KERNEL-ADAPTIVE PHYSICS-INFORMED SHALLOW META-LEARNING FOR PARAMETRIC LINEAR PDES

Vikas Dwivedi, Monica Sigovan & Bruno Sixou
CREATIS, INSA-Lyon; Inserm, U1044; CNRS UMR 5220
Université Lyon 1; Université de Lyon
Lyon, 69621, France
{vikas.dwivedi,monica.sigovan,bruno.sixou}@creatis.insa-lyon.fr

ABSTRACT

Parametric families of linear partial differential equations (PDEs) arise widely in scientific and engineering applications, where solutions depend continuously on physical parameters such as source location, transport velocity, or diffusion strength. Many physics-informed and neural operator approaches either require retraining for each parameter instance or rely on high-dimensional learned representations that can be difficult to interpret and control. We propose a kernel-adaptive physics-informed meta-learning framework for parametric linear PDEs in which task variation is modeled through low-dimensional deformations of kernel distributions rather than task-specific network weights. The solution is represented as a linear combination of physics-aligned kernel functions whose centers, widths, sparsity gates, and optionally coefficients are conditioned on PDE parameters through a lightweight meta-parameterization. By exploiting linearity, the formulation preserves superposition and can be trained without inner-loop optimization; in the studied settings, it admits an interpretation in terms of parameter-dependent kernel responses analogous to Green's-function behavior. We evaluate the method on representative elliptic (Poisson), hyperbolic (advection), and mixed parabolic–hyperbolic (advection–diffusion) PDE families and compare against parameter-conditioned neural baselines including PI-DeepONet and FiLM–HyperPINN. Across these cases, the proposed approach achieves $\mathcal{O}(10^{-2})$–$\mathcal{O}(10^{-1})$ relative L^2 errors within the training parameter ranges without per-instance retraining, and exhibits structured error growth under extrapolation consistent with resolution and parameter-support limits of the learned kernel manifold. These comparisons illustrate how separable operator decompositions, feature-wise modulation, and adaptive kernel geometry induce distinct inductive biases under parameter shift.

1 INTRODUCTION

Parametric families of partial differential equations (PDEs) arise ubiquitously in science and engineering, where solutions depend continuously on physical parameters such as source location, transport velocity, or diffusion strength. Applications include uncertainty quantification, design optimization, inverse problems, and multi-query simulations, where repeatedly solving a PDE for different parameter values can be computationally expensive. A natural learning objective is therefore to construct a *single* solution representation that generalizes across parameter space without requiring retraining for each new instance.

Physics-informed neural networks (PINNs (Raissi et al., 2019; Ramabathiran & Ramachandran, 2021)) approximate PDE solutions by minimizing differential equation residuals without labeled data. While effective for individual PDE instances, standard PINNs are typically trained separately for each parameter configuration and do not directly yield operator-level generalization across parametric families. Extensions based on adaptive sampling (Lu et al., 2021b), curriculum strategies (Krishnapriyan et al., 2021; Dwivedi et al., 2025), and domain decomposition Jagtap et al. (2020); Dwivedi et al. (2021) improve robustness and convergence in many settings, but remain largely instance-specific.

To address parametric dependence, several works have explored meta-learning (Finn et al., 2017) and parameterized PINN formulations (de Avila Belbute-Peres et al., 2021; Qin et al., 2022). These approaches condition model parameters on physical inputs using hypernetworks or weight-based adaptation. More generally, feature-wise modulation mechanisms such as FiLM (Perez et al., 2018) provide a lightweight strategy for conditioning intermediate network representations on external variables and have been adopted in various conditional modeling contexts. While flexible and expressive, such conditioning schemes typically operate through high-dimensional latent features and may require inner-loop optimization or fine-tuning, which can complicate interpretation and analysis of generalization behavior.

In parallel, neural operator methods such as DeepONet (Lu et al., 2021a) and Fourier Neural Operators (FNOs) (Li et al., 2020) learn mappings between function spaces using supervised data. These approaches parameterize solution operators through separable branch–trunk architectures or global spectral representations and have demonstrated strong empirical performance. However, they rely on dense latent embeddings whose geometric interpretation and resolution limits can be difficult to characterize, particularly under parameter extrapolation.

In this work, we focus on parametric *linear* PDEs. Rather than learning task-specific weights or dense latent operator embeddings, we exploit linearity to represent solutions as linear combinations of kernel functions whose *geometry*—including centers, widths, and sparsity patterns—is conditioned on physical parameters. This kernel-adaptive, physics-informed meta-learning formulation yields a compact parameter-conditioned solution representation that generalizes across the studied parameter regimes without per-instance retraining. By explicitly structuring the hypothesis class through adaptive kernel geometry, the approach enables analysis of generalization and extrapolation behavior in terms of kernel resolution and parameter support, providing a transparent link between representation structure and approximation performance.

CONTRIBUTIONS

The primary contributions of this work are:

- **Kernel-distribution meta-learning for linear PDEs.** We introduce a formulation in which task variation across a parametric PDE family is captured through low-dimensional, parameter-conditioned deformations of kernel distributions (centers, widths, and sparsity gates), while maintaining a shared set of global coefficients.
- **Physics-aligned linear solution representation.** By exploiting the linearity of the underlying operators, the proposed model represents solutions as linear combinations of adaptive kernel functions, preserving superposition and enabling physics-informed training without inner-loop optimization in the settings considered.
- **Geometry-aware kernel parameterization.** We design kernel parameterizations aligned with problem structure, including characteristic transport directions in hyperbolic equations and diffusion-controlled length scales in parabolic regimes.
- **Empirical evaluation across PDE classes.** We evaluate the approach on representative Poisson, advection, and advection–diffusion problems, and compare against PI-DeepONet and FiLM-conditioned HyperPINN baselines, highlighting distinct inductive biases under interpolation and parameter extrapolation.
- **Interpretable generalization behavior.** Through systematic evaluation, we show that observed failure modes align with spectral and parameter-support limitations of the learned kernel manifold, providing insight into approximation behavior under parameter shift.

Organization of the paper. Section 2 introduces the formulation for parametric families of linear PDEs and presents the kernel-adaptive meta-learning hypothesis class together with the physics-informed training objective. Section 3 reports quantitative and qualitative results across elliptic (Poisson), hyperbolic (advection), and mixed advection–diffusion families, including comparisons with parameter-conditioned neural baselines. Section 4 concludes with directions for future work.

