling. UNO further represents $W(k)$ using a bank of Gaussian filters. HC-UNO predicts the filter geometry from the input and is related to hypernetworks (Ha et al., 2017) and adaptive frequency filters (Huang et al., 2023). LNO (Cao et al., 2024) parameterizes $W(k)$ using poles and residues along a single frequency axis. ALNO combines this representation with a Fourier multiplier. PDNO (Shin et al., 2022) learns $W(k)$ with a neural network using Fourier-feature encoding of the frequency (Tancik et al., 2020).

BALANI-NO uses a different parameterization. Its poles depend on the isotropic quantity $|k|^2$ rather than a single frequency axis. The numerator depends polynomially on (ik_x, ik_y). A small learned table provides an additional low-frequency correction. This isotropic parameterization is suited to the static isotropic fields considered here and provides a direct comparison with LNO and ALNO. Further details are given in Appendix C.

2 ARCHITECTURE OF BALANI-NO

Let $a \in \mathbb{R}^{C_{in}\times H\times W}$ denote the input field(s) and $u \in \mathbb{R}^{C_{out}\times H\times W}$ the target field(s), discretized on a regular grid. We seek a learned operator $\mathcal{N}_\Theta : a \mapsto \hat{u}$ that approximates the true solution operator $\mathcal{G} : a \mapsto u$. The channel counts, resolution and dataset sizes vary by task. We write $k = (k_x, k_y)$ for a 2D Fourier index, measured in cycles per domain length so that it does not depend on the grid, $|k|^2 = k_x^2 + k_y^2$, $\mathcal{F}$ and $\mathcal{F}^{-1}$ for the (real) 2D Fourier transform and its inverse, σ = GELU, and GN for GroupNorm.

BALANI-NO is a stack of spectral blocks. Inside a block all channels share a common width C. Given a learned, frequency-dependent symbol $W(k) \in \mathbb{C}^{C\times C}$, the spectral convolution is

$$v(x) = \mathcal{F}^{-1}\big[W(k)\,\mathcal{F}[h](k)\big](x), \tag{1}$$

and the block combines v with a local (1×1 convolution) skip connection, normalization and nonlinearity:

$$\mathcal{B}(h; W(\cdot)) = \sigma\Big(\mathrm{GN}\big(v + W_{1\times1}\,h\big)\Big). \tag{2}$$

The network lifts the input, concatenated with two coordinate channels, to width C with a 1×1 convolution, applies L blocks $h_\ell = \mathcal{B}(h_{\ell-1}; W_\ell)$, and maps back to C_{out} channels with a 1×1 convolution–GELU–1×1 convolution head. The Darcy and Navier–Stokes models use GroupNorm with two groups; the lensing models omit it and use $L = 4$ (Appendix D). All baselines except CNN, DeepONet and POD-DeepONet share this template and differ only in how $W(k)$ is parameterized (Appendix B).

BALANI-NO combines a learned pole (rational) branch, placed in the isotropic magnitude $|k|$, with a standard low-mode Fourier table:

$$W_{\text{BALANI}}(k) = R(k) + T(k). \tag{3}$$

The pole branch is a partial-fraction expansion with J learned poles,

$$R(k) = \sum_{j=1}^{J} \frac{N_j(ik_x, ik_y)}{|k|^2 + s_j^2}, \qquad N_j(ik_x, ik_y) = \sum_{p+q\le \deg} c_{j,p,q}\,(ik_x)^p (ik_y)^q, \qquad k \neq 0, \tag{4}$$

with coefficient matrices $c_{j,p,q} \in \mathbb{C}^{C\times C}$ and pole locations $s_j^2 = \text{softplus}(\hat{s}_j) > 0$, shared by all channel pairs of a block. At $k = 0$ the value $R(0) = \sum_j c_{j,0,0}/s_j^2$ is tied to the pole locations, so we replace it by a separate learned offset $D \in \mathbb{C}^{C\times C}$. The table branch is a truncated multiplier in the style of FNO (Li et al., 2021),

$$T(k) = \begin{cases} T_l(k_x, k_y), & (k_x, k_y) \in \mathcal{K}_m, \\ 0, & \text{otherwise}, \end{cases} \qquad T_l(k_x, k_y) \in \mathbb{C}^{C\times C}, \tag{5}$$

over the retained set $\mathcal{K}_m = \{0 \le k_x < m,\ -m \le k_y < m\}$, with independent weights for the $k_y \ge 0$ and $k_y < 0$ halves. It provides a complementary low-frequency correction to R.

The parameterization is based on isotropy, asymptotic behavior, and discretization independence. Since the poles depend on $|k|^2$, each term corresponds to a screened-Poisson resolvent $(s_j^2 - \Delta)^{-1}$ followed by a differential operator of order at most deg. This treats all coordinate directions equivalently. For the lensing problem, the mapping $\kappa \mapsto \boldsymbol{\alpha}$ has symbol proportional to $ik/|k|^2$. This corresponds to the limit $s_j \to 0$ of a single degree-one term.

At large wavenumbers $|k| \gg s_j$, the rational component satisfies $R(k) \approx \sum_j N_j(ik)/|k|^2$. It therefore has a homogeneous dependence on k with degree up to $\deg -2$. The branch represents the symbol over the full frequency range using $J\binom{\deg+2}{2}$ coefficient matrices. A finite spectral table instead provides coefficients only within its retained frequency range.

The parameterization is independent of the grid dimensions (H, W). The function $R(k)$ can be evaluated on any Fourier grid and T acts on a fixed set of modes. The same learned operator can therefore be evaluated at different resolutions by changing the FFT grid without resampling the learned weights. LNO and ALNO parameterize their poles along a single axis that serves as a formal Laplace variable (Appendix C). This differs from the isotropic structure of the static fields considered here. All hyperparameters are given in Appendix D.

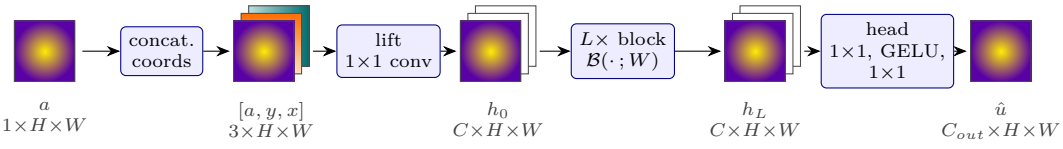

Figure 1: Skeleton shared by BALANI-NO, FNO, ALNO, LNO and PDNO. Normalised pixel coordinates are appended to the input, a pointwise 1×1 convolution lifts the result to C channels, L blocks (Eq. equation 2) keep the shape unchanged, and a pointwise head maps to C_{out} channels. Only the symbol $W(k)$ inside each block differs between the five models (Figures 12–16). None of the layers depends on H or W, so the same weights apply at any resolution.

3 Experiments

We evaluate BALANI-NO against ten baselines on three tasks: 2D Darcy flow, 2D Navier–Stokes vorticity, and gravitational-lensing deflection fields. The Darcy and Navier–Stokes experiments assess zero-shot transfer across resolutions. The lensing experiment evaluates transfer across dark-matter models by training on warm dark matter (WDM) and testing

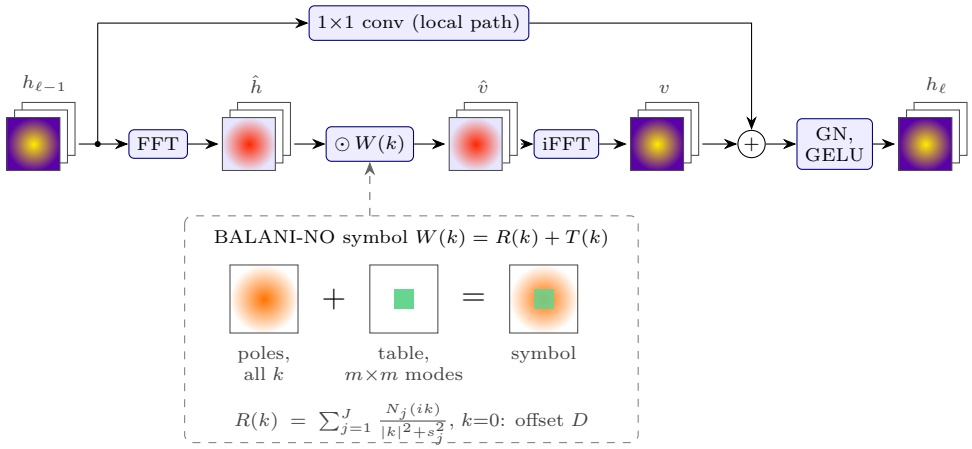

Figure 2: One BALANI-NO block. The symbol adds a rational pole term, defined at every frequency and radial in $|k|$, to a small table on the lowest modes.

on cold dark matter (CDM) and axion dark matter without retraining. We also ablate the two branches of the BALANI-NO symbol and evaluate whether the learned poles recover known screening lengths from synthetic Green's-function data.

All eleven architectures are sized to BALANI-NO's parameter budget (within +6.2% on Darcy and Navier–Stokes, and −1.9% to +10.4% on lensing); configurations are in Appendix D. We report the relative L_2 error $|\hat{u} - u|_2/|u|_2$. For Darcy and Navier–Stokes, we compute this ratio for each sample and average it over each test batch and then across batches. For lensing, we compute the ratio of the summed numerator and denominator norms over the full test set. We define degradation as the ratio of the mean zero-shot error to the mean in-distribution error. This ratio should be considered together with the absolute errors since a model with large errors in both settings can still have a degradation close to 1×.

All models are trained for 60 epochs using a cosine-annealed learning rate with an initial value of 10^{-3} and weight decay of 10^{-5}. Darcy and Navier–Stokes experiments use AdamW with batch size 16 and MSE loss. Lensing experiments use Adam with batch size 32 and relative-L_2 loss with gradient-norm clipping at 1.0. We retain the checkpoint with the lowest in-distribution test error. No separate validation set is used, so the in-distribution results may be mildly optimistic across all models. Zero-shot test sets are excluded from model selection. Each configuration is trained with seven seeds: $7, 13, 42, 64, 97, 128, 2024$. The seeds affect initialization and data ordering while keeping the train/test split fixed. Tables report the mean ± standard deviation across seeds using the population standard deviation. All experiments are run on a single NVIDIA T4 GPU.

3.1 Darcy flow

We consider the steady-state Darcy equation $-\nabla \cdot (a(x)\nabla u(x)) = f(x)$ on the unit square with Dirichlet boundary conditions, where a is a piecewise-constant diffusion coefficient and u the pressure field. We use the benchmark of Li et al. (2021), obtained through the neuraloperator library from a public Zenodo archive. Models are trained at 16×16 on 1,000 samples and evaluated in-distribution at 16×16 (100 samples) and zero-shot at 32×32 (50 samples).

Table 1 reports a zero-shot error of 0.1194 for BALANI-NO compared with 0.1300 for FNO and 0.1353 for ALNO. Its in-distribution error is 0.0901 compared with 0.0775 for CNN and 0.0861 for PDNO. Among the pole-residue methods, BALANI-NO has lower errors than ALNO and LNO at both resolutions. CNN has the largest degradation (8.61×) because its 3×3 convolutions are tied to the 16×16 grid, while PDNO degrades by 4.79× because its frequency representation depends on the grid's Nyquist frequency.

Table 1: Relative L_2 error on Darcy@16 (in-distribution) and Darcy@32 (zero-shot), mean ± std over 7 seeds. Degradation is the ratio of zero-shot to in-distribution error.

Architecture	Params	Darcy@16 (test)	Darcy@32 (0-shot)	Degradation
BALANI-NO	112,387	0.0901 ± 0.0012	$\mathbf{0.1194} \pm 0.0021$	1.33×
FNO	111,857	0.1028 ± 0.0036	0.1300 ± 0.0046	1.26×
ALNO	115,169	0.1092 ± 0.0043	0.1353 ± 0.0037	1.24×
LNO	113,257	0.1420 ± 0.0021	0.1561 ± 0.0015	1.10×
PDNO	116,465	0.0861 ± 0.0025	0.4120 ± 0.0143	4.79×
UNO	115,265	0.1022 ± 0.0040	0.1788 ± 0.0183	1.75×
U-FNO	119,309	0.0980 ± 0.0027	0.3319 ± 0.0170	3.39×
HC-UNO	112,065	0.0925 ± 0.0030	0.1525 ± 0.0133	1.65×
CNN	112,961	$\mathbf{0.0775} \pm 0.0013$	0.6677 ± 0.0193	8.61×
DeepONet	112,833	0.1898 ± 0.0023	0.2053 ± 0.0029	1.08×
POD-DeepONet	112,176	0.1466 ± 0.0032	0.1660 ± 0.0023	1.13×

3.2 Navier–Stokes

We consider the 2D incompressible Navier–Stokes equations in vorticity form, using the dataset distributed with the neuraloperator library (Kossaifi et al., 2026) on a public zenodo archive, which was generated by a classical solver at Reynolds number 500. The task maps a vorticity field to the vorticity field at a later time. We use the 128×128 version with 1,000 training and 200 test samples. All fields are truncated in Fourier space to the Nyquist band of the target grid, with a normalization that keeps amplitudes independent of resolution: 32×32 for training and in-distribution testing, and 64×64 for zero-shot evaluation. The zero-shot set is the same 200 test fields at 64×64.

Table 2: Relative L_2 error on Navier–Stokes@32 (in-distribution) and @64 (zero-shot), mean ± std over 7 seeds. Degradation is the ratio of zero-shot to in-distribution error.

Architecture	Params	NS@32 (test)	NS@64 (0-shot)	Degradation
BALANI-NO	112,387	$\mathbf{0.1245} \pm 0.0033$	$\mathbf{0.1247} \pm 0.0033$	1.00×
FNO	111,857	0.1466 ± 0.0055	0.1467 ± 0.0055	1.00×
ALNO	115,169	0.2080 ± 0.0037	0.2080 ± 0.0037	1.00×
LNO	113,257	0.3808 ± 0.0047	0.3809 ± 0.0047	1.00×
PDNO	116,465	0.1374 ± 0.0059	0.4835 ± 0.0122	3.52×
UNO	115,265	0.2276 ± 0.0184	0.4028 ± 0.0127	1.77×
U-FNO	119,309	0.1454 ± 0.0043	0.2799 ± 0.0153	1.92×
HC-UNO	112,065	0.1323 ± 0.0079	0.1562 ± 0.0076	1.18×
CNN	112,961	0.1810 ± 0.0020	0.5773 ± 0.0051	3.19×
DeepONet	112,833	0.5292 ± 0.0027	0.5313 ± 0.0026	1.00×
POD-DeepONet	112,176	0.5090 ± 0.0010	0.5069 ± 0.0015	1.00×

Table 2 shows that BALANI-NO achieves the lowest error at both resolutions (0.1245 in-distribution and 0.1247 zero-shot) with essentially no resolution degradation. FNO, ALNO, LNO, and the two branch–trunk models also show little change with resolution. BALANI-NO has the lowest in-distribution error, followed by HC-UNO (0.1323) and PDNO (0.1374). At 64×64, FNO (0.1467) and HC-UNO (0.1562) remain above the BALANI-NO seed spread. The fields are smooth at this Reynolds number, with less than 0.05% of the spectral energy in a representative 64×64 test target at $16 \leq |k| < 32$. Thus, the $32 \to 64$ experiment primarily tests discretization consistency rather than extrapolation to unresolved scales. PDNO and CNN show larger degradation (3.52× and 3.19×) because their parameters depend on the input grid. In PDNO, the input representation is defined relative to the grid's Nyquist frequency, while CNN uses fixed-size spatial kernels tied to the training grid. Pooling introduces further resolution dependence by changing the spatial scale of feature representations, contributing to the degradation of UNO (1.77×), U-FNO (1.92×), and HC-UNO. UNO also shows large seed variance. DeepONet and POD-DeepONet have low degradation but errors above 0.5 at both resolutions, indicating consistently high errors rather than strong resolution transfer.

3.3 Gravitational lensing

The lensing task maps a convergence map κ to the deflection field $\boldsymbol{\alpha} = \nabla\psi = (\alpha_x, \alpha_y)$, where ψ is the lensing potential. Since $\nabla^2\psi \propto \kappa$, in Fourier space $\hat{\boldsymbol{\alpha}}(k) \propto i\,k\,|k|^{-2}\,\hat{\kappa}(k)$. The solution operator is therefore a rational function of k with a degree-one numerator, which is the form of the pole branch in Eq. equation 4. We consider two variants: a full-field variant ($\kappa \to \boldsymbol{\alpha}$) and a subhalo variant ($\kappa_{\text{sub}} \to \boldsymbol{\alpha}_{\text{sub}}$), which isolates the substructure contribution.

The data are simulated with lenstronomy (Birrer & Amara, 2018; Birrer et al., 2021) and provide κ and ψ maps at native resolution 128×128 for three dark-matter models: WDM, CDM and axion. All fields are resampled bilinearly to 64×64, and the deflection is computed by finite differences of ψ on this grid, in grid units up to a fixed constant. Relative errors are unaffected by that constant.

We train on WDM and test on held-out WDM samples (in-distribution), then evaluate the same checkpoints on CDM and axion samples. These come from different dark-matter models and are never used for training or checkpoint selection. We pool use 2,000 files for training and 250 for testing. The zero-shot sets contain 250 CDM and 250 axion samples.

Table 3: Full-field lensing ($\kappa \to \boldsymbol{\alpha}$): relative L_2 error on WDM (train distribution, held-out) and zero-shot CDM and Axion, mean $\pm$ std over 7 seeds. Degradation is relative to the WDM error. BALANI-NO has 832,890 parameters.

Architecture	WDM	CDM (0-shot)	Axion (0-shot)	Deg. (CDM)	Deg. (Axion)
BALANI-NO	**0.0038** $\pm$ 0.0001	**0.0060** $\pm$ 0.0002	**0.0064** $\pm$ 0.0004	1.58×	1.68×
FNO	0.0046 $\pm$ 0.0000	0.0084 $\pm$ 0.0002	0.0086 $\pm$ 0.0003	1.83×	1.87×
ALNO	0.0081 $\pm$ 0.0003	0.0138 $\pm$ 0.0004	0.0154 $\pm$ 0.0007	1.70×	1.90×
LNO	0.0144 $\pm$ 0.0004	0.0238 $\pm$ 0.0006	0.0262 $\pm$ 0.0010	1.65×	1.82×
PDNO	0.0067 $\pm$ 0.0003	0.0093 $\pm$ 0.0014	0.0099 $\pm$ 0.0029	1.39×	1.48×
UNO	0.0089 $\pm$ 0.0007	0.0160 $\pm$ 0.0009	0.0184 $\pm$ 0.0016	1.80×	2.07×
U-FNO	0.0039 $\pm$ 0.0001	0.0092 $\pm$ 0.0004	0.0109 $\pm$ 0.0004	2.36×	2.79×
HC-UNO	0.0072 $\pm$ 0.0008	0.0145 $\pm$ 0.0013	0.0154 $\pm$ 0.0012	2.01×	2.14×
CNN	0.0090 $\pm$ 0.0006	0.0150 $\pm$ 0.0011	0.0210 $\pm$ 0.0022	1.67×	2.33×
DeepONet	0.0202 $\pm$ 0.0008	0.0325 $\pm$ 0.0006	0.0446 $\pm$ 0.0016	1.61×	2.21×
POD-DeepONet	0.0184 $\pm$ 0.0002	0.0315 $\pm$ 0.0002	0.0455 $\pm$ 0.0014	1.71×	2.47×

Table 4: Subhalo lensing ($\kappa_{\text{sub}} \to \boldsymbol{\alpha}_{\text{sub}}$): same protocol and format as Table 3. Best value per error column in bold.

Architecture	WDM	CDM (0-shot)	Axion (0-shot)	Deg. (CDM)	Deg. (Axion)
BALANI-NO	**0.0058** $\pm$ 0.0002	**0.0109** $\pm$ 0.0003	**0.0124** $\pm$ 0.0006	1.88×	2.14×
FNO	0.0173 $\pm$ 0.0001	0.0387 $\pm$ 0.0009	0.0310 $\pm$ 0.0009	2.24×	1.79×
ALNO	0.0243 $\pm$ 0.0008	0.0457 $\pm$ 0.0014	0.0479 $\pm$ 0.0022	1.88×	1.97×
LNO	0.0478 $\pm$ 0.0013	0.0839 $\pm$ 0.0020	0.0992 $\pm$ 0.0039	1.76×	2.08×
PDNO	0.0081 $\pm$ 0.0004	0.0147 $\pm$ 0.0022	0.0221 $\pm$ 0.0077	1.81×	2.73×
UNO	0.0315 $\pm$ 0.0025	0.0635 $\pm$ 0.0035	0.0879 $\pm$ 0.0084	2.02×	2.79×
U-FNO	0.0124 $\pm$ 0.0002	0.0331 $\pm$ 0.0013	0.0480 $\pm$ 0.0013	2.67×	3.87×
HC-UNO	0.0200 $\pm$ 0.0021	0.0448 $\pm$ 0.0041	0.0627 $\pm$ 0.0051	2.24×	3.14×
CNN	0.0373 $\pm$ 0.0024	0.0624 $\pm$ 0.0047	0.0953 $\pm$ 0.0059	1.67×	2.55×
DeepONet	0.0469 $\pm$ 0.0019	0.0940 $\pm$ 0.0018	0.0769 $\pm$ 0.0015	2.00×	1.64×
POD-DeepONet	0.0512 $\pm$ 0.0006	0.1061 $\pm$ 0.0006	0.0938 $\pm$ 0.0032	2.07×	1.83×

Results. BALANI-NO has the lowest error on every zero-shot split in both variants (Tables 3 and 4). In the full-field variant it is 29% and 26% below the best baseline (FNO) on CDM and Axion (0.0060 vs. 0.0084; 0.0064 vs. 0.0086). On WDM it ties with U-FNO (0.0038 ± 0.0001 vs. 0.0039 ± 0.0001), nonetheless, UFNO degrades to 0.0092 and 0.0109 on CDM and Axion. In the subhalo variant, which is a harder task for all models, BALANI-NO's margin over the runner-up (PDNO) is 28% on WDM, 26% on CDM and 44% on Axion.

PDNO has the lowest degradation in the full-field setting (1.39× and 1.48×), but its zero-shot errors (0.0093 and 0.0099) are 55% higher than those of BALANI-NO. The effect of the distribution shift also varies across models. In the subhalo setting, FNO, DeepONet, and POD-DeepONet have lower errors on Axion than on CDM, while the other models show higher errors. The WDM, CDM, and axion cases share the same $\kappa \to \boldsymbol{\alpha}$ operator and differ

only in the input distribution. The experiment therefore measures the robustness of the learned operator to changes in input statistics.

3.4 Ablation: pole and table branches

BALANI-NO's symbol is $W(k) = R(k) + T(k)$, a pole branch plus a low-mode table (Section 2). We train poles only (T=0), table only (R=0), and the full model, each with seven seeds, using the recipe of Section 3 for Darcy and Navier–Stokes and the lensing ablation schedule. To separate the benefit of combining branches from that of added capacity, we also train parameter-matched poles-only and table-only variants, enlarged (more poles/degree, or a larger retained-mode window m) to roughly BALANI-NO's own parameter count. The 1×1 skip, normalization and depth are unchanged in every variant.

Table 5: Darcy and Navier–Stokes branch-composition ablation, mean ± std over 7 seeds. Best value per column in bold.

		Darcy		Navier–Stokes	
Variant	Params	@16	@32 (0-shot)	@32	@64 (0-shot)
Poles only	50,179	0.0891 ± 0.0017	0.1172 ± 0.0008	0.1517 ± 0.0020	0.1519 ± 0.0020
Poles only, matched	113,429	**0.0888** ± 0.0030	**0.1200** ± 0.0028	0.1514 ± 0.0023	0.1516 ± 0.0023
Table only	63,775	0.1023 ± 0.0021	0.1281 ± 0.0016	0.1764 ± 0.0013	0.1764 ± 0.0013
Table only, matched	111,857	0.1038 ± 0.0017	0.1323 ± 0.0044	0.1446 ± 0.0046	0.1447 ± 0.0046
Both (BALANI-NO)	112,387	0.0904 ± 0.0013	0.1207 ± 0.0016	**0.1223** ± 0.0022	**0.1224** ± 0.0022

Table 6: Lensing branch-composition ablation, full-field and subhalo variants, mean ± std over 7 seeds. Best value per column in bold.

		Full-field			Subhalo		
Variant	Params	WDM	CDM (0-shot)	Axion (0-shot)	WDM	CDM (0-shot)	Axion (0-shot)
Poles only	308,602	0.0071 ± 0.0002	0.0116 ± 0.0003	0.0130 ± 0.0007	0.0073 ± 0.0002	0.0118 ± 0.0002	0.0143 ± 0.0007
Poles only, matched	849,318	0.0072 ± 0.0004	0.0141 ± 0.0002	0.0200 ± 0.0007	0.0076 ± 0.0008	0.0138 ± 0.0002	0.0198 ± 0.0007
Table only	529,762	0.0149 ± 0.0001	0.0329 ± 0.0023	0.0262 ± 0.0008	0.0150 ± 0.0002	0.0320 ± 0.0016	0.0275 ± 0.0007
Table only, matched	824,674	0.0118 ± 0.0002	0.0288 ± 0.0010	0.0226 ± 0.0010	0.0117 ± 0.0002	0.0282 ± 0.0010	0.0216 ± 0.0013
Both (BALANI-NO)	832,890	**0.0059** ± 0.0001	**0.0113** ± 0.0004	**0.0125** ± 0.0006	**0.0057** ± 0.0003	**0.0112** ± 0.0002	**0.0132** ± 0.0006

Across the ten evaluation columns, Both branches combined achieves the lowest error in eight, with the two exceptions occurring on Darcy. On Navier–Stokes, the full model consistently outperforms both single-branch variants, including the parameter-matched pole-only model (0.1223 vs. 0.1514 at @32 and 0.1224 vs. 0.1516 at @64). On lensing, it achieves the lowest error on all six splits, including relative to the parameter-matched pole-only model, with reductions of approximately 19–38% on the full-field splits and 19–33% on the subhalo splits. These gains at comparable parameter budgets indicate that the improvement is not attributable solely to the additional capacity of the combined model.

Overall, the results suggest that the two branches provide complementary information, with their combination yielding substantial gains on Navier–Stokes and lensing, while Darcy performance remains dominated by the pole-only variants.

4 Discussion

The Darcy ablation in Section 3.4 reflects a pattern also observed in the two additional tasks in Appendix E and Appendix F. BALANI-NO has the lowest zero-shot error across all four tasks, while its training-resolution error is lowest on only one. Table 7 summarizes these results. On Darcy, CAMELS, and EMAG2, CNN or PDNO achieves lower error at the training resolution but degrades substantially after a resolution change. CNN uses local kernels tied to the grid, while PDNO uses an unconstrained frequency-dependent representation evaluated at the training-grid frequencies. BALANI-NO gives up some training-resolution accuracy while maintaining stronger transfer across resolutions.

We do not attribute the three cases where BALANI-NO is not the training-resolution winner to a single mechanism. The underlying cause differs across tasks.

Table 7: Training-resolution vs. zero-shot winners across all four near-elliptic tasks. "Training-res. winner" is evaluated at the resolution/region used for model selection in each task (Sections 3.1, 3.2, 3.3, Appendices E, F); for EMAG2 this set is additionally held out geographically from training even at matched resolution (Appendix F). For lensing, "training-res." is the WDM split and "zero-shot" is CDM; the Axion split and the full numbers for both variants are in Tables 3–4.

Task	Training-res. winner	BALANI-NO rank (training-res.)	Zero-shot winner
Darcy (§3.1)	CNN (0.0775)	3rd (0.0901)	BALANI-NO (0.1194)
CAMELS gravity (App. E)	PDNO (0.2256)	2nd (0.2318)	BALANI-NO (0.1769)
EMAG2 geomag (App. F)	CNN (0.0067)	4th (0.0113)	BALANI-NO (0.0399)
Navier–Stokes (§3.2)	BALANI-NO (0.1245)	1st	BALANI-NO (0.1247)
Lensing, full-field (§3.3)	BALANI-NO (0.0038)	1st (tied U-FNO, 0.0039)	BALANI-NO (0.0060)
Lensing, subhalo (§3.3)	BALANI-NO (0.0058)	1st	BALANI-NO (0.0109)

Darcy: no global symbol exists. The Darcy coefficient field $a(x)$ varies spatially, so the solution operator is not shift-invariant and does not have a single frequency-domain symbol. Any choice of $W(k)$, whether a pole parameterization, a spectral table, or their combination, therefore provides a fixed input-independent approximation to an operator that varies across realizations. CNN's local kernels and PDNO's per-frequency MLP offer greater flexibility on the finite training grid and can adapt to the coefficient fields seen during training. This flexibility also leads to larger degradation when the grid changes (Table 1).

CAMELS: the input transform changes the learned map. The physical operator in this task is a 2D Poisson solve and therefore lies within the regime targeted by the pole branch, corresponding to a single pole at $s_j \to 0$ with a degree-zero numerator. The network input is $\log_{10} \rho$ rather than ρ, while the Poisson solve is linear in ρ (Appendix E). The mapping from the network input to the target is therefore nonlinear even though the underlying physical operator is linear. This mismatch makes the pole parameterization less direct at the training resolution. The resulting limitation differs from the spatially varying coefficient in Darcy but leads to a similar reduction in training-resolution accuracy.

EMAG2: a linear global operator with heterogeneous inputs. The upward-continuation operator is linear and globally shift-invariant, so the pole parameterization matches the structure of the physical operator. The input distribution is more complex because it consists of real magnetic-anomaly measurements from heterogeneous marine, airborne, and satellite surveys (Appendix F). These measurements contain broadband and locally varying structure that a smooth low-order radial filter may not capture as well at the training resolution. CNN and PDNO can adapt more closely to this structure on the training grid, but this flexibility leads to larger degradation when the grid changes.

Navier–Stokes: low high-frequency content and a compatible operator structure. Two factors may explain why Navier–Stokes is the exception. First, less than 0.05% of the spectral energy in a representative target lies in $16 \leq |k| < 32$ at this Reynolds number, so the fields are dominated by low frequencies. Second, the linear diffusion term has Fourier symbol $\exp(-\nu|k|^2\Delta t)$, which is a smooth function of $|k|^2$ that a small sum of poles can approximate. The nonlinear advection term can then be represented by the repeated local nonlinear path outside $W(k)$. This gives a possible correspondence between the architecture and the equation, with the pole branch representing diffusion and the nonlinear path representing advection. We have not tested this interpretation against the learned pole locations, so it remains a hypothesis.

Lensing: The lensing deflection operator $\kappa \mapsto \boldsymbol{\alpha}$ has Fourier symbol $ik|k|^{-2}$, which matches the pole branch in Eq. equation 4 with $s_j \to 0$. The operator is fixed and shift-invariant across samples, and the network receives κ directly without an input transform. This provides the closest match to the assumptions of the pole branch among the tasks considered here. BALANI-NO also performs well at the training resolution and has the lowest zero-shot error across both lensing variants. On full-field WDM, it ties with U-FNO (0.0038 vs. 0.0039), while on subhalo WDM it achieves 0.0058, which is 28% below PDNO. CNN

and PDNO perform well at the training resolution on Darcy, CAMELS, and EMAG2 but do not show the same advantage here. Their full-field WDM errors are 0.0090 and 0.0067, compared with 0.0038 for BALANI-NO.

5 Conclusion

We introduced BALANI-NO, a spectral neural operator that combines an isotropic, grid-independent pole branch with a small low-mode table. We evaluated it against ten baselines under matched parameter budgets on Darcy flow, Navier–Stokes, and gravitational lensing using seven seeds per configuration.

BALANI-NO achieves the lowest zero-shot error on Darcy (0.1194 versus 0.1300 for FNO) and on all zero-shot lensing splits for both the full-field and subhalo variants. It also achieves the lowest error at both resolutions on Navier–Stokes. Its in-distribution accuracy is competitive but not consistently the lowest. It ranks third on Darcy behind CNN and PDNO and ties with U-FNO on WDM for full-field lensing. CNN, PDNO, and the pooled spectral models generally show larger errors after a resolution change. The ablations show that the two BALANI-NO branches provide complementary information. Their combination gives substantial gains on Navier–Stokes and lensing, while Darcy remains dominated by the pole-only variants.