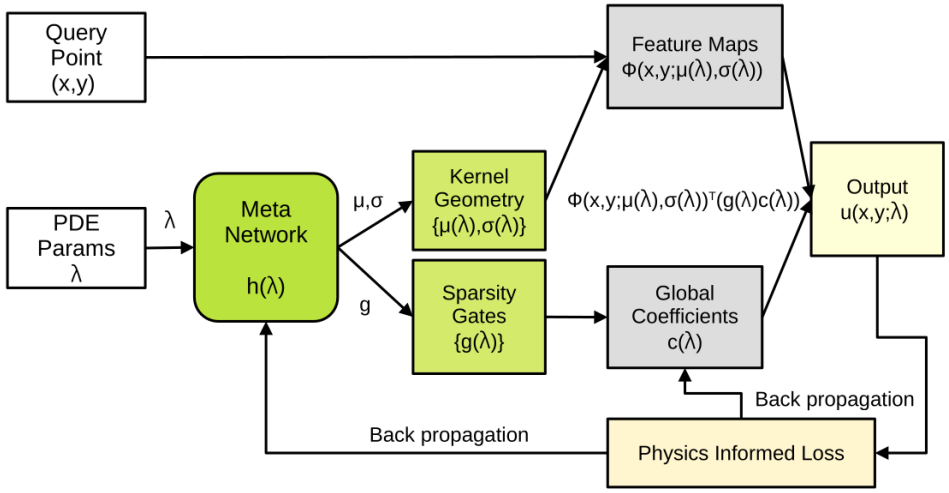

Figure 1: Overview of the kernel-adaptive physics-informed meta-learning framework. Given PDE parameters λ, a lightweight meta-network $h(\lambda)$ generates task-conditioned kernel geometry parameters $\{\mu(\lambda), \sigma(\lambda)\}$, sparsity gates $g(\lambda)$, and optionally kernel coefficients $c(\lambda)$. For each query point (x, y) or (x, t), feature maps $\Phi(x; \mu(\lambda), \sigma(\lambda))$ are evaluated and linearly combined to produce the solution $u(x; \lambda)$. Depending on the PDE family, the kernel coefficients may either be shared across tasks (global coefficients) or generated by the meta-network as parameter-conditioned coefficients. Training is performed by minimizing a physics-informed loss enforcing the PDE and boundary/initial conditions across a distribution of parameters λ. The architecture preserves linear superposition, avoids task-specific retraining, and yields an interpretable, geometry-aware meta-solution operator.

2 Mathematical Formulation

We first give a high-level overview of the proposed kernel-adaptive physics-informed meta-learning architecture (Fig. 1), before introducing the formal problem statement and mathematical details.

2.1 Problem statement: parametric families of linear PDEs

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with boundary $\partial\Omega$, and let $\Lambda \subset \mathbb{R}^p$ denote a compact parameter set. For each parameter $\lambda \in \Lambda$, consider a linear boundary/initial value problem assumed to be well-posed in an appropriate function space, of the form

$$\mathcal{L}_\lambda u_\lambda(x) = f_\lambda(x), \qquad x \in \Omega, \tag{1}$$
$$\mathcal{B}_\lambda u_\lambda(x) = g_\lambda(x), \qquad x \in \partial\Omega \text{ (and/or on an initial hypersurface).} \tag{2}$$

Here $\mathcal{L}_\lambda$ is a linear differential operator (with λ-dependent coefficients allowed), f_λ is a source term, and $\mathcal{B}_\lambda$ is a linear boundary/initial operator encoding Dirichlet, Neumann, periodic, or initial conditions. We assume that for each $\lambda \in \Lambda$, the problem admits a unique solution u_λ in a suitable function space $\mathcal{U}$ (e.g., a Sobolev space consistent with $\mathcal{L}_\lambda$ and $\mathcal{B}_\lambda$).

Goal (meta-solution operator). Rather than solving (1)–(2) for a single λ, we seek to learn a *single* parameter-conditioned mapping

$$\mathcal{S} : \Lambda \to \mathcal{U}, \qquad \lambda \mapsto u_\lambda, \tag{3}$$

i.e., a *solution operator* over the PDE family. Our objective is to construct a model $u_\theta(x; \lambda)$ such that $u_\theta(\cdot; \lambda) \approx u_\lambda(\cdot)$ for all λ in the training distribution, while also characterizing how performance degrades under parameter extrapolation.

2.2 THREE REPRESENTATIVE LINEAR PDE FAMILIES

We consider three canonical linear PDE classes—elliptic, hyperbolic, and mixed parabolic–hyperbolic—each parameterized by a low-dimensional λ that controls spatial variation and gradients of initial conditions or source terms.

Test Case-1: Poisson with parametric Gaussian source. Let $\Omega = [0,1]^2$ and $\lambda = (x_0, y_0, \nu)$. Define

$$-\Delta u(x, y; \lambda) = f(x, y; \lambda), \qquad u|_{\partial\Omega} = 0, \tag{4}$$

where a representative choice is a Gaussian source

$$f(x, y; \lambda) = \frac{1}{2\pi\nu^2} \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\nu^2}\right). \tag{5}$$

Test Case-2: Periodic advection with parametric initial condition. Let $\Omega = [0,1] \times [0,1]$ with $\lambda = (x_0, \nu)$. Consider

$$u_t + u_x = 0, \qquad u(x, 0) = u_0(x; \lambda), \qquad u(0, t) = u(1, t). \tag{6}$$

A representative u_0 is a periodic Gaussian bump centered at x_0 with width ν.

Test Case-3: Advection–diffusion with parametric transport and diffusion. Let $\Omega = [0,1] \times [0,T]$ and $\lambda = (a, \nu)$. Consider

$$u_t + a\, u_x - \nu\, u_{xx} = 0, \tag{7}$$

together with an initial condition $u(x, 0) = u_0(x; \lambda)$ and boundary conditions (Dirichlet or periodic). In our experiments, boundary or initial traces may be enforced by matching known (exact) traces, but the formulation below only requires that $\mathcal{B}_\lambda u = g_\lambda$ be linear and well-defined.

2.3 META-LEARNED SOLUTION OPERATOR

We approximate the solution operator $\mathcal{S}$ by a parametric model

$$u_\theta : \Omega \times \Lambda \to \mathbb{R}, \qquad (x, \lambda) \mapsto u_\theta(x; \lambda), \tag{8}$$

where θ is shared across all tasks λ. The meta-learning aspect is that a *single* parameter set θ must support many PDE instances, with no per-instance retraining at test time.

2.4 DISTRIBUTIONAL RBF HYPOTHESIS CLASS

The core modeling choice is to represent $u_\theta(\cdot; \lambda)$ as a linear combination of basis functions whose *distributional parameters* depend on λ.

Kernel dictionary with parameter-conditioned geometry. Let $\{\phi(\cdot; \mu, \sigma)\}$ be a radial basis kernel family, for example a Gaussian RBF,

$$\phi(x; \mu, \sigma) = \exp\left(-\frac{\|x - \mu\|_2^2}{2\sigma^2}\right), \tag{9}$$

where $\mu \in \mathbb{R}^d$ is a center and $\sigma > 0$ is a width.

We introduce M kernels with parameter-conditioned centers and widths,

$$\mu_j(\lambda) \in \mathbb{R}^d, \qquad \sigma_j(\lambda) \in \mathbb{R}_{>0}, \qquad j = 1, \dots, M. \tag{10}$$

We also introduce sparsity gates $g_j(\lambda) \in (0, 1)$ that modulate kernel activity.

The resulting model takes the form

$$u_\theta(x; \lambda) = \sum_{j=1}^{M} c_j(\lambda)\, g_j(\lambda)\, \phi\big(x; \mu_j(\lambda), \sigma_j(\lambda)\big), \tag{11}$$

where the coefficients $c_j(\lambda)$ weight the contribution of each kernel. Depending on the PDE family, these coefficients may either be shared across tasks (global coefficients c_j) or generated by the meta-network as parameter-conditioned coefficients $c_j(\lambda)$. In all cases, task variation is primarily captured through the kernel geometry $(\mu_j(\lambda), \sigma_j(\lambda))$ and sparsity gates $g_j(\lambda)$.

Meta-parameterization. We parameterize the maps

$$\lambda \mapsto \mu_j(\lambda), \qquad \lambda \mapsto \sigma_j(\lambda), \qquad \lambda \mapsto g_j(\lambda), \qquad \lambda \mapsto c_j(\lambda) \tag{12}$$

using a low-capacity meta-model (e.g., a small neural network or affine map). To ensure well-defined kernels, we enforce constraints by construction, for example

$$g_j(\lambda) = \sigma(\tilde{g}_j(\lambda)) \in (0, 1), \qquad \sigma_j(\lambda) = \mathrm{softplus}(\tilde{\sigma}_j(\lambda)) + \varepsilon > 0. \tag{13}$$

If needed, kernel centers can also be constrained to the computational domain using squashing maps (e.g., sigmoid transformations) or periodic wrapping.

Boundary-condition satisfaction (optional trial ansatz). When homogeneous Dirichlet conditions are present, one may enforce them exactly via a multiplicative factor

$$u_\theta(x; \lambda) = \psi(x)\, \tilde{u}_\theta(x; \lambda), \tag{14}$$

where $\psi(x) = 0$ on $\partial\Omega$ (e.g., $\psi(x, y) = x(1 - x)y(1 - y)$ on $[0, 1]^2$). In that case $\tilde{u}_\theta$ may be modeled by (11). For periodic or mixed constraints, boundary terms can alternatively be enforced through residual penalties.

2.5 Physics-informed meta-training objective

Let ρ be a distribution over parameters Λ (task distribution). We define residual operators

$$\mathcal{R}_\lambda[u](x) := \mathcal{L}_\lambda u(x) - f_\lambda(x), \qquad x \in \Omega, \tag{15}$$

and boundary or initial residuals

$$\mathcal{R}^{\mathrm{bc}}_\lambda[u](x) := \mathcal{B}_\lambda u(x) - g_\lambda(x), \qquad x \in \partial\Omega. \tag{16}$$

The population objective is

$$\min_\theta \ \mathbb{E}_{\lambda\sim\rho}\Big[\|\mathcal{R}_\lambda[u_\theta(\cdot; \lambda)]\|^2_{L^2(\Omega)} + \beta\, \|\mathcal{R}^{\mathrm{bc}}_\lambda[u_\theta(\cdot; \lambda)]\|^2_{L^2(\partial\Omega)} + \gamma\, \mathcal{S}_\lambda(\theta)\Big]. \tag{17}$$

A natural sparsity regularizer is

$$\mathcal{S}_\lambda(\theta) = \frac{1}{M} \sum_{j=1}^{M} |g_j(\lambda)|. \tag{18}$$

PDE	Test case	Parameters	Rel. L^2 error	Regime
Poisson	Center (in-range)	$(x_0, y_0, \nu) = (0.50, 0.50, 0.07)$	2.01×10^{-2}	interpolation
Poisson	Off-center (in-range)	$(0.45, 0.55, 0.09)$	1.20×10^{-2}	interpolation
Poisson	Shifted (out-of-range)	$(0.30, 0.30, 0.06)$	3.57×10^{-1}	spatial extrapolation
Poisson	Narrow (out-of-range)	$(0.50, 0.50, 0.03)$	3.87×10^{-2}	stiffness extrapolation
Advection	Center (in-range)	$(x_0, \nu) = (0.50, 0.07)$	2.30×10^{-2}	interpolation
Advection	Off-center (in-range)	$(0.30, 0.09)$	2.33×10^{-2}	interpolation
Advection	Narrow (in-range)	$(0.65, 0.04)$	3.21×10^{-2}	stiff but in-range
Advection	Narrow (out-of-range)	$(0.50, 0.02)$	2.41×10^{-1}	stiffness extrapolation
Adv.–Diff.	Mid-range	$(a, \nu) = (0.75, 0.03)$	6.48×10^{-2}	interpolation
Adv.–Diff.	Low a, high ν	$(0.55, 0.045)$	3.62×10^{-2}	diffusion-dominated
Adv.–Diff.	High a, low ν	$(0.95, 0.015)$	1.78×10^{-1}	transport-dominated
Adv.–Diff.	Low ν (out-of-range)	$(0.75, 0.008)$	4.09×10^{-1}	stiffness extrapolation

Table 1: Meta-learning performance across elliptic (Poisson), hyperbolic (advection), and mixed advection–diffusion PDEs. Training ranges: Poisson $x_0, y_0 \in [0.4, 0.6]$, $\nu \in [0.05, 0.1]$; Advection $x_0 \in [0.2, 0.8]$, $\nu \in [0.03, 0.12]$; Advection–diffusion $a \in [0.5, 1.0]$, $\nu \in [0.01, 0.05]$.

Empirical (collocation) approximation. In practice the expectation over λ is approximated via Monte Carlo sampling of tasks and collocation sets. For each sampled λ we define interior and boundary collocation sets $\mathcal{X}_{\text{int}}(\lambda)$ and $\mathcal{X}_{\text{bc}}(\lambda)$, leading to the empirical loss used during training.

Remark. The meta-learning occurs through the parameter-conditioned kernel distribution $\lambda \mapsto (\mu_j(\lambda), \sigma_j(\lambda), g_j(\lambda), c_j(\lambda))$, which defines a low-dimensional *kernel manifold* shared across tasks. In some settings (e.g., elliptic problems) the coefficients may remain global, while in others (e.g., transport-dominated PDEs) they can be parameter-conditioned to capture solution variations along characteristic directions.