Across the four main and appendix tasks, three factors can give flexible grid-tied models an advantage at the training resolution. In Darcy, the solution operator depends on a spatially varying coefficient. In CAMELS, the network receives $\log_{10}\rho$ while the physical operator acts linearly on ρ. In EMAG2, the inputs are real measurements with heterogeneous and locally irregular structure. These properties give CNN and PDNO more flexibility to fit the training grid. They also lead to larger degradation when the resolution changes.

These factors are less important for lensing and Navier–Stokes. Lensing has a fixed synthetic input distribution and an exactly rational, sample-independent operator that is given directly in terms of κ. Navier–Stokes also has a fixed generative process, and the tested fields contain very little high-frequency energy. BALANI-NO therefore matches or exceeds the flexible baselines at the training resolution on these tasks while retaining strong resolution transfer. Across all four tasks, BALANI-NO has the lowest zero-shot error after a resolution change. The results support the pole branch as a transfer-oriented inductive bias. Its restricted parameterization reduces the flexibility available at a single grid while limiting dependence on the discretization.

6 Limitations.

BALANI-NO has no exploitable global symbol when the operator itself varies with the input, as on Darcy, which is why less constrained baselines can still win at the training resolution there. The additional CAMELS and EMAG2 tasks (Appendices E–F) are preliminary, and for CAMELS a single fixed-cosmology pilot dataset.

References

Morad Aaboud et al. Constraints on mediator-based dark matter and scalar dark energy models using $\sqrt{s} = 13$ TeV pp collision data collected by the ATLAS detector. Journal of High Energy Physics, 05:142, 2019.

D. S. Akerib et al. Results from a search for dark matter in the complete LUX exposure. Physical Review Letters, 118(2):021303, 2017.

Stephon Alexander, Sergei Gleyzer, Evan McDonough, Michael W. Toomey, and Emanuele Usai. Deep learning the morphology of dark matter substructure. The Astrophysical Journal, 893(1):15, 2020a.

Stephon Alexander, Sergei Gleyzer, Hanna Parul, Pranath Reddy, Michael W. Toomey, Emanuele Usai, and Ryker Von Klar. Decoding dark matter substructure without supervision. arXiv preprint arXiv:2008.12731, 2020b.

E. Aprile et al. Dark matter search results from a one ton-year exposure of XENON1T. Physical Review Letters, 121(11):111302, 2018.

Simon Birrer and Adam Amara. lenstronomy: Multi-purpose gravitational lens modelling software package. Physics of the Dark Universe, 22:189–201, 2018.

Simon Birrer, Anowar J. Shajib, Daniel Gilman, Aymeric Galan, Jelle Aalbers, Martin Millon, Robert Morgan, Giulia Pagano, Ji Won Park, Luca Teodori, Nicolas Tessore, Madison Ueland, Lyne Van de Vyvere, Sebastian Wagner-Carena, Ewoud Wempe, Lilan Yang, Xuheng Ding, Thomas Schmidt, Dominique Sluse, Ming Zhang, and Adam Amara. lenstronomy II: A gravitational lensing software ecosystem. Journal of Open Source Software, 6(62):3283, 2021.

Richard J. Blakely. Potential Theory in Gravity and Magnetic Applications. Cambridge University Press, 1996.

Paul Bode, Jeremiah P. Ostriker, and Neil Turok. Halo formation in warm dark matter models. The Astrophysical Journal, 556(1):93, 2001.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. Nature Machine Intelligence, 6(6):631–640, 2024. doi: 10.1038/s42256-024-00844-4.

Geo Jolly Cheeramvelil, Sergei Gleyzer, and Michael W. Toomey. Equivariant neural networks for signatures of dark matter morphology in strong lensing data. In Machine Learning and the Physical Sciences Workshop, NeurIPS 2023, 2023.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Transactions on Neural Networks, 6(4):911–917, 1995.

Xiangyi Cui et al. Dark matter results from 54-ton-day exposure of PandaX-II experiment. Physical Review Letters, 119(18):181302, 2017.

Yashwardhan A. Deshmukh, Sergei Gleyzer, Kartik Sachdev, and Michael W. Toomey. Learning dark matter representation from strong lensing images through self-supervision. In Machine Learning and the Physical Sciences Workshop, NeurIPS 2023, 2023.

Ana Diaz Rivero and Cora Dvorkin. Direct detection of dark matter substructure in strong lens images with convolutional neural networks. Physical Review D, 101(2):023515, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

A. K. Drukier, Katherine Freese, and D. N. Spergel. Detecting cold dark matter candidates. Physical Review D, 33:3495–3508, 1986.

W. Gao, J. Luo, R. Xu, and Y. Liu. Dynamic Schwartz–Fourier neural operator for enhanced expressive power. Transactions on Machine Learning Research, 2025.

Mark W. Goodman and Edward Witten. Detectability of certain dark matter candidates. Physical Review D, 31:3059, 1985.

David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In International Conference on Learning Representations (ICLR), 2017.

Zhipeng Huang, Zhizheng Zhang, Cuiling Lan, Zheng-Jun Zha, Yan Lu, and Baining Guo. Adaptive frequency filters as efficient global token mixers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6049–6059, 2023.

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.

Jean Kossaifi, Nikola Kovachki, Zongyi Li, David Pitt, Miguel Liu-Schiaffini, Robert Joseph George, Boris Bonev, Kamyar Azizzadenesheli, Julius Berner, Valentin Duruisseaux, and Anima Anandkumar. A library for learning neural operators. Journal of Machine Learning Research, 27(205):1–6, 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.

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

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

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.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. Computer Methods in Applied Mechanics and Engineering, 393:114778, 2022.

Hareth S. Mahdi, Pascal J. Elahi, Geraint F. Lewis, and Chris Power. Matter in the beam: Weak lensing, substructures and the temperature of dark matter. The Astrophysical Journal, 826(2):212, 2016.

Brian Meyer, Arnaud Chulliat, and Richard Saltus. Derivation and error analysis of the earth magnetic anomaly grid at 2 arc min resolution version 3 (EMAG2v3). Geochemistry, Geophysics, Geosystems, 18:4522–4537, 2017.

Ashutosh Ojha, Sergei Gleyzer, Michael W. Toomey, and Pranath Reddy Kumbam. LensPINN: Physics informed neural network for learning dark matter morphology in lensing. In Machine Learning and the Physical Sciences Workshop, NeurIPS 2024, 2024.

Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. U-NO: U-shaped neural operators. Transactions on Machine Learning Research, 2023.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. Advances in Neural Information Processing Systems (NeurIPS), 36:77187–77200, 2023.

Jin Young Shin, Jae Yong Lee, and Hyung Ju Hwang. Pseudo-differential neural operator: Generalized Fourier neural operator for learning solution operators of partial differential equations. arXiv preprint arXiv:2201.11967, 2022.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. Advances in Neural Information Processing Systems (NeurIPS), 35:1596–1611, 2022.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In Advances in Neural Information Processing Systems (NeurIPS), volume 33, pp. 7537–7547, 2020.

Sreedevi Varma, Malcolm Fairbairn, and Julio Figueroa. Dark matter subhalos, strong lensing and machine learning. arXiv preprint arXiv:2005.05353, 2020.

Kyriakos Vattis, Michael W. Toomey, and Savvas M. Koushiappas. Deep learning the astrometric signature of dark matter substructure. Physical Review D, 104:123541, 2021.

Lucas J. Veloso, Michael W. Toomey, and Sergei Gleyzer. Lensformer: A physics-informed vision transformer for gravitational lensing. In Machine Learning and the Physical Sciences Workshop, NeurIPS 2023, 2023.

Francisco Villaescusa-Navarro, Daniel Anglés-Alcázar, Shy Genel, David N. Spergel, Rachel S. Somerville, Romeel Dave, Annalisa Pillepich, Lars Hernquist, Dylan Nelson, Paul Torrey, Desika Narayanan, Yin Li, Oliver Philcox, Valentina La Torre, Ana Maria Delgado, Shirley Ho, Sultan Hassan, Blakesley Burkhart, Digvijay Wadekar, Nicholas Battaglia, Gabriella Contardo, and Greg L. Bryan. The CAMELS project: Cosmology and astrophysics with MachinE learning simulations. The Astrophysical Journal, 915(1): 71, 2021.

Francisco Villaescusa-Navarro, Shy Genel, Daniel Anglés-Alcázar, Leander Thiele, Romeel Dave, Desika Narayanan, Andrina Nicola, Yin Li, Pablo Villanueva-Domingo, Benjamin Wandelt, David N. Spergel, Rachel S. Somerville, Jose Manuel Zorrilla Matilla, Faizan G. Mohammad, Sultan Hassan, Helen Shao, Digvijay Wadekar, Michael Eickenberg, Kaze W. K. Wong, Gabriella Contardo, Yongseok Jo, Emily Moser, Erwin T. Lau, Luis Fernando Machado Poletti Valle, Lucia A. Perez, Daisuke Nagai, Nicholas Battaglia, and Mark Vogelsberger. The CAMELS multifield dataset: Learning the universe's fundamental parameters with artificial intelligence. The Astrophysical Journal Supplement Series, 259(2):61, 2022.

Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. U-FNO—an enhanced Fourier neural operator-based deep-learning model for multiphase flow. Advances in Water Resources, 163:104180, 2022.

Zecheng Zhang, Wing Tat Leung, and Hayden Schaeffer. BelNet: Basis enhanced learning, a mesh-free neural operator. arXiv preprint arXiv:2212.07336, 2022.

List of Appendices

.1 Pole recovery
A Additional Qualitative Results
B Mathematical Description of architectures
C Architecture Diagrams
D Hyperparameters and implementation details
E Additional Task: Gravitational Potential Recovery on Cosmological Density Maps
F Additional Task: Magnetic Field Upward Continuation on Real-World Geophysical Data
G Architecture configurations for the additional tasks
H Statement on the Use of Large Language Models

.1 Pole recovery

We test whether the learned pole parameters s_j in Eq. equation 4 recover known scales when the underlying operator has a screened-Poisson form. We generate data from $\hat{u}(k) = \sum_j c_j\, \hat{a}(k)/(|k|^2 + s_j^2)$ using white-noise inputs a on a 64×64 grid with 200 training and 50 test samples. We fit a bare pole layer with a degree-0 numerator and no table, skip connection, or nonlinearity, so the model class contains the ground-truth operator exactly. For fixed pole locations, the amplitudes c_j enter linearly. We therefore search over pole locations on a dense grid and solve for the amplitudes in closed form at each candidate. For $J = 2$, the search is performed jointly over both pole locations, and the parameters with the lowest training error are selected.

Table 8: Pole recovery from synthetic Green's-function data with one pole (J=1, $c = 1$, top) and two poles (J=2, amplitudes $(1, 1)$, $(1, 0.5)$ and $(0.8, 1.2)$ for the three cases in order, bottom), fit by grid search over s_j with c_j solved in closed form. Error is $|s_{\text{learned}} - s_{\text{true}}|/s_{\text{true}}$.

s_{true}	s_{learned}	Error (%)	Test rel. L_2
0.5	0.5000	0.00	$<10^{-6}$
1	1.0000	0.00	10^{-6}
2	2.0000	0.00	$<10^{-6}$
4	4.0000	0.00	10^{-6}
8	8.0000	0.00	$<10^{-6}$
16	16.0006	0.00	1.7×10^{-5}
24	24.0260	0.11	4.4×10^{-4}
(2, 10)	(1.998, 9.936)	(0.09, 0.64)	2.6×10^{-4}
(1, 8)	(0.998, 7.474)	(0.21, 6.57)	1.5×10^{-3}
(4, 20)	(3.983, 19.485)	(0.42, 2.57)	1.1×10^{-3}

Single-pole recovery is nearly exact across the tested range (Table 8, top), with errors below 0.11% for all $s_{\text{true}} \in [0.5, 24]$ and consistently low test errors. For two poles (Table 8, bottom), the smaller pole is recovered within 0.42% in all cases, while the larger pole has errors of 0.64–6.57%. Despite this variation, the relative L_2 test error remains below 1.5×10^{-3}. The larger uncertainty in the second pole reflects the weaker identifiability of overlapping pole contributions once the first pole captures the dominant low-frequency behavior.

A Additional Qualitative Results

This appendix shows the predictions of all eleven architectures on one fixed test sample per setting, complementing the quantitative results of Section 3. The ground truth is shown first in every figure, and on the lensing pages it is repeated at the top for reference. The images are for inspection only; the tables report the quantitative comparison, and small differences between the most accurate models are not visible by eye.

Darcy flow (Figure 3). Each cell shows the prediction at the training resolution (16×16, top) and the zero-shot prediction at 32×32 (bottom). A resolution-consistent model should produce the same field at both resolutions, only sampled more finely. This is the case for BALANI-NO, FNO, ALNO and LNO. The two branch–trunk models produce visibly smoother fields, in line with their higher in-distribution error. CNN and PDNO, which have the largest degradation ratios in Table 1 ($8.61\times$ and $4.79\times$), show the largest differences between their two rows.

Navier–Stokes (Figure 4). Each cell shows the prediction at the training resolution (32×32, top) and at the zero-shot resolution (64×64, bottom). Because the 64×64 target holds essentially no energy beyond what the 32×32 grid can resolve (Section 3.2), a resolution-consistent model should produce the same field in both rows, sampled more finely. PDNO and CNN, whose input representation or kernels are tied to the training grid, and the pooled models UNO and U-FNO, and to a lesser degree HC-UNO degrade. BALANI-NO, FNO, ALNO and LNO look the same in both rows. DeepONet and POD-DeepONet also look the same across rows, but their fields are smooth and low-contrast, consistent with errors above 0.5 at both resolutions.

Gravitational lensing (Figures 5–10). Each row is one model and the columns are the WDM (in-distribution), CDM (zero-shot) and Axion (zero-shot) samples. Figures 5–7 show the full-field variant ($\kappa \to \boldsymbol{\alpha}$) and Figures 8–10 the subhalo variant ($\kappa_{\mathrm{sub}} \to \boldsymbol{\alpha}_{\mathrm{sub}}$), each split over three pages with the ground truth repeated at the top. All models are shown on the same three input samples. Since the lensing errors of the leading models are around one percent, visible failures are confined to the less accurate architectures, and the zero-shot columns are where the ranking by error in Tables 3 and 4 is most likely to show.

All spectral baselines use the block defined in Eq. equation 2 together with the lift and projection layers described in Section 2. They differ in the parameterization of the symbol $W(k)$ and in the block arrangement used by the U-shaped models. CNN, DeepONet, and POD-DeepONet use separate architectures. Frequencies are represented by integer Fourier indices throughout.

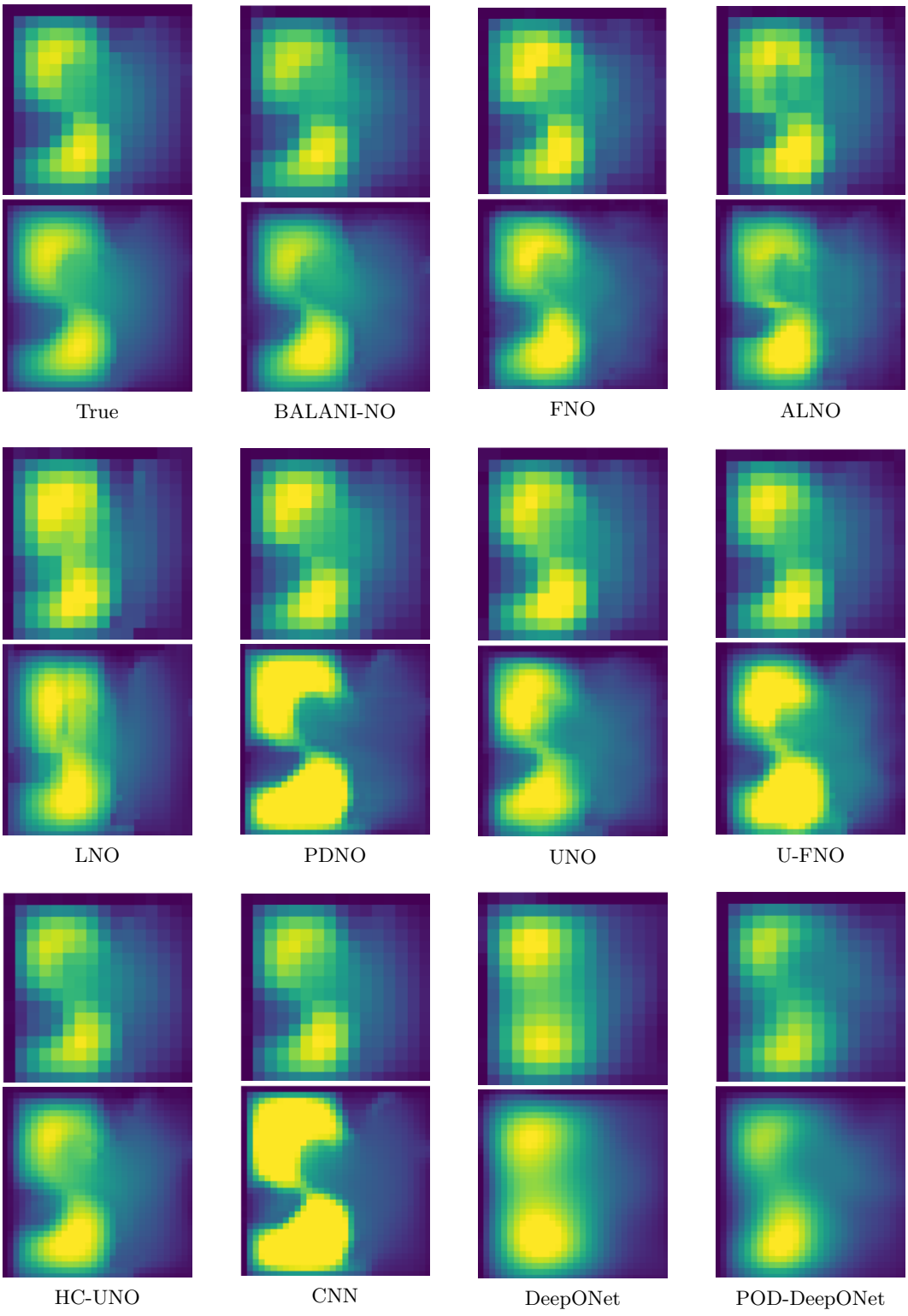

Figure 3: Qualitative comparison of Darcy flow predictions across all eleven architectures. Within each cell, the top image is the prediction at the training resolution (16×16) and the bottom image is the zero-shot prediction at 32×32; the ground truth ("True") is shown first.

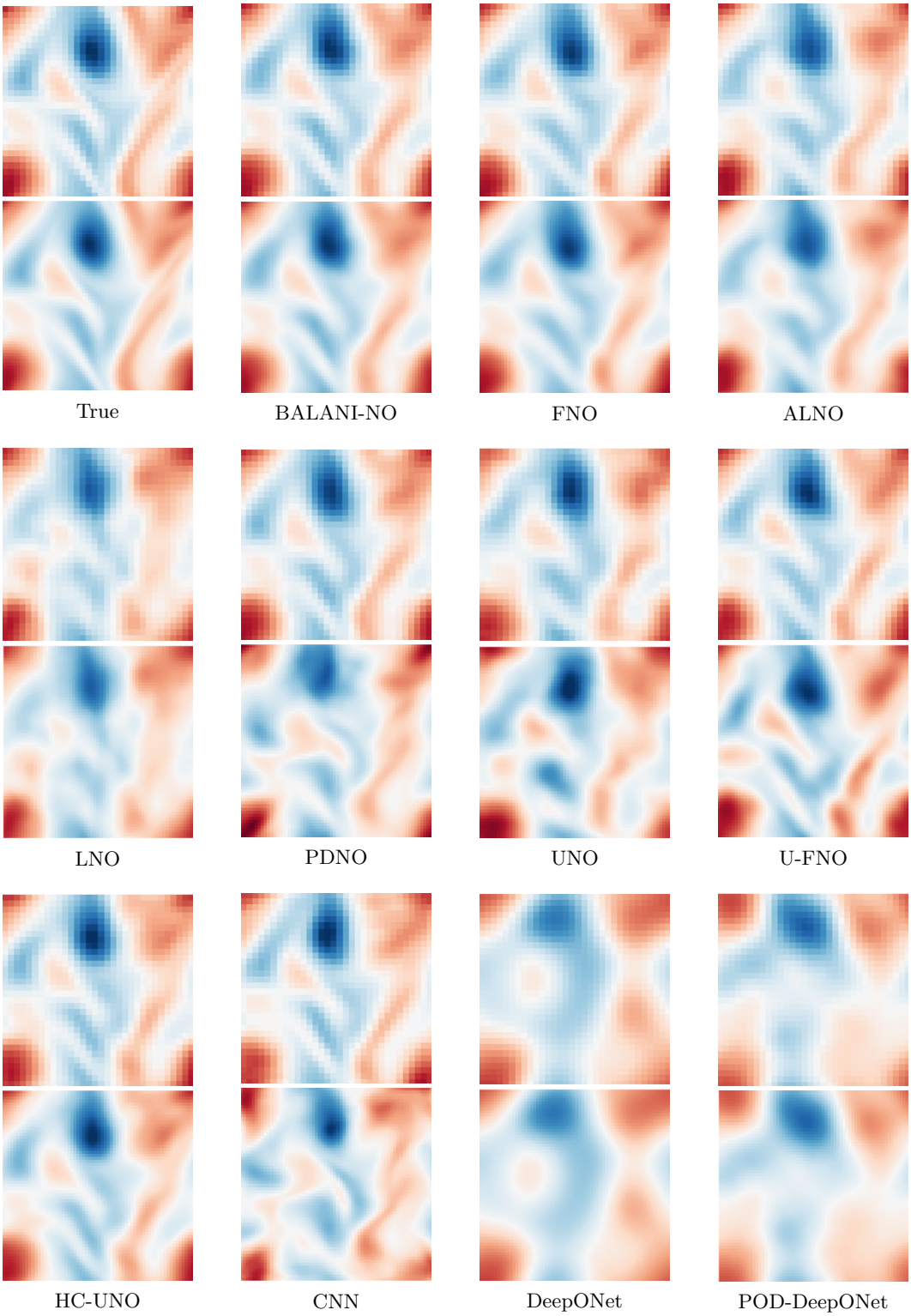

Figure 4: Qualitative comparison of Navier–Stokes vorticity predictions across all eleven architectures. Within each cell, the top image is the prediction at the training resolution (32×32) and the bottom image is the zero-shot prediction at 64×64; the ground-truth field ("True") is shown first.

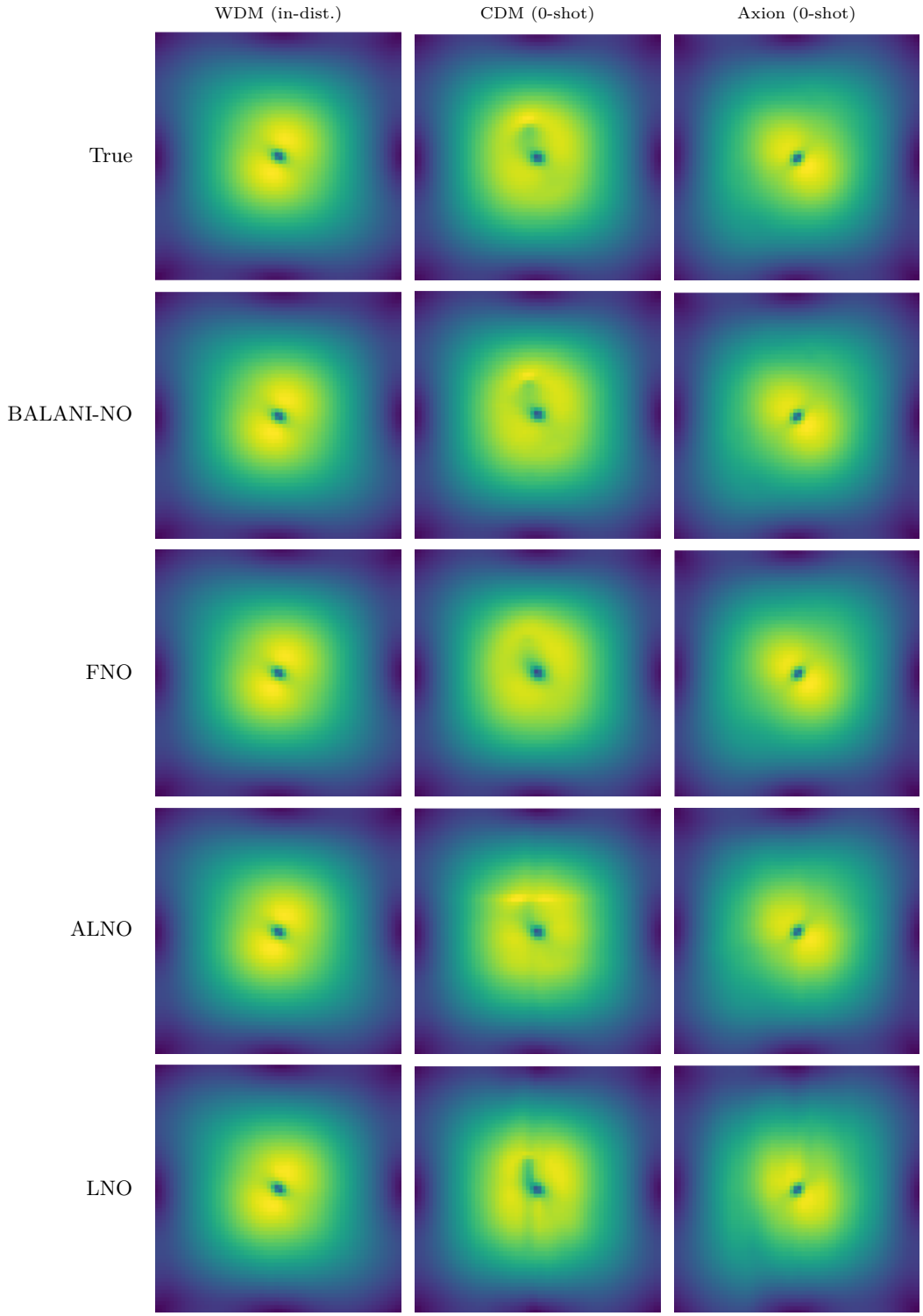

Figure 5: Full-field lensing predictions ($\kappa \to \boldsymbol{\alpha}$), part 1 of 3: BALANI-NO, FNO, ALNO and LNO. Rows are models; columns are the WDM (in-distribution), CDM and Axion (zero-shot) samples. The ground truth ("True") is repeated as the top row of each part.

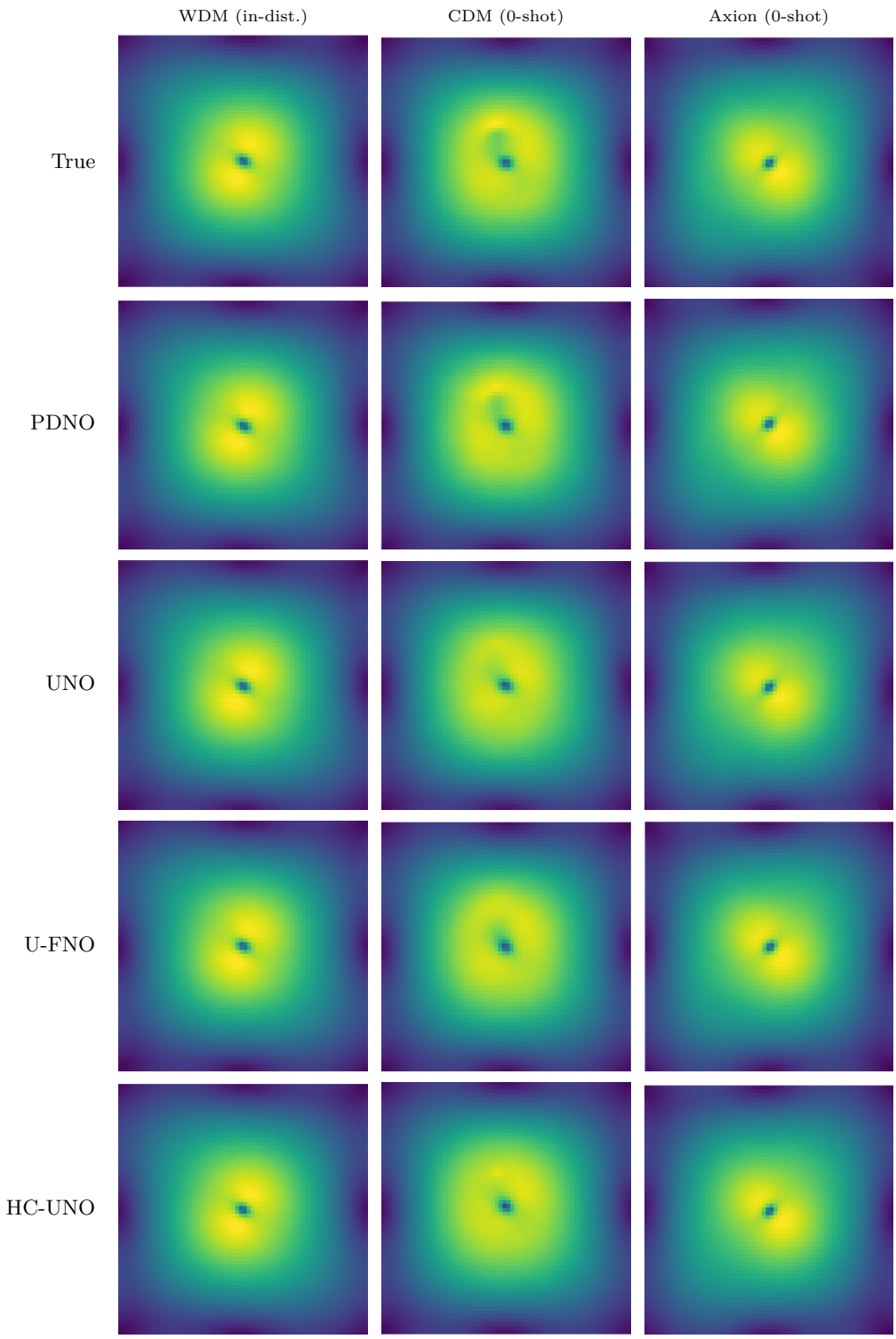

Figure 6: Full-field lensing predictions, part 2 of 3: PDNO, UNO, U-FNO and HC-UNO.