3 Results and Discussion

We evaluate the proposed kernel-adaptive meta-learning framework on three representative linear PDE families: an elliptic Poisson problem, a hyperbolic advection equation, and a mixed advection–diffusion equation. In all cases, a single trained model is tested on multiple parameter instances without per-instance retraining, enabling a systematic study of in-distribution generalization and extrapolation beyond the training parameter support. Relative L^2 errors are reported in Table 1, and performance is analyzed in relation to operator structure and kernel resolution, with qualitative solution and error comparisons shown in Figs. 2–3 (Poisson), Fig. 4 (advection), and Fig. 5 (advection–diffusion).

3.1 Test Case-1: Poisson equation

We first consider the elliptic Poisson problem with a parametric Gaussian source and homogeneous Dirichlet boundary conditions. As shown in Table 1, the proposed meta-solution achieves relative L^2 errors on the order of 10^{-2} for in-range parameters, including both centered and moderately off-centered source locations. This indicates consistent generalization across the training support without per-instance retraining.

Two forms of extrapolation are examined. When the source width is reduced below the training range, the error increases moderately, reflecting the increased difficulty of resolving higher spatial frequencies. In contrast, shifting the source location outside the trained spatial region leads to a larger increase in error, indicating that the learned representation captures a local solution manifold in parameter space rather than enforcing global translation equivariance. In these out-of-range cases, errors remain structured rather than oscillatory; in our experiments, this behavior is consistent with approximation limits of the learned kernel manifold rather than numerical instability.

For additional context, we include a controlled baseline comparison against PI-DeepONet and FiLM–HyperPINN trained on the same Poisson family using the same parameter ranges and test cases; see Section C. Figure 2 shows the finite-difference reference solutions together with predictions from KAPI, FiLM–HyperPINN, and PI-DeepONet, while Fig. 3 visualizes the corresponding absolute error maps for the same parameter settings.

3.2 TEST CASE-2: ADVECTION EQUATION

We next consider the one-dimensional linear advection equation with periodic boundary conditions. In contrast to the Poisson case, in-range accuracy is largely insensitive to spatial shifts of the initial condition, with relative L^2 errors remaining approximately 2.3×10^{-2} across different values of x_0. This behavior is consistent with the translation equivariance of the advection operator, which is encoded through the characteristic coordinate in the kernel parameterization.

As the initial condition becomes sharper, accuracy degrades gradually. For narrow but in-range profiles, the error increases moderately, while extrapolation to significantly smaller diffusion widths leads to a larger increase. In the stiff regime, the error remains aligned with characteristic directions, and no visually apparent spurious oscillations are observed in our experiments. The observed degradation is consistent with resolution limits of the finite kernel representation.

3.3 TEST CASE-3: ADVECTION–DIFFUSION EQUATION

Finally, we consider the linear advection–diffusion equation, which interpolates between diffusion-dominated and transport-dominated regimes. In diffusion-dominated cases, the model achieves relatively lower errors, consistent with the smoothing effect of stronger diffusion. As the dynamics become increasingly transport-dominated, accuracy degrades, reflecting the growing difficulty of resolving sharp, advecting features with finite kernel bandwidth.

Extrapolation beyond the trained diffusion range leads to further increases in error, paralleling the stiffness-related degradation observed in both the Poisson and pure advection settings. In these cases, predicted solutions remain free of visually apparent spurious oscillations in our experiments, and error structures continue to align with characteristic transport directions. Overall, these results suggest that the proposed kernel-adaptive meta-learning framework provides a common approximation strategy across the studied elliptic, hyperbolic, and mixed PDE regimes, with limitations governed by parameter support and kernel resolution.

Limitations While the proposed kernel-adaptive physics-informed meta-learning framework shows encouraging performance for parametric families of linear PDEs in the studied settings, several limitations should be noted.

- **Restriction to linear PDEs.** The current formulation explicitly exploits linearity to preserve superposition and enable a shared set of global coefficients across tasks. Extending the approach to nonlinear PDEs would require additional mechanisms to capture nonlinear interactions and may alter the structure of the learned solution manifold.
- **Resolution-limited representation under strong stiffness.** Accuracy is inherently tied to the spectral resolution of the learned kernel dictionary. As observed in the experiments, performance degrades under strong stiffness extrapolation (e.g., very small diffusion coefficients or sharply localized sources), reflecting intrinsic bandwidth limitations of a finite kernel representation.
- **Local parameter-space generalization.** The learned meta-representation captures a solution manifold over the training parameter support. Large extrapolations—particularly spatial shifts not encoded through symmetry-aware parameterizations—lead to increased error. Global equivariance or invariance properties are not enforced by the current architecture.
- **Limited geometric and dimensional complexity.** Experiments are restricted to low-dimensional domains and relatively simple geometries. Extension to higher-dimensional problems, complex geometries, or irregular domains would require additional design choices in kernel parameterization and sampling strategies.

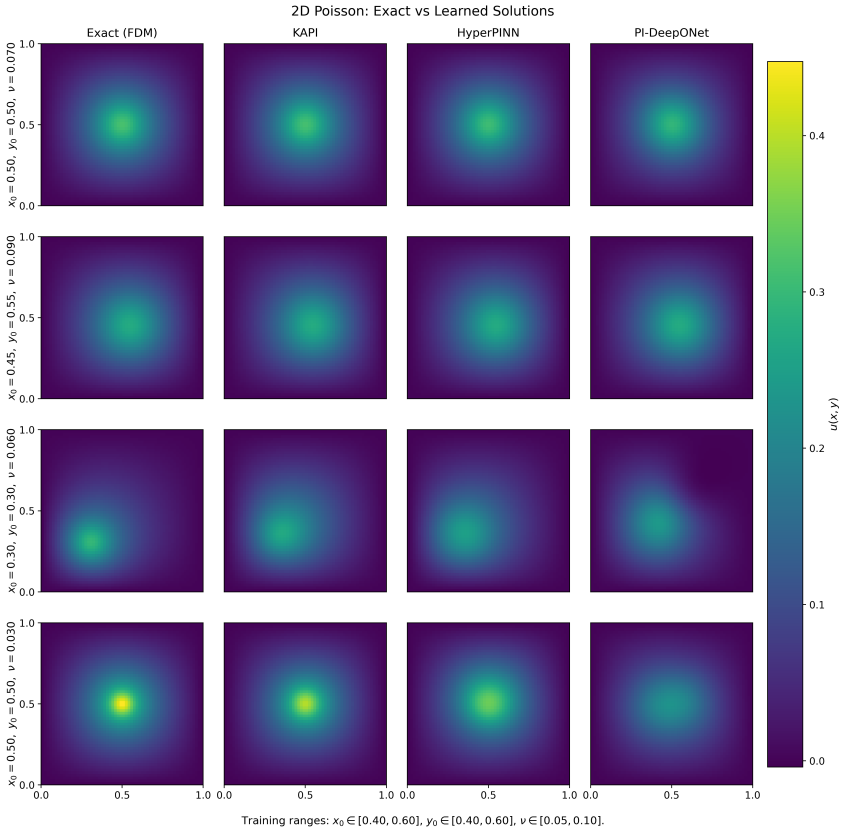

Figure 2: Exact and learned solutions for the parametric Poisson problem. Rows correspond to different parameter settings (x_0, y_0, ν) including in-distribution cases (top rows) and extrapolation cases (bottom rows). Columns show the finite-difference reference solution (FDM), the KAPI prediction, the FiLM–HyperPINN prediction, and the PI-DeepONet prediction. Training ranges are $x_0, y_0 \in [0.4, 0.6]$ and $\nu \in [0.05, 0.1]$.