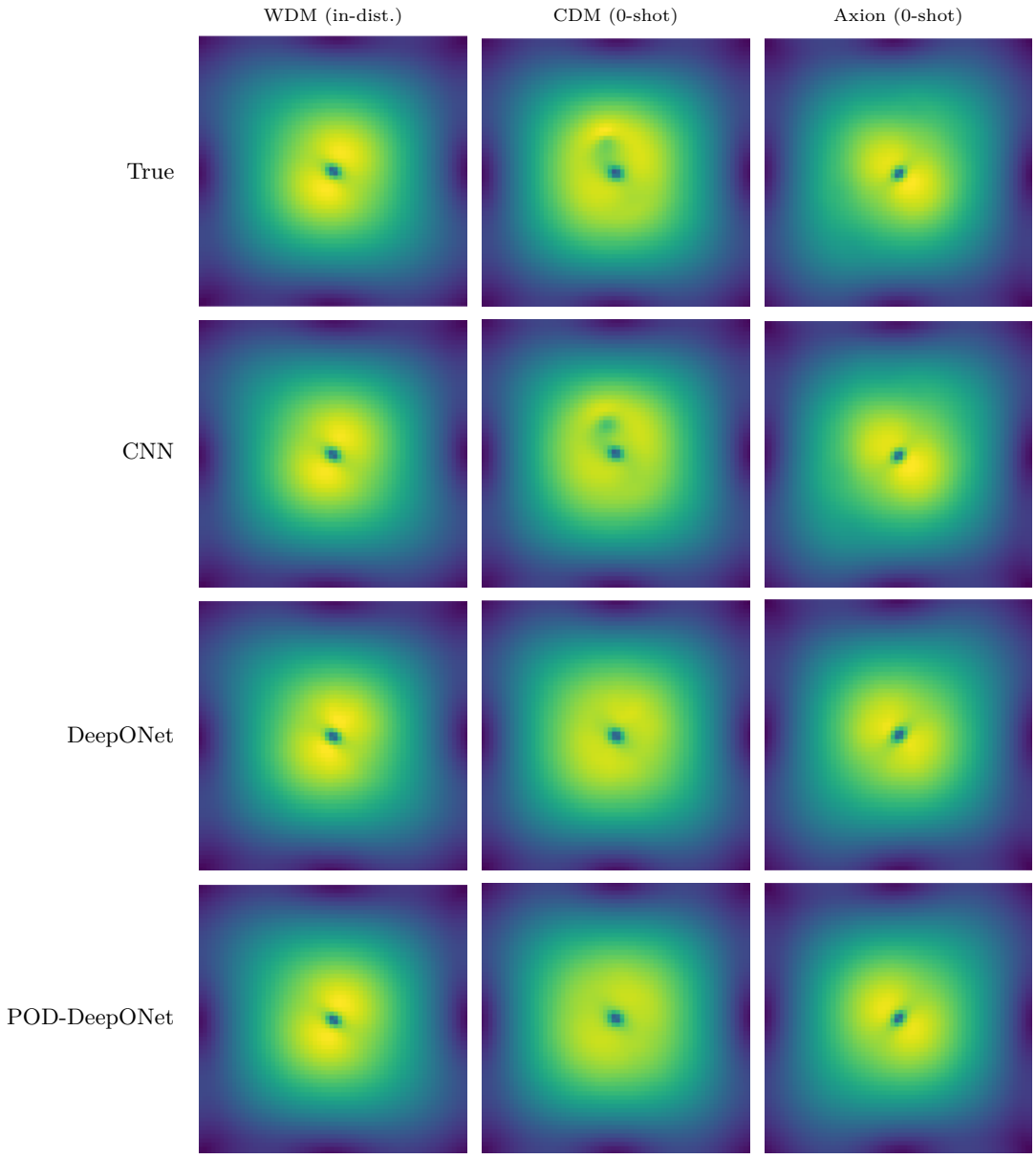

Figure 7: Full-field lensing predictions, part 3 of 3: CNN, DeepONet and POD-DeepONet.

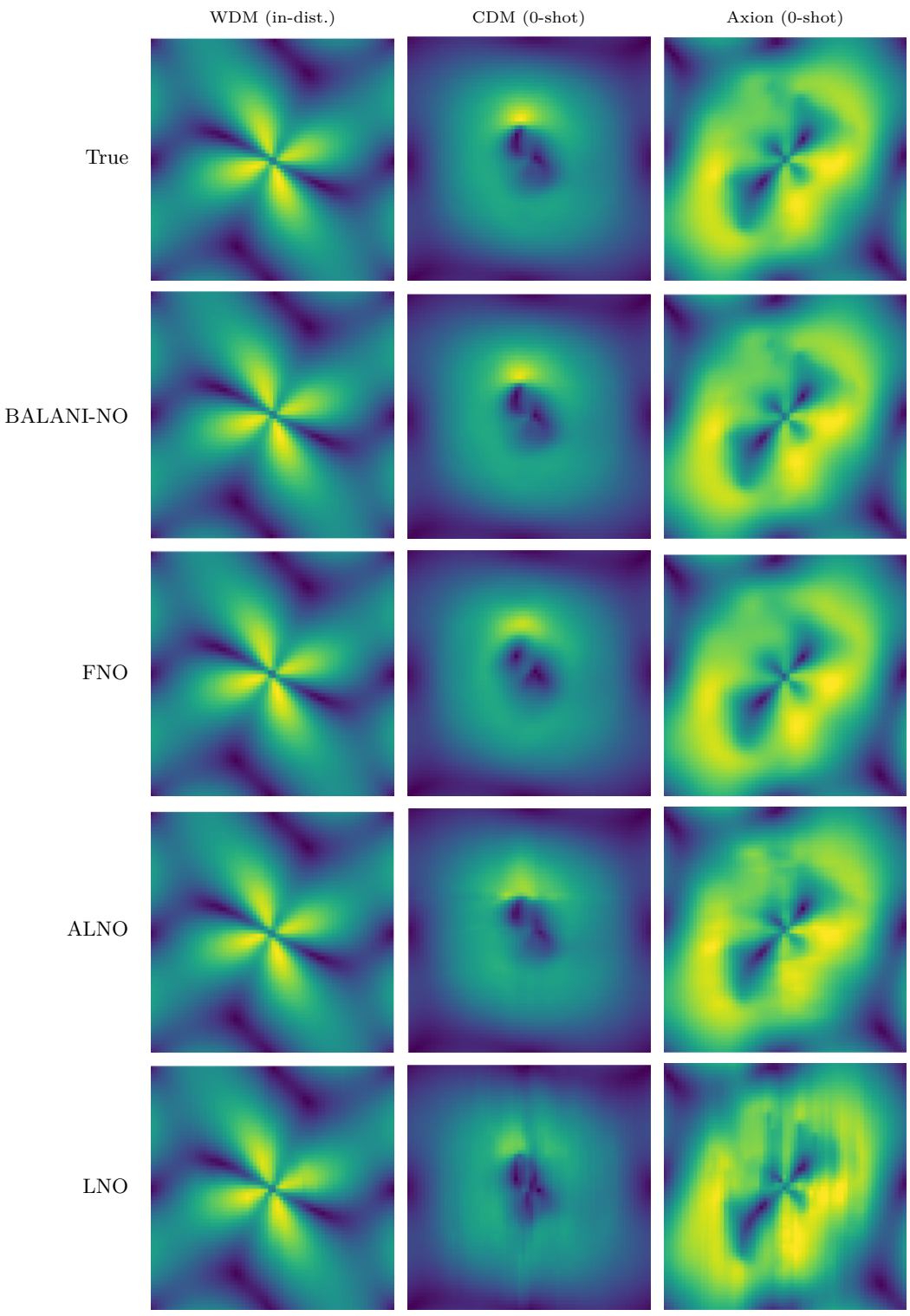

Figure 8: Subhalo lensing predictions ($\kappa_{\text{sub}} \to \boldsymbol{\alpha}_{\text{sub}}$), part 1 of 3: BALANI-NO, FNO, ALNO and LNO; same layout as Figure 5.

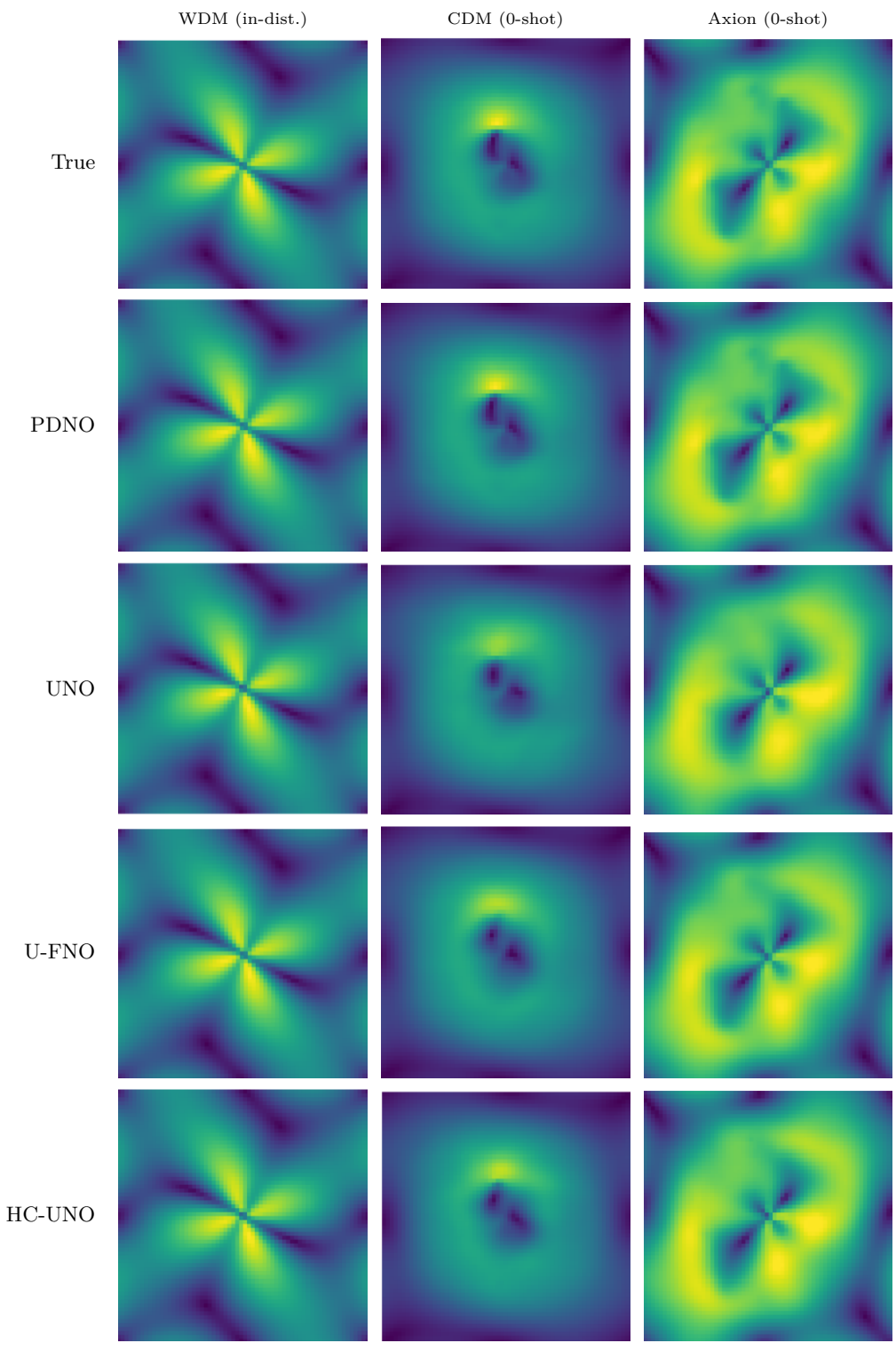

Figure 9: Subhalo lensing predictions, part 2 of 3: PDNO, UNO, U-FNO and HC-UNO.

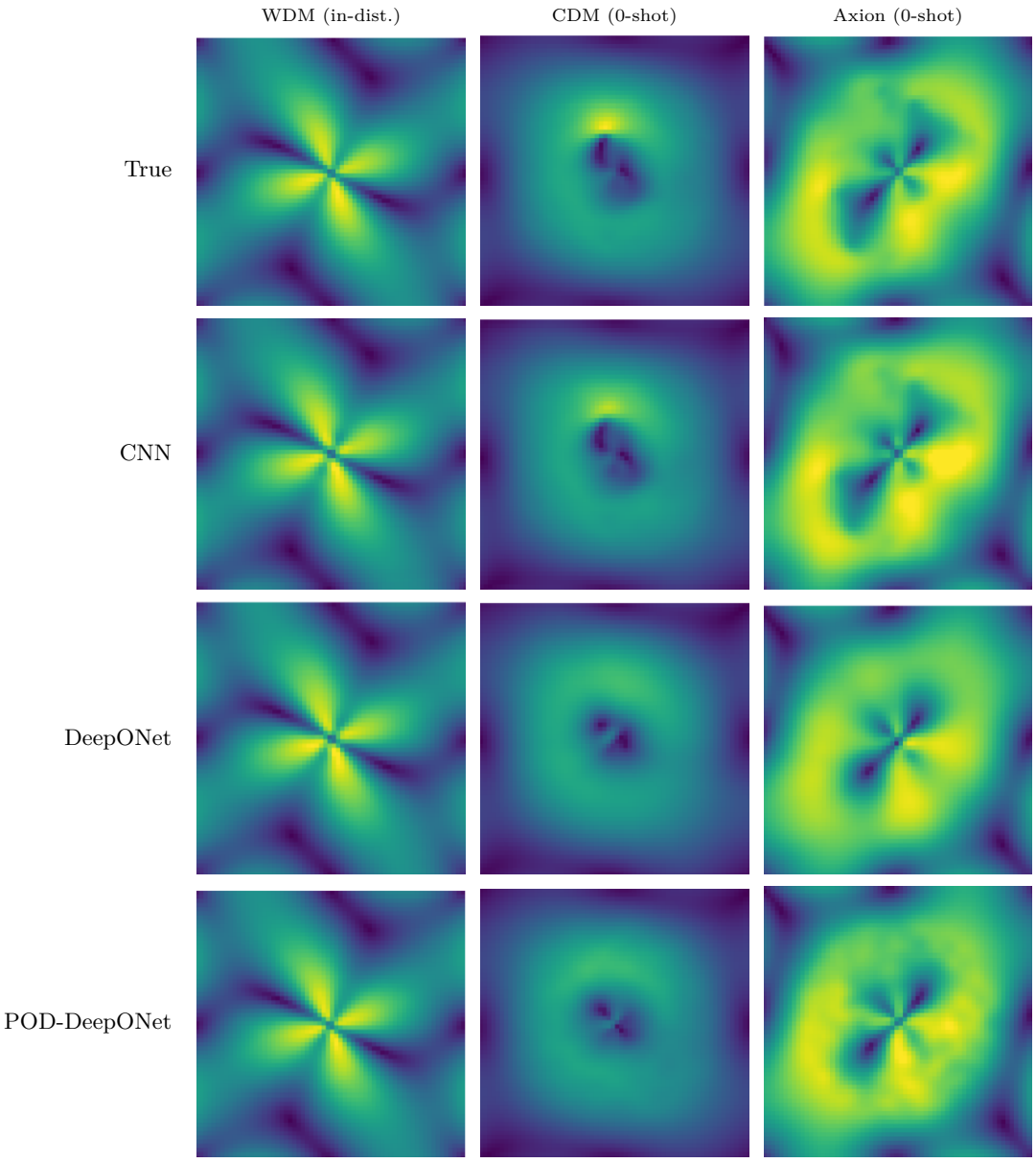

Figure 10: Subhalo lensing predictions, part 3 of 3: CNN, DeepONet and POD-DeepONet.

B Mathematical Description of architectures

B.1 FNO

FNO (Li et al., 2021) uses a lookup table, learned independently per retained mode and zero elsewhere:

$$W_{\mathrm{FNO}}(k) = \begin{cases} R_l(k_x, k_y), & (k_x, k_y) \in \mathcal{K}_{m_1,m_2}, \\ 0, & \text{otherwise}, \end{cases} \qquad R_l(k_x, k_y) \in \mathbb{C}^{C\times C}, \tag{6}$$

where $\mathcal{K}_{m_1,m_2} = \{0 \le k_x < m_2,\ -m_1 \le k_y < m_1\}$, with independent weight blocks for the $k_y \ge 0$ and $k_y < 0$ halves. The table branch of BALANI-NO is the same construct with $m_1 = m_2 = m$.

B.2 ALNO and LNO

Both place poles along a single axis (k_y), treated as a formal Laplace variable $s = i\omega_y$ with $\omega_y = 2\pi k_y$:

$$K(i\omega_y) = \sum_{n=1}^{N} \frac{\beta_n}{i\omega_y - \mu_n}, \qquad \beta_n, \mu_n \in \mathbb{C}^{C\times C}, \tag{7}$$

where the fraction is taken entrywise, so every channel pair has its own poles. The real parts of μ_n are initialized in $[-2, -1)$ and the imaginary parts linearly spaced in $[-3, 3]$; they are not constrained afterwards. LNO (Cao et al., 2024) uses this term alone at every k_x:

$$W_{\mathrm{LNO}}(k_x, k_y) = K(i\omega_y). \tag{8}$$

ALNO adds an FNO-style multiplier $H(k_x) \in \mathbb{C}^{C\times C}$ over the untransformed axis, truncated to $k_x < M$ and applied at every k_y:

$$W_{\mathrm{ALNO}}(k_x, k_y) = K(i\omega_y) + H(k_x)\mathbb{1}\{k_x < M\}. \tag{9}$$

For static, isotropic fields, assigning k_y a privileged "temporal" role has no physical basis.

B.3 PDNO

PDNO (Shin et al., 2022) replaces any structural prior on $W(k)$ by an MLP acting on a Fourier-feature encoding of the frequency coordinate:

$$W_{\mathrm{PDNO}}(k) = \mathrm{MLP}_\theta\big(\gamma(\tilde{k})\big) \in \mathbb{C}^{C\times C}, \quad \gamma(\tilde{k}) = \big[\tilde{k},\ \sin(2^i\pi\tilde{k}),\ \cos(2^i\pi\tilde{k})\big]_{i=0}^{n_{freq}-1}, \quad \tilde{k} = \frac{k}{k_{\mathrm{nyq}}}, \tag{10}$$

with $k_{\mathrm{nyq}} = \max(H, W)/2$ the Nyquist index of the current grid, so the feature vector has $2(1 + 2n_{freq})$ channels. The MLP is implemented as a stack of 1×1 convolutions over the frequency grid (GELU activations) whose head outputs the real and imaginary parts of the $C \times C$ symbol. It is evaluated at every retained frequency, with no truncation, table or pole structure. Because $\tilde{k}$ is normalized by the grid's own Nyquist index, the same network input corresponds to different physical frequencies at different resolutions.

B.4 U-shaped models: UNO, U-FNO and HC-UNO

These share an encoder–decoder recursion with average pooling $\mathcal{P}$ (factor 2) and bilinear upsampling $\mathcal{U}$. With h^0 the lifted input and L levels,

$$s^l = \mathcal{B}^l(h^{l-1}; W^l), \quad h^l = \mathcal{P}(s^l) \quad (l = 1, \dots, L), \qquad h^{l-1}_{dec} = \mathcal{B}'^l\big(\mathcal{U}(h^l_{dec}) + s^l;\, W'^l\big) \quad (l = L, \dots, 1), \tag{11}$$

with $h^L_{dec} = h^L$ and the projection head applied to h^0_{dec}. Encoder and decoder blocks have separate symbols. The number of levels is $\min(L, \lfloor \log_2 \min(H, W) \rfloor)$, which equals $L = 3$ at every resolution used here.

U-FNO. (Wen et al., 2022) Every block uses the FNO symbol of Eq. equation 6 with its own weights.

UNO. (Rahman et al., 2023) Each block's symbol is a filter bank of N_f real, anisotropic Gaussian windows in absolute frequency, combined with per-filter complex channel-mixing matrices:

$$W_{\mathrm{UNO}}(k) = \sum_{j=1}^{N_f} G_j(k)\, W_j, \qquad G_j(k_x, k_y) = \exp\left[-\left(\frac{u_j}{b_j}\right)^2 - \left(\frac{v_j}{b_j/3}\right)^2\right], \tag{12}$$

where (u_j, v_j) is $(k_x - c_{x,j},\, k_y - c_{y,j})$ rotated by a learned angle θ_j, $b_j = \mathrm{softplus}(\beta_j) + 0.5$, centers are $F_{\max}\tanh(\cdot)$ of a free parameter, and $W_j \in \mathbb{C}^{C\times C}$. The filter geometry $\{c_{x,j}, c_{y,j}, b_j, \theta_j\}$ is a single dataset-level set shared by all blocks and resized bilinearly to each level's frequency grid, while the channel-mixing matrices W_j are separate for every block.

HC-UNO. The filter geometry is instead predicted per sample by a hypernetwork applied once to the model input and reused, after resizing, at every level:

$$q = W_h\, \mathrm{Flatten}\Big[\mathrm{Pool}_1\Big(\sigma\big(\mathrm{Conv}^{(2)}_{3\times3}(\sigma(\mathrm{Conv}^{(1)}_{3\times3}(\mathrm{Pool}_8(a))))\big)\Big)\Big] \in \mathbb{R}^{N_f\times 4}, \tag{13}$$

$$(c_{x,j}, c_{y,j}, b_j, \theta_j) = \big(F_{\max}\tanh q_{x,j},\ F_{\max}\tanh q_{y,j},\ \mathrm{softplus}(q_{b,j}) + 0.5,\ \pi\,\mathrm{sig}(q_{\theta,j})\big), \tag{14}$$

where Pool_n is adaptive average pooling to $n\times n$ and sig is the logistic sigmoid. The symbol is $W_{\mathrm{HC}}(k; a) = \sum_j G_j(k; a)W_j$, with channel-mixing matrices W_j still dataset-level and separate per block, so only the filter geometry is input-conditioned.

B.5 CNN

The CNN baseline uses no spectral component. Each block is purely local,

$$\mathcal{B}_{\mathrm{CNN}}(h) = \sigma\Big(\mathrm{GN}\big(\mathrm{Conv}^{(2)}_{3\times3}(\sigma(\mathrm{GN}(\mathrm{Conv}^{(1)}_{3\times3}(h))))\big)\Big), \tag{15}$$

and is embedded in the same pooling/upsampling recursion of Eq. equation 11, with the same additive skips.

B.6 DeepONet and POD-DeepONet

DeepONet (Lu et al., 2021) represents the operator as an inner product between a branch network acting on the input field and a trunk network acting on the query coordinate:

$$\hat{u}_c(a)(x) = \sum_{p=1}^{P} b_{c,p}(a)\, t_p(x) + \beta_c, \qquad b_\theta(a) \in \mathbb{R}^{C_{out}\times P},\ t_\phi(x) \in \mathbb{R}^P,\ \beta \in \mathbb{R}^{C_{out}}. \tag{16}$$

The trunk is shared across output channels. The branch is an MLP on the flattened input field (bilinearly resampled to a fixed size when the resolution differs, see Appendix D), and the trunk is an MLP on coordinates $x \in [0,1]^2$, so it can be queried on any grid.

POD-DeepONet (Lu et al., 2022) replaces the learned trunk by a fixed, data-derived basis obtained by Proper Orthogonal Decomposition (top-P principal components) of the centered training targets, and learns only the coefficient (branch) network:

$$\hat{u}(a)(x) = \mu(x) + \sum_{p=1}^{P} c_p(a)\, \varphi_p(x), \qquad c_\theta : a \mapsto \mathbb{R}^P. \tag{17}$$

The branch is a three-layer strided convolutional encoder followed by an MLP. Unlike DeepONet's trunk, $\{\varphi_p\}$ and μ are tied to the training grid; at another resolution we resample each mode and the mean bilinearly. On lensing the two deflection components have separate bases and the coefficient head outputs $2P$ values.

C Architecture Diagrams

Each architecture is illustrated by tracing a tensor through the network, with arrows indicating the operations applied at each stage. Stacked rectangles represent multiple channels. The diagrams are schematic. Fourier coefficients are shown with k=0 at the centre for readability, although the actual FFT layout places low frequencies at the corners. Tensor shapes correspond to the Darcy configuration with a 16×16 grid described in Appendix D. Figure 11 shows the common structure of the five non-hierarchical spectral models: BALANI-NO, FNO, ALNO, LNO, and PDNO. These models differ only in the symbol $W(k)$ used within the spectral block, with the individual blocks shown in Figures 12–16. Figure 17 shows the U-shaped hierarchy used by UNO, U-FNO, HC-UNO, and CNN. Their corresponding blocks are shown in Figures 18–20; U-FNO uses the FNO block shown in Figure 13. Figures 21 and 22 show the branch–trunk architectures of DeepONet and POD-DeepONet, which do not use spectral blocks.

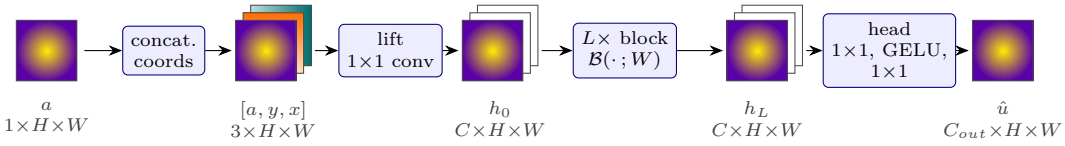

Figure 11: Skeleton shared by BALANI-NO, FNO, ALNO, LNO and PDNO. Normalised pixel coordinates are appended to the input, a pointwise 1×1 convolution lifts the result to C channels, L blocks (Eq. equation 2) keep the shape unchanged, and a pointwise head maps to C_{out} channels. Only the symbol $W(k)$ inside each block differs between the five models (Figures 12–16). None of the layers depends on H or W, so the same weights apply at any resolution.

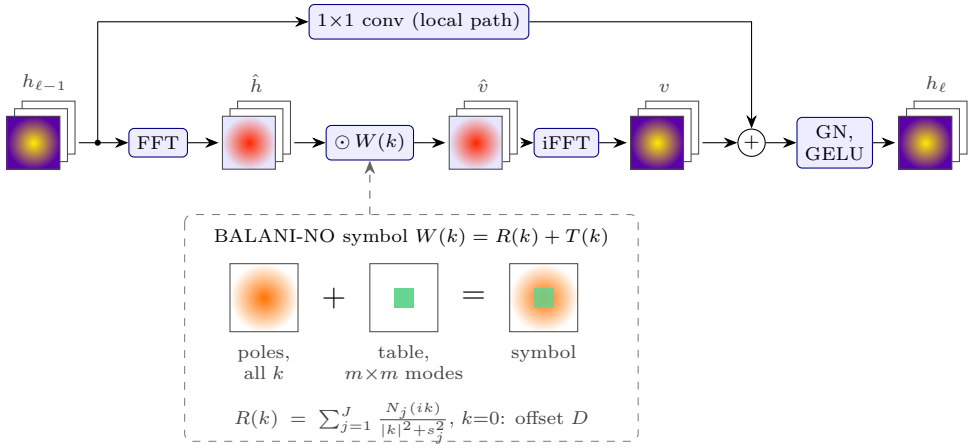

Figure 12: One BALANI-NO block. The symbol adds a rational pole term, defined at every frequency and radial in $|k|$, to a small table on the lowest modes.

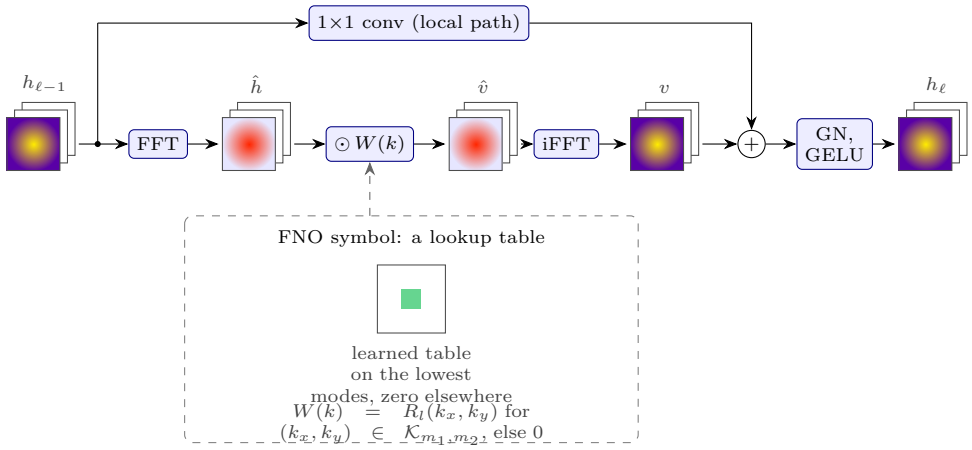

Figure 13: One FNO block. The symbol is a table with one learned $C \times C$ matrix per retained mode and nothing beyond the cutoff. The U-FNO block is identical.

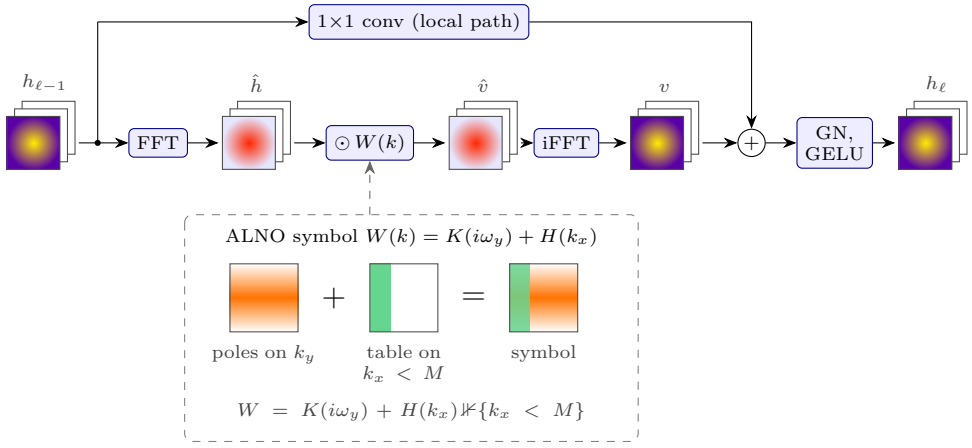

Figure 14: One ALNO block. Poles act along the single axis k_y (constant along k_x), and a truncated table acts along k_x (constant along k_y); the two are added.

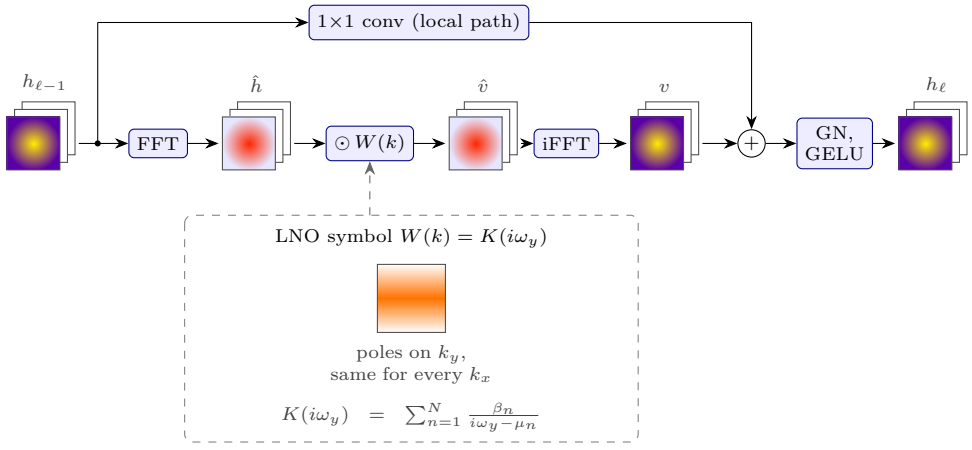

Figure 15: One LNO block. The symbol is a sum of complex poles and residues, per channel pair, along k_y only. There is no table and no dependence on k_x.

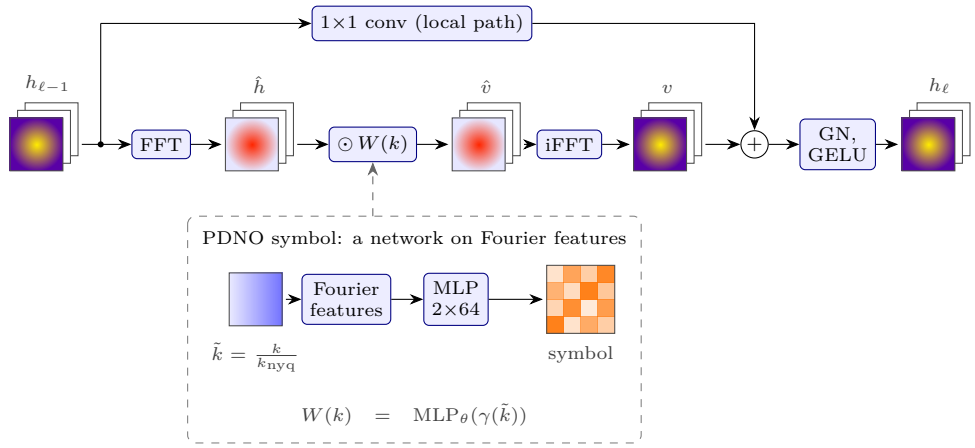

Figure 16: One PDNO block. The symbol is the output of a small network evaluated at every frequency, with no truncation, table or pole structure. The features are $\gamma(\tilde{k}) = [\tilde{k}, \sin(2^i\pi\tilde{k}), \cos(2^i\pi\tilde{k})]_i$, and the frequency is normalised by the Nyquist index of the current grid.