- **Comparative scope and scaling.** The present study focuses on representative canonical PDE families to analyze structural behavior. Broader empirical comparisons across larger-scale benchmarks, additional baselines, and higher-dimensional settings would further clarify the method's practical scalability and relative strengths.
- **Lack of theoretical guarantees.** The work does not provide formal approximation error bounds or convergence guarantees for the learned meta-solution operator. Developing theoretical analysis for kernel-distribution meta-learning remains an open direction for future research.

4 Conclusion

We introduced a kernel-adaptive physics-informed meta-learning framework for parametric families of linear partial differential equations, in which solutions are represented as linear combinations of kernel functions with parameter-conditioned geometry and sparsity structure. Depending on the PDE family, kernel coefficients may be shared across tasks or generated by a lightweight meta-network as parameter-conditioned weights. By exploiting linearity, the approach yields a compact and interpretable solution representation that generalizes across parameter instances without per-instance retraining. Across representative Poisson, advection, and advection–diffusion problems, the method demonstrates consistent in-distribution accuracy and structured, geometry-aligned error growth under extrapolation. Comparative experiments against PI-DeepONet and FiLM–HyperPINN highlight how different conditioning strategies—separable operator decomposition, feature-wise

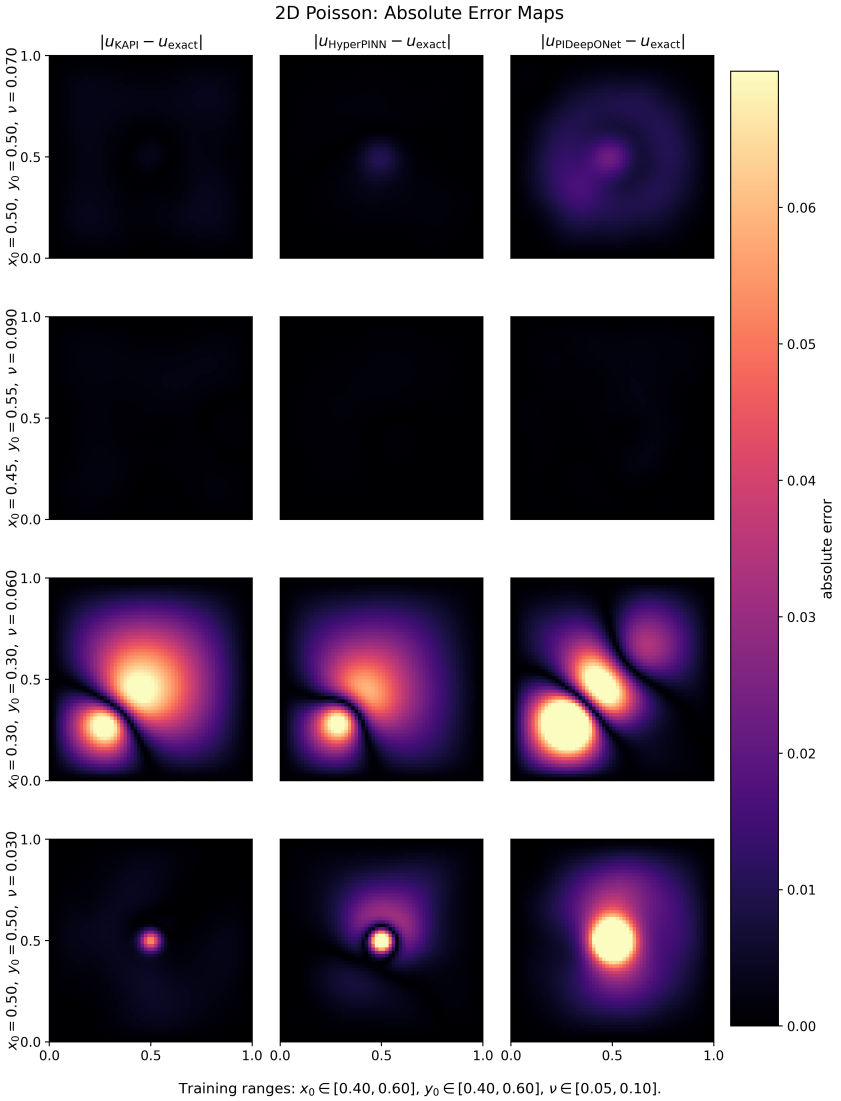

Figure 3: Absolute error maps for the same Poisson test cases shown in Fig. 2. Columns show $|u_{\text{KAPI}} - u_{\text{exact}}|$, $|u_{\text{FiLM-HyperPINN}} - u_{\text{exact}}|$, and $|u_{\text{PI-DeepONet}} - u_{\text{exact}}|$. Errors remain small for in-distribution parameters (top rows) and increase under spatial and stiffness extrapolation (bottom rows).

modulation, and adaptive kernel geometry—induce distinct inductive biases under parameter shift. In particular, explicit control of kernel bandwidth can provide advantages in stiffness-dominated regimes, while global feature modulation may better accommodate certain geometric shifts. These findings suggest that structuring the hypothesis class through adaptive kernel geometry offers a transparent framework for studying approximation and generalization in parametric linear PDE settings. Future work will explore nonlinear extensions, broader empirical benchmarking, and theoretical characterization of approximation guarantees.

REFERENCES

Filipe de Avila Belbute-Peres, Yi-fan Chen, and Fei Sha. Hyperpinn: Learning parameterized differential equations with physics-informed hypernetworks. *The symbiosis of deep learning and differential equations*, 690, 2021.