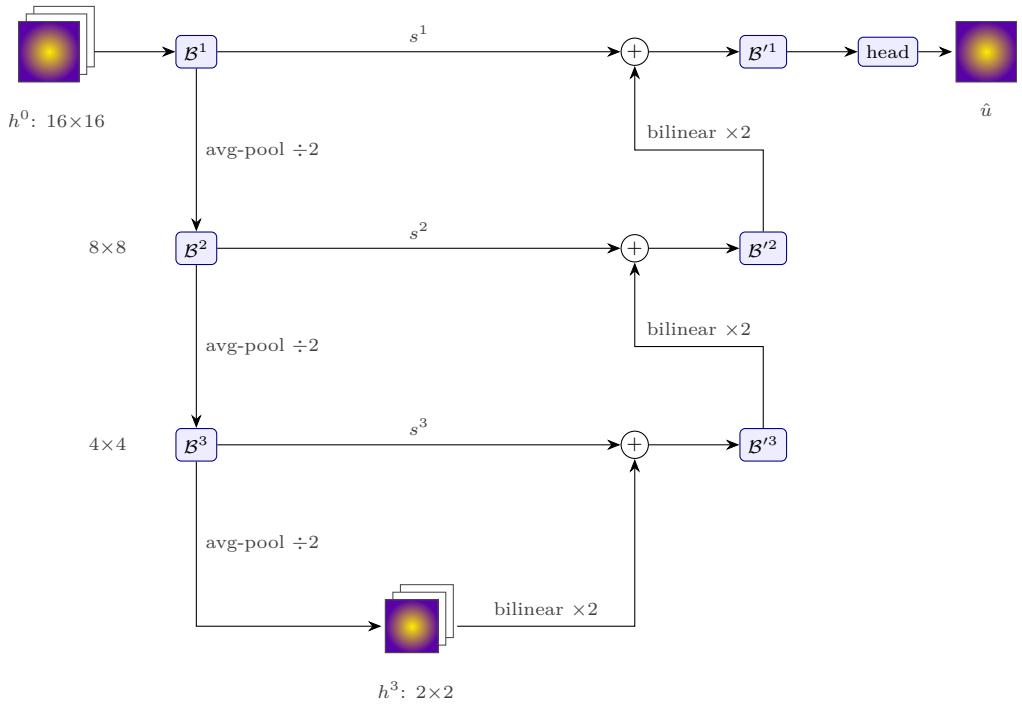

Figure 17: U-shaped hierarchy of UNO, U-FNO, HC-UNO and the CNN (L=3 levels, 16×16 input). The encoder applies a block $\mathcal{B}^l$ and average-pools; each block output s^l is kept as a skip connection. The decoder starts from h^3, and at each level bilinearly upsamples, adds the skip s^l and applies a block $\mathcal{B}'^l$. Encoder and decoder blocks have separate weights. The blocks are those of Figures 18 (UNO, HC-UNO), 13 (U-FNO) and 20 (CNN).

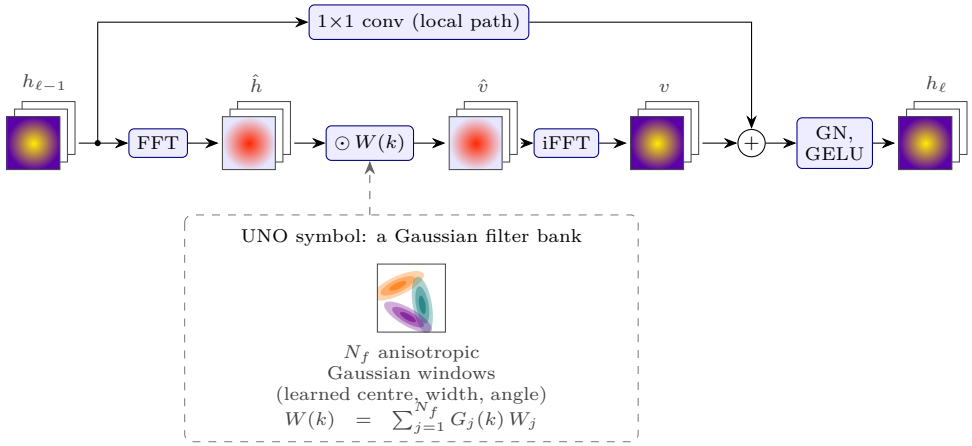

Figure 18: One UNO block (also used by HC-UNO). The symbol is a bank of N_f Gaussian windows in absolute frequency, each with a learned channel-mixing matrix W_j. The window geometry is one dataset-level set shared by all blocks and resized to each level's grid; the matrices W_j are separate for every block.

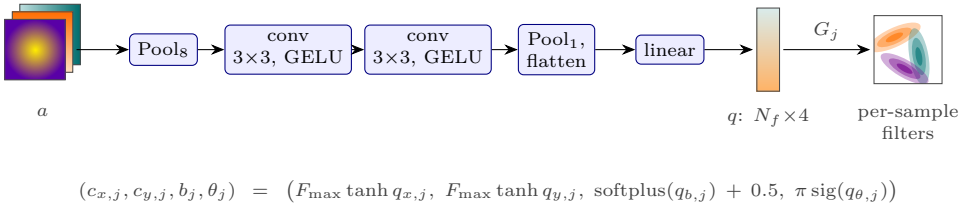

Figure 19: HC-UNO hypernetwork. It is applied once to the raw input a and predicts the geometry of the N_f Gaussian windows of the UNO block (Figure 18) for that sample. The predicted windows are resized to each level's grid and reused in every block, while the channel-mixing matrices W_j stay dataset-level.

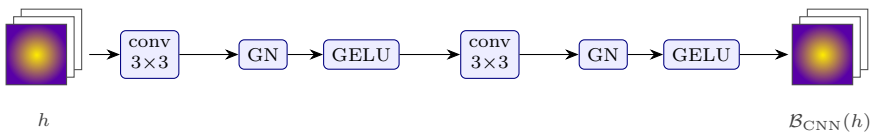

Figure 20: One CNN block: two local 3×3 convolutions, each followed by GroupNorm and GELU. There is no Fourier transform, no symbol and no skip inside the block; the blocks sit in the U-shaped hierarchy of Figure 17.

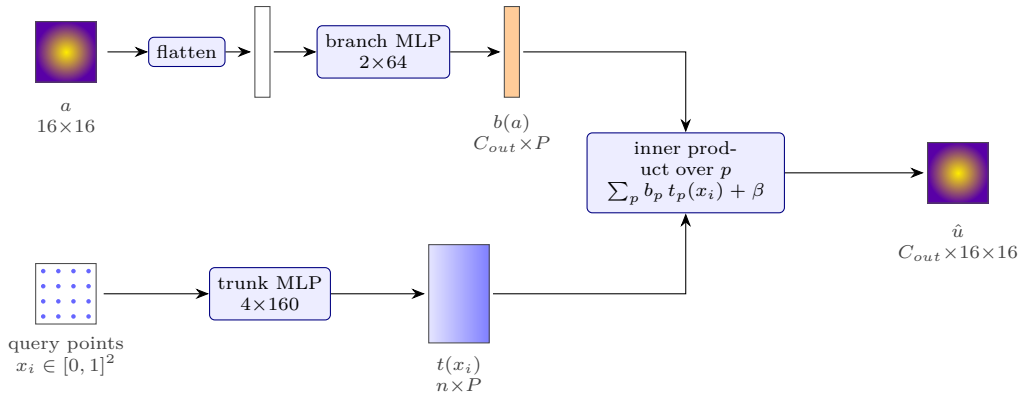

Figure 21: DeepONet. The branch MLP encodes the (resampled to 16×16, flattened) input field into P coefficients per output channel. The trunk MLP encodes each query coordinate into P basis values, shared across channels. The output at every query point is their inner product plus a bias. Because the trunk takes coordinates, the output grid can be changed at test time without changing any weight.

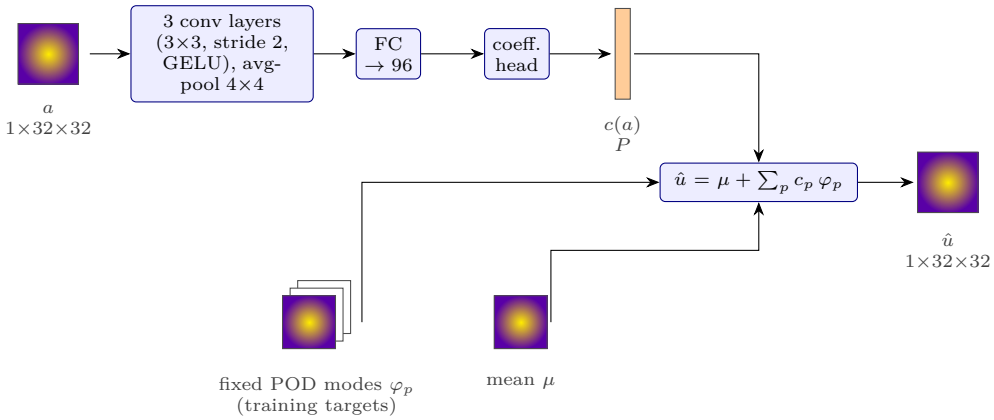

Figure 22: POD-DeepONet (shown for Navier–Stokes, 32×32). A convolutional encoder and a small head map the input to P coefficients $c(a)$. The prediction is the mean plus the coefficient-weighted sum of a fixed POD basis computed once from the training targets. At another resolution every basis mode and the mean are resampled bilinearly. On Darcy the input is 16×16.

D Hyperparameters and implementation details

D.1 Training

Table 9: Training configuration. Every setting is identical across architectures within a task.

	Darcy and Navier–Stokes	Lensing
Optimizer	AdamW	Adam (L2 weight decay)
Learning rate / weight decay	10^{-3} / 10^{-5}	10^{-3} / 10^{-5}
Schedule	cosine annealing over 60 epochs	cosine annealing over 60 epochs
Epochs	60	60
Batch size	16 (last partial batch dropped)	32
Loss	MSE	relative L_2 (mean over samples)
Gradient clipping	none	global norm 1.0
FFT normalization (spectral layers)	orthonormal	PyTorch default
Checkpoint selection	lowest in-distribution test error	lowest in-distribution (WDM) test error
Seeds	13, 42, 97, 7, 64, 128, 2024	

D.2 Architecture configurations

The Darcy and Navier–Stokes tasks use identical architecture configurations (Table 10). Here L is the number of spectral blocks, and "levels" is the number of pooling levels of the U-shaped models. The lensing configuration is in Table 11.

Table 10: Architecture hyperparameters for Darcy and Navier–Stokes, matched to the 112,387 parameters of BALANI-NO. $F_{\max}$ is 8 on Darcy and 16 on Navier–Stokes.

Model	Configuration	Params
BALANI-NO	width 18, J=4 poles, deg =2, m=4, L=3	112,387
FNO	width 16, (m_1, m_2)=(6, 6), L=3	111,857
ALNO	width 32, N=6 poles, M=6, L=3	115,169
LNO	width 24, N=16 poles, L=3	113,257
PDNO	width 16, MLP: 2 hidden layers of 64, n_{freq}=3, L=3	116,465
UNO	width 24, N_f=16, 3 levels	115,265
U-FNO	width 14, (m_1, m_2)=(5, 5), 3 levels	119,309
HC-UNO	width 20, N_f=20, hypernetwork hidden width 32, 3 levels	112,065
CNN	width 32, 3 levels	112,961
DeepONet	branch input 16×16, P=64, branch MLP 2×64, trunk MLP 4×160	112,833
POD-DeepONet	P=64 modes, conv channels 16/32/48, FC width 96, branch dim 96, coefficient head 64	112,176

Table 11: Architecture hyperparameters for lensing. BALANI-NO has 832,890 parameters and all baselines lie between −1.9% and +10.4% of this. $F_{\max}$=16. The spectral non-U-shaped models omit GroupNorm.

Model	Configuration	Params
BALANI-NO	width 32, J=6 poles, deg =2, m=8, L=4	832,890
FNO	width 56, (m_1, m_2)=(6, 6), L=4	919,466
ALNO	width 80, N=4 poles, M=8, L=4	852,082
LNO	width 80, N=8 poles, L=4	852,082
PDNO	width 40, MLP: 3 hidden layers of 64, n_{freq}=4, L=4	878,586
UNO	width 64, N_f=16, 3 levels	816,770
U-FNO	width 44, (m_1, m_2)=(6, 6), 3 levels	851,006
HC-UNO	width 64, N_f=16, hypernetwork hidden width 32, 3 levels	828,962
CNN	width 88, 3 levels	847,882
DeepONet	branch input 64×64 (flattened), P=96, branch MLP 2×160, trunk MLP 5×160	831,170
POD-DeepONet	P=160 modes per component, conv channels 32/64/128, FC width 320, branch dim 128, coefficient head 96	832,864

D.3 Implementation details

- Coordinates. On Darcy and Navier–Stokes, two coordinate channels (uniform on $[0, 1]$) are concatenated to the input of all spectral and CNN models; DeepONet and POD-DeepONet receive the raw field. On lensing the models append a fixed 64×64 coordinate grid internally, and DeepONet and POD-DeepONet receive the raw κ map.

- BALANI-NO initialization. Pole parameters $\hat{s}_j$ are linearly spaced in $[-6, 3]$; the coefficients $c_{j,p,q}$ are Gaussian with standard deviation $1/C$; the offset D is initialized to zero. Table weights are initialized at scale $1/C^2$ (Gaussian on Darcy and Navier–Stokes, uniform on $[0, 1)$ in real and imaginary parts on lensing). FNO tables use the same scale.
- $F_{\max}$. UNO and HC-UNO use $F_{\max} = 8$ on Darcy, 16 on Navier–Stokes and 16 on lensing.
- Branch inputs at other resolutions. DeepONet's branch on Darcy and Navier–Stokes takes a 16×16 bilinear resampling of the input (so on Navier–Stokes it sees a downsampled 32×32 or 64×64 field), which keeps the parameter budget matched. POD-DeepONet's branch input is resampled to the training resolution (16×16 on Darcy, 32×32 on Navier–Stokes) at other resolutions.
- POD basis. Computed once per task from the centered training targets with randomized low-rank PCA (rank P, four power iterations), independent of the seed. On lensing the 160 modes per component explain essentially all variance (1.0000 to four decimals).
- Ablation variants. Poles-only removes T from every block; table-only removes R and the offset D. All other components are unchanged.

E Additional Task: Gravitational Potential Recovery on Cosmological Density Maps

As a further test of the pole branch outside the three benchmarks of Section 3, we consider the recovery of the gravitational potential ϕ from a projected matter density field ρ via the 2D Poisson equation, $\nabla^2\phi \propto \rho$. In Fourier space this is $\hat{\phi}(k) = -\hat{\rho}(k)/|k|^2$ (with the $k = 0$ mode fixed by mean-subtracting the source), a single-pole, degree-0 instance of equation 4 in the limit $s_j \to 0$: it is even more directly matched to the pole branch's assumed structure than the lensing operator of Section 3.3, which additionally requires a degree-one numerator. As in the lensing task, ϕ is defined only up to an overall multiplicative constant (the discrete Laplacian is solved in dimensionless pixel units, not rescaled to the simulation's physical box size), so relative errors are unaffected.

We use total-matter surface-density maps from the CAMELS Multifield Dataset (CMD) (Villaescusa-Navarro et al., 2022), generated from the gravity-only (N-body) IllustrisTNG suite of the CAMELS project (Villaescusa-Navarro et al., 2021). Specifically, we use the CV (cosmic-variance) set at $z = 0$: 27 simulation volumes at a single fixed fiducial cosmology, each viewed from 15 projections, giving 405 native-resolution 256×256 maps of a $(25\,h^{-1}\mathrm{Mpc})^2$ region. This is a smaller and less cosmologically diverse dataset than the CAMELS Latin-Hypercube (LH) set(1000 cosmologies $\times$ 15 projections, $\sim$37$\times$ more maps, varying Ω_m and σ_8), thus we request to read the results as a fixed-cosmology, single-suite pilot, instead of a test of generalization across cosmological parameters.