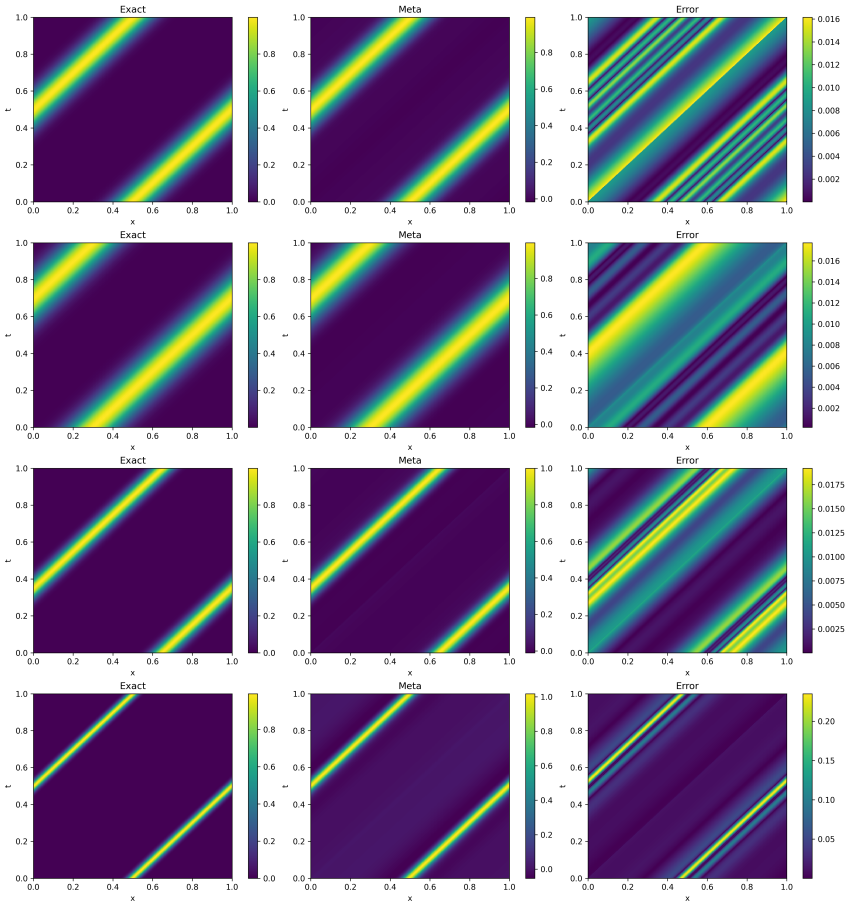

Figure 4: Advection equation results for multiple initial condition parameters (x_0, ν). Top rows show in-distribution cases and lower rows include increasingly stiff profiles. Columns display the exact solution, the meta-learned solution, and the absolute error.

Vikas Dwivedi, Nishant Parashar, and Balaji Srinivasan. Distributed learning machines for solving forward and inverse problems in partial differential equations. *Neurocomputing*, 420:299–316, 2021.

Vikas Dwivedi, Bruno Sixou, and Monica Sigovan. Curriculum learning-driven pielms for fluid flow simulations. *Neurocomputing*, 650:130924, 2025.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.

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

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, 2021a.

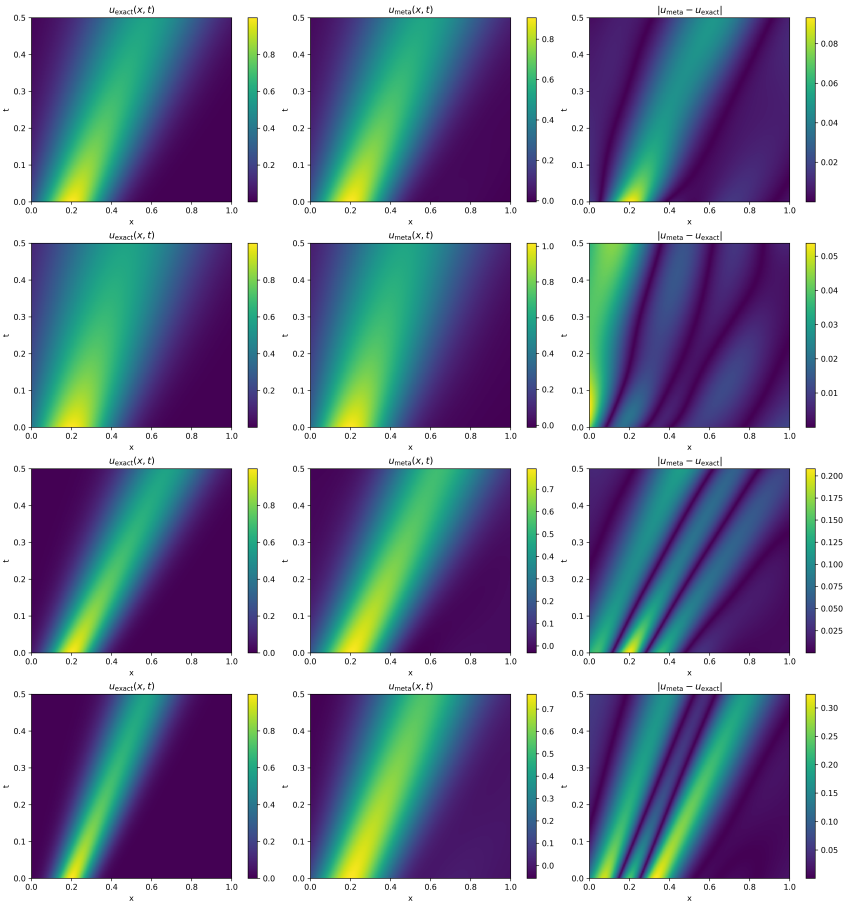

Figure 5: Advection–diffusion results for different parameter pairs (a, ν) spanning diffusion- and transport-dominated regimes. Columns show the exact solution, the meta-learned solution, and the absolute error.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM review*, 63(1):208–228, 2021b.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Tian Qin, Alex Beatson, Deniz Oktay, Nick McGreivy, and Ryan P Adams. Meta-pde: Learning to solve pdes quickly without a mesh. *arXiv preprint arXiv:2211.01604*, 2022.

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.

Amuthan A Ramabathiran and Prabhu Ramachandran. Spinn: Sparse, physics-based, and partially interpretable neural networks for pdes. *Journal of Computational Physics*, 445:110600, 2021.

A Implementation Details

This appendix summarizes the network architecture, kernel parameterization, and training procedure used across all experiments.

A.1 META-NETWORK ARCHITECTURE

The kernel geometry is generated by a lightweight meta-network $h(\lambda)$ that maps PDE parameters λ to kernel distribution parameters. The meta-network is implemented as a fully connected multilayer perceptron with:

- **Input dimension:** equal to the number of PDE parameters (3 for Poisson, 2 for advection and advection–diffusion),
- **Hidden layers:** two fully connected layers of width 64,
- **Activation function:** hyperbolic tangent ($\tanh$),
- **Output heads:** separate linear layers producing kernel centers $\mu_j(\lambda)$, log-widths $\log \sigma_j(\lambda)$, sparsity gates $g_j(\lambda)$, and (when applicable) kernel coefficients.