The potential is computed once, at the native 256×256 resolution, by solving the discrete Poisson equation in Fourier space with periodic boundary conditions on the mean-subtracted density (Eq. equation 18 below), then both the density and the resulting potential are bilinearly downsampled to the target resolutions. Both fields therefore derive from a single consistent high-resolution solve, as in the Darcy and Navier–Stokes benchmarks (Section 3), rather than being resolved independently at each target resolution.

$$\hat{\phi}_{256}(k) = -\frac{\widehat{(\rho_{256} - \bar{\rho}_{256})}(k)}{|k|^2}, \qquad k \neq 0, \qquad \hat{\phi}_{256}(0) = 0. \tag{18}$$

The network input is $\log_{10}(\rho + 10^{-6})$, not ρ itself. Because the Poisson solve of equation 18 is linear in ρ but the network is only given $\log_{10}\rho$, the resulting input–output map is not linear in the network's own input channel, unlike the Darcy coefficient field or the lensing convergence map.

We hold out 10% of the 405 maps (40 samples) as a joint in-distribution/zero-shot test set, resized to 32×32 and 64×64 respectively from the same samples; the remaining 365 maps, downsampled to 32×32, are used for training. Inputs and targets are standardized using the mean and standard deviation of the 32×32 training split only, applied unchanged at both test resolutions. The accompanying cosmological-parameter file released with CMD is not used: the model receives only the single-channel density map, with no conditioning on cosmology.

Architectures and training otherwise follow the Darcy/Navier–Stokes protocol of Section 3 and Table 9, with parameter budgets closely matched to BALANI-NO's 112,351 parameters (within -0.8% to $+2.3\%$ across all eleven architectures).

BALANI-NO has the lowest zero-shot error at 64×64 (0.1769), 11% below the runner-up FNO (0.1989). At the training resolution it is narrowly second to PDNO (0.2318 vs. 0.2256, a 3% gap), which then collapses at 64×64 to a relative error above 1 (1.6430, a 7.28$\times$ degradation) – markedly worse even than its own degradation on Darcy (4.79$\times$, Table 1). CNN shows the same pattern (4.45$\times$), consistent with both models' parameters being tied to the training grid. Unlike Darcy and Navier–Stokes, most of the grid-independent spectral models (BALANI-NO, FNO, ALNO, LNO, and POD-DeepONet) have lower error at 64×64 than at 32×32 (degradation $< 1\times$). A plausible explanation is that, because both ρ and ϕ are downsampled from a single 256×256 solve, the 8$\times$ downsampling to 32×32 discards more of the true potential's structure than the 4$\times$ downsampling to 64×64, making the 32×32 regression target relatively noisier; we do not diagnose this further here. UNO and

Table 12: Gravitational potential recovery from CAMELS (Mtot, N-body IllustrisTNG, CV set): relative L_2 error at 32×32 (in-distribution) and 64×64 (zero-shot), mean $\pm$ std over 7 seeds. Degradation is the ratio of zero-shot to in-distribution error.

Architecture	Params	@32 (test)	@64 (0-shot)	Degradation
BALANI-NO	112,351	0.2318 ± 0.0019	$\mathbf{0.1769} \pm 0.0050$	0.76×
FNO	111,825	0.2464 ± 0.0092	0.1989 ± 0.0071	0.81×
ALNO	115,105	0.2686 ± 0.0019	0.2105 ± 0.0039	0.78×
LNO	113,209	0.4681 ± 0.0079	0.4380 ± 0.0065	0.94×
PDNO	116,433	$\mathbf{0.2256} \pm 0.0070$	1.6430 ± 0.0217	7.28×
UNO	115,217	0.2645 ± 0.0104	0.5356 ± 0.1607	2.03×
U-FNO	119,281	0.2401 ± 0.0038	0.2819 ± 0.0337	1.17×
HC-UNO	111,449	0.2411 ± 0.0026	0.2459 ± 0.0130	1.02×
CNN	112,897	0.2885 ± 0.0032	1.2841 ± 0.0218	4.45×
DeepONet	112,833	0.7329 ± 0.0121	0.7668 ± 0.0119	1.05×
POD-DeepONet	112,176	0.4533 ± 0.0027	0.4318 ± 0.0068	0.95×

U-FNO, whose pooling ties them to the input grid at each level, degrade in the expected direction (2.03× and 1.17×).

The evidence here is preliminary relative to the rest of the paper: a single fixed-cosmology simulation suite (405 maps, 40 held out for testing, vs. 1,000–2,000 training samples elsewhere in this paper) and a nonlinear (log-density-to-potential) rather than linear-elliptic map, for the reasons discussed above. We thus report this as a preliminary additional data point for the pole branch's degree-0, single-pole and leave a Latin-Hypercube-based version of this task, with cosmology-parameter conditioning, to future work.

F Additional Task: Magnetic Field Upward Continuation on Real-World Geophysical Data

As a complementary test to Appendix E, we consider upward continuation of the Earth's magnetic anomaly field. The exact operator is isotropic and grid-independent, as in the pole branch of Eq. equation 4, but its Fourier symbol is not rational. Upward continuation by a height Δz multiplies each Fourier mode by $\exp(-|k|\Delta z)$ (Blakely, 1996). This provides a test of whether a finite sum of learned poles can transfer across resolutions when the target operator has a transcendental rather than rational Fourier symbol. Unlike the other tasks in this paper, the input data are measured magnetic fields rather than synthetic simulations.

Away from the magnetized source region, the magnetic anomaly field satisfies Laplace's equation. The field at height Δz is therefore related to the surface field by $\hat{F}(k, \Delta z) = \hat{F}(k, 0) \exp(-|k|\Delta z)$ (Blakely, 1996). We use $\Delta z = 2$ km throughout. Larger values of Δz produce a more localized real-space kernel and are expected to be harder to approximate with a low-order rational function. Such cases are not considered here.

We use the Earth Magnetic Anomaly Grid at 2 arc-minute resolution, version 3 (EMAG2v3) (Meyer et al., 2017), a global compilation of marine, airborne, and satellite magnetic measurements released by NOAA's National Centers for Environmental Information. We use the continuous 4 km-altitude EMAG2v3 grid, downloaded from NOAA NCEI at https://www.ngdc.noaa.gov/geomag/data/EMAG2/EMAG2_V3_UpCont_DataTiff.tif (DOI 10.7289/V5H70CVX). The reference surface is therefore the standard 4 km altitude above the geoid, so the $\Delta z = 2$ km continuation applied here is from 4 km to 6 km altitude, not from sea level.

We use 64×64 pixel patches within the latitude band $-30°$ to $30°$ to maintain the small-angle approximation for pixel-to-kilometer conversion. The north–south spacing is 111.32 km/deg $\times$ $(2/60)°$, while the east–west spacing is scaled by $\cos(\text{lat}_0)$ at the patch center. The two spacings differ by at most $\sim 13\%$ within this band. Before each FFT-based continuation, patches are tapered with a Tukey window ($\alpha = 0.25$) and zero-padded by 50% on each side to reduce wraparound artifacts. The inverse-transformed field is then cropped

to 64×64. Each patch is normalized using its own mean and standard deviation, and the upward-continuation target is computed from the normalized field.

Training patches are drawn only from the western half of the sampled longitude range and test patches only from the eastern half, so training and evaluation regions never overlap geographically. This means the "@64" column below is not an in-distribution test in the sense used elsewhere in this paper (held-out samples from the same pool): it is a same-resolution, held-out-region test, and only the "@128" column adds a resolution shift on top of that. Patches containing missing or non-finite values are rejected during sampling; targets are up to 1,200 training and 300 test patches.

The 128×128 evaluation uses the same test locations and measured 64×64 fields, which are bicubically upsampled to 128×128 over the same physical extent. The target is recomputed at the finer spacing using the exact kernel $\exp(-|k|\Delta z)$. Since the native EMAG2v3 resolution corresponds to the original 64×64 grid, upsampling introduces no new physical high-frequency information. The experiment therefore tests whether the learned operator remains consistent under a finer discretization, analogous to the Darcy $16 \to 32$ and Navier–Stokes $32 \to 64$ tests, rather than generalization to higher-resolution measurements.

Table 13: Magnetic field upward continuation on EMAG2v3 ($\Delta z = 2$ km): relative L_2 error at 64×64 (same resolution, held-out region) and 128×128 (zero-shot resolution, same held-out region), mean $\pm$ std over 7 seeds. Degradation is the ratio of the two.

Architecture	Params	@64	@128 (0-shot)	Degradation
BALANI-NO	112,387	0.0113 ± 0.0002	$\mathbf{0.0399} \pm 0.0001$	3.54×
FNO	111,857	0.0127 ± 0.0006	0.0402 ± 0.0003	3.17×
ALNO	115,169	0.0209 ± 0.0008	0.0454 ± 0.0001	2.18×
LNO	113,257	0.0361 ± 0.0002	0.0550 ± 0.0002	1.53×
PDNO	116,465	0.0091 ± 0.0003	0.2755 ± 0.0140	30.42×
UNO	115,265	0.0132 ± 0.0016	0.0888 ± 0.0069	6.71×
U-FNO	119,309	0.0130 ± 0.0002	0.0734 ± 0.0114	5.63×
HC-UNO	112,065	0.0109 ± 0.0006	0.0661 ± 0.0063	6.09×
CNN	112,961	$\mathbf{0.0067} \pm 0.0002$	0.2791 ± 0.0027	41.56×
DeepONet	112,833	0.2888 ± 0.0018	0.2890 ± 0.0015	1.00×
POD-DeepONet	112,176	0.2077 ± 0.0008	0.2087 ± 0.0008	1.00×

At matched resolution (@64), BALANI-NO ranks fourth with an error of 0.011, behind CNN (0.0067), PDNO (0.0091), and HC-UNO (0.0109). These models have parameters more closely tied to the 64×64 grid, consistent with the results for Darcy (Table 1) and CAMELS (Appendix E). At @128, BALANI-NO achieves the lowest error (0.0399), followed by FNO, ALNO (0.0454), and LNO (0.0550). PDNO and CNN show the largest degradation (30.42× and 41.56×), reaching errors of approximately 0.28. Their degradation is substantially larger than on Darcy (4.79× and 8.61×) and CAMELS (7.28× and 4.45×), reflecting the stronger resolution shift in this experiment. DeepONet and POD-DeepONet maintain 1.00× degradation but have substantially higher errors (0.21–0.29).

BALANI-NO therefore transfers well despite using a rational pole parameterization while the exact continuation kernel $\exp(-|k|\Delta z)$ is non-rational in k. This result is consistent with the pole branch providing a low-order approximation to smooth isotropic spectral profiles beyond exactly rational operators.

The inputs are real measured fields, while the targets are generated by applying the known analytic kernel rather than from independent measurements at a second altitude. The reported errors therefore measure approximation to the analytic operator. The @64 results include a geographic shift between the western training region and eastern test region, so they are not directly comparable to the in-distribution results in Sections 3.1–3.3. We use $\Delta z = 2$ km throughout. Larger continuation heights provide a stronger test because the kernel becomes more sharply decaying and harder to approximate with a low-order rational function. We leave this regime for future work.

G Architecture configurations for the additional tasks

The gravitational-potential-recovery task (Appendix E) and the magnetic-field upward-continuation task (Appendix F) use the eleven per-architecture configurations from Table 10 without modification. This includes the network widths, pole counts, retained modes, and MLP or convolutional branch sizes. The only resolution-dependent parameter is F_{max}, which follows the Nyquist frequency of the training grid (Appendix D). We use $F_{max} = 16$ for the 32×32 gravitational task and $F_{max} = 32$ for the 64×64 magnetic task. POD-DeepONet uses $P = 64$ modes in both tasks with the randomized low-rank PCA procedure described in Appendix D. Both tasks use three input channels consisting of the field and two coordinate channels and one output channel.

Training follows the Darcy/Navier–Stokes configuration in Table 9: AdamW with learning rate 10^{-3} and weight decay 10^{-5}, cosine annealing over 60 epochs, batch size 16, MSE loss, no gradient clipping, and orthonormal FFT normalization. Checkpoints are selected using the lowest error on the same-resolution test split and never on the zero-shot split. Relative L_2 error is computed per sample and averaged first within batches and then across batches, following the Darcy and Navier–Stokes evaluation.

H Statement on the Use of Large Language Models

In accordance with the conference policy on the disclosure of generative AI usage, we report below how large language models (LLMs) were used in the preparation of this work.

- Generate synthetic datasets. Not used. The Darcy flow and Navier–Stokes benchmarks are pre-existing public datasets obtained from Zenodo via the neuraloperator library. The gravitational-lensing dataset was generated by the authors using lenstronomy (Birrer & Amara, 2018; Birrer et al., 2021).
- Help develop theoretical models or conceptual frameworks. AI assistance was used only in the development of HC-UNO. BALANI-NO, the paper's main architecture, was conceived and developed by the authors. These are the only two novel architectures introduced in this work; all other architectures are previously published methods used as baselines.
- Formulate mathematical claims. Not applicable. The paper makes no mathematical claims requiring formal statement or proof.
- Provide critical ingredients for proving mathematical claims. Not applicable, for the same reason as above.
- Assist in the writing of proofs. Not applicable. The paper contains no proofs.
- Propose or refine hypotheses. AI was used to help refine and polish the text explaining why BALANI-NO's isotropic pole parameterization is preferable to the axis-aligned pole structure used by ALNO and LNO.
- Design or provide feedback on research methodology or experiments. AI was used to run a parameter sweep across architectures to match parameter budgets, ensuring a fair comparison at matched capacity.
- Implement methods. Not used for architecture design. All architectures were implemented by the authors.
- Assist with translation. Not applicable. No translation was performed as part of this research.
- Clean and reformat datasets. Not used. The Darcy and Navier–Stokes datasets were obtained from Zenodo and downsampled by the authors without AI assistance.
- Support qualitative and thematic data analysis. AI was used to polish text and to help create six of the eleven architecture diagrams in the appendix, specifically all diagrams except those for BALANI-NO, PDNO, LNO, ALNO, and FNO.
- Interpret results. Results were interpreted by the authors. AI was used only to polish the accompanying text.

- Formulate questions for surveys or interviews. Not applicable. No surveys or interviews were conducted as part of this research.
- Create or modify scientific figures or images. AI was used to help create six of the eleven architecture diagrams in the appendix. All plots, remaining figures, and images were created by the authors.
- Suggest experimental parameters. AI was used to perform a parameter sweep so that all eleven architectures were matched in parameter budget for a fair comparison.
- Create or edit software code. Approximately 35% of the code used in this work was AI-written; the remaining 65% was written by the authors. All AI-generated code was reviewed and independently tested by the authors.
- Creation of artifacts. AI-assisted artifacts include portions of the software code and six of the eleven architecture diagrams in the appendix. Further details are provided in the individual disclosures above.
- Draft parts of a research paper. AI was used to draft the introduction, the related work section, and parts of the ablation and appendix sections. These sections were subsequently reviewed and revised by the authors.
- Transcribe recordings of research material. Not applicable. No such transcription was performed.
- Summarize or analyze existing literature. AI was used to summarize and analyze existing literature and to assist in drafting the related work section. All cited literature was reviewed by the authors against the original sources.
- Discover research topics or identify gaps. Not used for this purpose.
- Brainstorming. AI was used for brainstorming, which contributed to the development of HC-UNO.
- Sourcing/searching for information. AI was used to search for information on relevant datasets. Information used in the paper was subsequently verified by the authors against the underlying sources.
- Edit a research paper to improve readability. AI was used to polish and edit text throughout the paper for readability. All final text was reviewed by the authors.
- Identify relevant literature. AI was used to help identify relevant literature. The relevance and bibliographic details of cited works were manually verified by the authors.
- Format references. AI was used to format references. Reference entries were reviewed by the authors for correctness.
- Suggest a structure for a research paper. Not used for this purpose.
- Propose a title or keywords for a research paper. Not used for this purpose.

We have reviewed all AI-assisted work. AI-generated code was reviewed and independently tested by the authors for correctness. AI-drafted text was fact-checked against the underlying experiments, results, and cited sources. AI-assisted architecture diagrams were checked by the authors against the corresponding architectures and equations in Appendix B. Literature identified or analyzed with AI assistance was manually verified against the original sources for relevance and accuracy. All final methodological decisions, experimental results, interpretations, claims, and conclusions were made and verified by the authors. We take responsibility for the final content of this work, including text, claims, code, figures, and other artifacts produced with the aid of generative AI.