All input parameters are normalized to $[0, 1]$ prior to evaluation by the meta-network.

A.2 KERNEL DICTIONARY

Solutions are represented using a fixed-size dictionary of Gaussian radial basis functions:

- **Number of kernels:** $M = 128$ for the Poisson problem and $M = 64$ for the advection and advection–diffusion problems,
- **Kernel widths:** enforced positive using a softplus transformation with a small offset,
- **Kernel gates:** constrained to $(0, 1)$ via a sigmoid nonlinearity to promote sparsity,
- **Boundary handling:** homogeneous Dirichlet conditions for Poisson are enforced analytically using a multiplicative trial function.

For the advection equation, kernels are defined along the characteristic coordinate $s = (x - t) \bmod 1$, explicitly encoding transport geometry.

A.3 TRAINING PROCEDURE

All models are trained using physics-informed collocation without labeled solution data:

- **Optimizer:** Adam,
- **Learning rate:** 10^{-3},
- **Training epochs:** 2000 (Poisson) and 3000 (advection and advection–diffusion),
- **Tasks per batch:** 4 (Poisson) and 8 (advection-based problems),
- **Collocation points per task:**
  - Poisson: 2048 interior points and 256 boundary points,
  - Advection: 512 interior points, 128 initial-condition points, and 128 periodic-boundary points.

A curriculum strategy is employed in all experiments, where training begins with smoother regimes (larger diffusion parameters) and gradually expands to the full parameter range. A sparsity regularizer on kernel gates is applied with weight in the range 10^{-4}–10^{-5} to encourage compact kernel representations.

A.4 GROUND TRUTH COMPUTATION AND REFERENCE SOLUTIONS

To evaluate the accuracy of the learned meta-solution, we compare against high-fidelity reference solutions. The reference construction differs slightly across the three PDE families.

Poisson equation (finite-difference reference). For the elliptic Poisson problem

$$-\Delta u(x, y; \lambda) = f(x, y; \lambda), \qquad u|_{\partial\Omega} = 0,$$

with Gaussian source

$$f(x, y; \lambda) = \frac{1}{2\pi\nu^2} \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\nu^2}\right),$$

no closed-form solution is available on the bounded domain with homogeneous Dirichlet boundary conditions.

We therefore compute a numerical reference solution using a second-order finite-difference discretization on a uniform $N \times N$ grid over $\Omega = [0, 1]^2$. The discrete Laplacian is constructed via the standard five-point stencil,

$$\Delta_h u_{i,j} = \frac{1}{h^2} \left(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}\right),$$

leading to a linear system

$$Lu = -f,$$

where L is assembled using Kronecker products of 1D second-derivative matrices. The interior system is solved directly using a dense linear solver. This finite-difference solution serves as u_{exact} for error computation.

Advection equation (analytic solution). For the periodic advection equation

$$u_t + u_x = 0, \qquad u(x, 0) = u_0(x; \lambda), \qquad u(0, t) = u(1, t),$$

with Gaussian initial condition

$$u_0(x; \lambda) = \exp\left(-\frac{d(x, x_0)^2}{2\nu^2}\right),$$

where $d(x, x_0)$ denotes the periodic wrapped distance on $[0, 1]$, the equation admits a closed-form solution via characteristics:

$$u(x, t; \lambda) = u_0(x - t; \lambda).$$

In practice, the periodic distance is implemented using

$$d(x, x_0) = \big((x - x_0 + 0.5) \bmod 1\big) - 0.5,$$

ensuring exact periodicity. This analytic solution is used directly for computing relative L^2 errors.

Advection–diffusion equation (analytic solution). For the linear advection–diffusion equation

$$u_t + a\, u_x - \nu u_{xx} = 0, \qquad u(x, 0) = u_0(x; \lambda),$$

with Gaussian initial condition, the exact solution remains Gaussian due to linearity and constant coefficients. The solution is given by

$$u(x, t; \lambda) = \frac{\nu_0}{\sqrt{\nu_0^2 + 2\nu t}} \exp\left(-\frac{(x - at - x_0)^2}{2(\nu_0^2 + 2\nu t)}\right),$$

where ν_0 is the initial Gaussian width parameter.

Thus, advection shifts the center by at along characteristics, while diffusion increases the variance linearly in time. This closed-form expression is used as the ground-truth solution for quantitative evaluation.

Error metric. For all PDEs, the relative L^2 error is computed on a uniform evaluation grid:

$$\frac{\|u_{\text{meta}} - u_{\text{exact}}\|_2}{\|u_{\text{exact}}\|_2}.$$

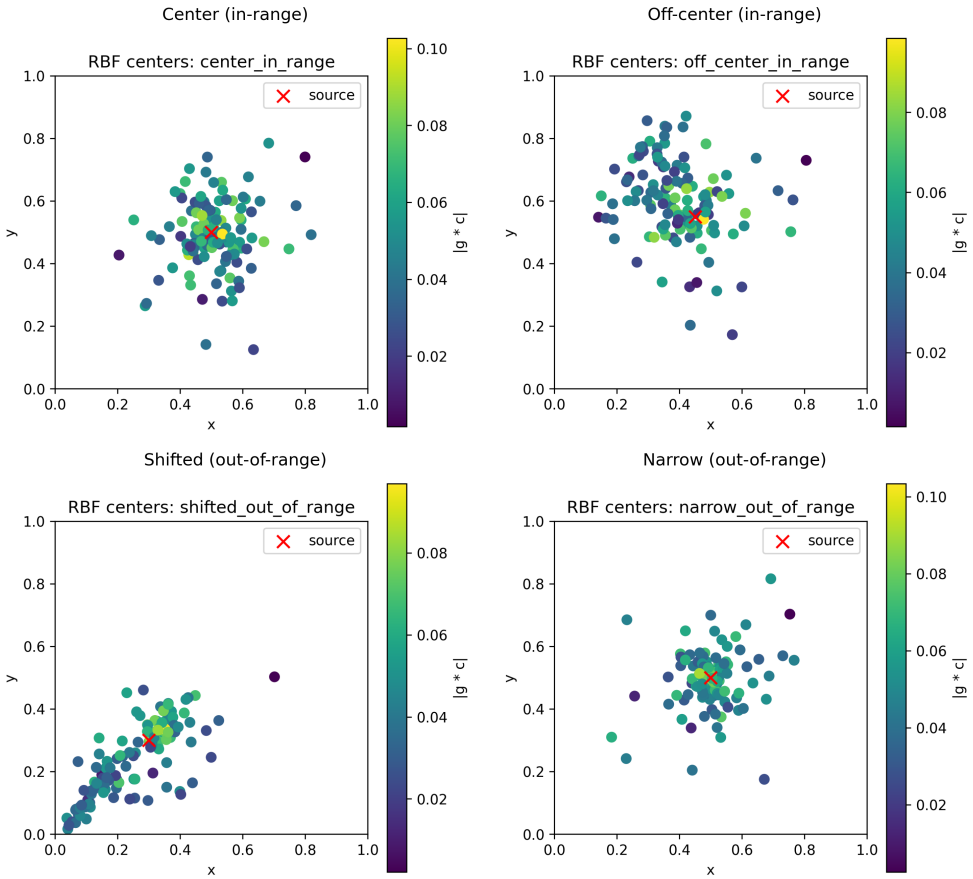

Figure 6: Learned kernel distributions for Poisson test cases. Each panel shows the inferred RBF centers (dots) with color indicating effective contribution magnitude $|g_j(\lambda)c_j|$; the red $\times$ marks the Gaussian source location. In in-range cases, high-contribution kernels concentrate near the source region, supporting accurate reconstruction. Under spatial extrapolation (shifted out-of-range), the kernel mass remains biased toward the training manifold, explaining the observed degradation. In stiffness extrapolation (narrow out-of-range), the model increases locality around the source but remains limited by the finite kernel resolution.

B Interpreting meta-generalization via learned kernel distributions (Poisson)

To better understand why the proposed meta-solution generalizes (and when it fails), we visualize the learned kernel distributions for representative Poisson test cases in Fig. 6. Each dot corresponds to a learned RBF center $\mu_j(\lambda)$, and the color encodes the effective contribution $|g_j(\lambda)c_j|$. In-distribution cases show that the meta-network consistently allocates high-contribution kernels around the active source region, yielding an interpretable "local Green's-function-like" response. In contrast, for the spatially shifted out-of-range case, the distribution of influential kernels remains concentrated near the training support rather than following the shifted source, which aligns with the elevated error reported in Table 1. For stiffness extrapolation (narrow source), the kernel mass becomes more localized but remains constrained by the finite bandwidth of the learned dictionary, providing a geometric explanation for resolution-limited degradation.

C COMPARISON BETWEEN KAPI META-LEARNING, PI-DEEPONET, AND FILM–HYPERPINN

We compare the proposed Kernel-Adaptive Physics-Informed (KAPI) meta-learning framework against two parameter-conditioned neural baselines: Physics-Informed DeepONet (PI-DeepONet) and a FiLM-conditioned HyperPINN. All models are trained in a label-free physics-informed manner on the same parametric Poisson family

$$-\Delta u(x, y; p) = f(x, y; p), \qquad u|_{\partial\Omega} = 0, \qquad p = (x_0, y_0, \nu),$$

with training ranges $x_0, y_0 \in [0.4, 0.6]$ and $\nu \in [0.05, 0.1]$. Collocation budgets and training protocols are kept comparable. The goal is not exhaustive hyperparameter tuning, but structural comparison of conditioning mechanisms under interpolation and extrapolation.

Interpolation. Within the training distribution, all three models achieve relative L^2 errors on the order of $\mathcal{O}(10^{-2})$. As shown in Fig. 2, predicted solutions are visually indistinguishable from the finite-difference reference in this regime, and the corresponding error maps in Fig. 3 remain uniformly small.

Geometric extrapolation (shifted center). When the Gaussian source is shifted outside the training region (e.g., $(x_0, y_0) = (0.3, 0.3)$), all methods degrade to $\mathcal{O}(10^{-1})$ error:

$$\text{KAPI: } \sim 3.6 \times 10^{-1}, \quad \text{PI-DeepONet: } \sim 4.1 \times 10^{-1}, \quad \text{FiLM–HyperPINN: } \sim 2.7 \times 10^{-1}.$$

FiLM–HyperPINN exhibits the lowest error in this spatial extrapolation test. The differing behavior reflects architectural structure: PI-DeepONet recombines fixed spatial modes, FiLM–HyperPINN modulates global features, while KAPI must realize spatial shifts through movement of kernel centers. The structured error patterns in Fig. 3 are consistent with these distinct conditioning mechanisms.

Scale extrapolation (narrow Gaussian). For ν below the training range (e.g., $\nu = 0.03$), sharper localized features must be resolved:

$$\text{KAPI: } \sim 3.9 \times 10^{-2}, \quad \text{PI-DeepONet: } \sim 2.7 \times 10^{-1}, \quad \text{FiLM–HyperPINN: } \sim 1.0 \times 10^{-1}.$$

Here, KAPI achieves the lowest error. While FiLM substantially improves over PI-DeepONet, its performance degrades more than KAPI as localization increases. This is consistent with the fact that KAPI can explicitly reduce kernel widths $\sigma_j(p)$, increasing spatial resolution in a controlled manner. In contrast, FiLM modifies feature responses of a fixed trunk network and does not explicitly adapt spatial bandwidth.

Structural summary.

Property	**KAPI**	**PI-DeepONet**	**FiLM–HyperPINN**
Representation	Adaptive kernel expansion	Separable branch–trunk	Conditioned trunk MLP
Parameter conditioning	Kernel geometry (μ, σ, g)	Branch-to-trunk coupling	Feature-wise affine modulation
Spatial basis	Explicit, movable	Fixed global modes	Implicit modulated features
Shifted-center OOD	$\sim 3.6 \times 10^{-1}$	$\sim 4.1 \times 10^{-1}$	$\sim 2.7 \times 10^{-1}$
Small-ν OOD	$\sim 3.9 \times 10^{-2}$	$\sim 2.7 \times 10^{-1}$	$\sim 1.0 \times 10^{-1}$
Interpretability	High (explicit kernels)	Moderate	Moderate

Model size. PI-DeepONet uses separate 3-layer branch and trunk networks (width 128), totaling $\sim 1.0 \times 10^5$ parameters. FiLM–HyperPINN uses a 4-layer trunk (width 128) and a 2-layer hypernetwork (width 64), totaling $\sim 1.2 \times 10^5$ parameters. KAPI employs a 2-layer meta-network (width 64) generating $M = 128$ adaptive kernels with shared coefficients, totaling $\sim 4.1 \times 10^4$ parameters. Thus, KAPI attains competitive or improved performance in certain regimes with substantially fewer parameters.

Discussion. All three physics-informed models generalize well within the training distribution. Under spatial extrapolation, FiLM–HyperPINN performs best in this experiment, suggesting that global feature modulation provides flexibility for geometric deformation. Under stiffness-driven scale extrapolation, KAPI performs best, indicating that explicit control of kernel bandwidth can be advantageous for resolving highly localized structures. These results illustrate how different conditioning strategies—separable operator decomposition, feature-wise modulation, and adaptive kernel geometry—induce distinct inductive biases under parameter shift.