IMPLICIT LEGENDRE REPRESENTATION FOR PHYSICS-INFORMED NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-informed neural networks (PINNs) commonly employ Fourier-based representations to more accurately approximate high-frequency components in solutions to partial differential equations (PDEs). In this work, we challenge the need for spectral priors in representation design and propose Legendre representation networks (LERENs), which integrate learnable Legendre expansions into multilayer perceptrons (MLPs). We evaluate LERENs on five challenging PDE problems involving high-frequency components, complex solution structures, and high-dimensional domains. Experimental results demonstrate that LERENs outperform state-of-the-art Fourier-based representation methods on the evaluated benchmarks, establishing Legendre-based representations as a competitive alternative to Fourier-based representations in PINNs. The proposed representation paradigm may extend to other bases used in spectral methods, allowing their favorable mathematical properties to inform the design of neural representations.

1 INTRODUCTION

Partial differential equations (PDEs) provide a mathematical foundation for understanding physical systems across fluid mechanics (Raissi et al., 2020), geophysics (Virieux & Operto, 2009), and quantum mechanics (Pfau et al., 2020). Their numerical solution is therefore central to scientific modeling and prediction. A broad range of tools has been developed for this purpose, from classical finite difference and finite element methods (LeVeque, 2007; Ciarlet, 1978) to learning-based approaches such as physics-informed neural networks (PINNs) (Raissi et al., 2019) and neural operators (Lu et al., 2021; Li et al., 2021; Kovachki et al., 2023). Among these approaches, PINNs are particularly attractive for their mesh-free formulation and unified treatment of forward and inverse problems. Guided by governing equations, PINNs can approximate solutions from limited observations and rely less on solution labels (Raissi et al., 2019; Karniadakis et al., 2021).

Despite these advantages, spectral bias poses a major challenge for PINNs in learning high-frequency PDE solutions. During gradient-based training, neural networks tend to learn low-frequency components before high-frequency ones (Rahaman et al., 2019; Wang et al., 2021b; Xu et al., 2020). As a result, a network may capture the large-scale structure of a solution without adequately resolving its rapid oscillations. To address this challenge, Fourier-based representations offer a natural response by incorporating periodic functions directly into the network. Fourier embeddings map input coordinates to sine and cosine features before passing them to an MLP (Mildenhall et al., 2020; Tancik et al., 2020). This input encoding helps the network resolve high-frequency variations in PDE solutions (Wang et al., 2021b; Song & Wang, 2023). Within implicit neural representations, SIREN uses sinusoidal hidden-layer activations to capture fine signal details and their derivatives with respect to the input coordinates (Sitzmann et al., 2020).

However, the effectiveness of Fourier-based representations can be sensitive to frequency selection, embedding dimension, and initialization. Even when high-frequency features are available, optimization may remain difficult for the corresponding PDE problem (Tancik et al., 2020; Benbarka et al., 2022; Liu et al., 2024; Novello et al., 2025; Wu et al., 2026). These limitations have motivated methods that control the represented frequencies, adapt frequency selection, or modify the training procedure (Lindell et al., 2022; Liu et al., 2024; Novello et al., 2025; Liu et al., 2025; Wu et al., 2026). Although these methods differ in how they construct and train the network, they retain Fourier-based representations as the basis for high-frequency approximation.

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.

This raises a central question: can PINNs accurately learn high-frequency PDE solutions without Fourier embeddings or sinusoidal activations?

In sinusoidal representation networks (SIRENs), composition generates frequencies that are integer combinations of the input frequencies (Novello et al., 2025). This property motivates us to examine whether composition can also support high-frequency approximation without sinusoidal activations. We develop this idea using neural representations based on Legendre polynomials, whose nested compositions can generate higher-degree components. The resulting Legendre representation networks (LERENs) integrate learnable Legendre expansions into multilayer perceptrons (MLPs) to approximate high-frequency PDE solutions.

Our main contributions are as follows:

- We extend the interpretation of sinusoidal networks as implicit spectral representations to Legendre representations through nested polynomial composition. Functions with properties like those of the Legendre basis may also be used to build neural representations.
- We propose LERENs, a neural representation built from learnable Legendre expansions that can approximate high-frequency PDE solutions without Fourier embeddings at the input or sinusoidal activations in the hidden layers.
- We evaluate Legendre representations on convection, Burgers, high-frequency Helmholtz, five-dimensional heat conduction (Heat-5D), and nonlinear diffusion. The results demonstrate accurate approximation on Fourier-friendly benchmarks with sinusoidal solutions or periodic structure. They also show that Legendre representations effectively approximate non-sinusoidal solutions in high-dimensional heat conduction and nonlinear diffusion.

2 PHYSICS-INFORMED PROBLEM FORMULATION

In forward problems, we seek the solution field u for a prescribed differential equation and auxiliary conditions. Let $\mathcal{D}$ denote the computational domain and $\mathbf{x}$ collect all input coordinates, including time for time-dependent problems. A common formulation for the problems studied here is

$$\begin{aligned} \mathcal{N}[u](\mathbf{x}) &= f(\mathbf{x}), \quad \mathbf{x} \in \mathcal{D}, \\ \mathcal{B}[u](\mathbf{x}) &= g_b(\mathbf{x}), \quad \mathbf{x} \in \Gamma_b, \\ u(\mathbf{x}) &= g_0(\mathbf{x}), \quad \mathbf{x} \in \Gamma_0, \end{aligned} \tag{1}$$

where $\mathcal{N}$ is the governing differential operator, f is a known source, $\mathcal{B}$ specifies the boundary conditions on Γ_b, and Γ_0 is the initial surface when time is present. Periodic conditions are represented through differences between paired boundary values and, where required, their derivatives. For steady problems, the initial condition is omitted.

A physics-informed neural network (PINN) represents the unknown field by a sufficiently differentiable function $u_\theta(\mathbf{x})$ and determines its trainable parameters θ by penalizing violations of these equations (Wang et al., 2023; Rathore et al., 2024). Define

$$\begin{aligned} r_r(\mathbf{x};\theta) &= \mathcal{N}[u_\theta](\mathbf{x}) - f(\mathbf{x}), \\ r_b(\mathbf{x};\theta) &= \mathcal{B}[u_\theta](\mathbf{x}) - g_b(\mathbf{x}), \\ r_0(\mathbf{x};\theta) &= u_\theta(\mathbf{x}) - g_0(\mathbf{x}), \end{aligned} \tag{2}$$

on their respective domains. The derivatives required by the PDE and boundary operators are evaluated by automatic differentiation (Wang et al., 2023; 2025; Baydin et al., 2018). Collocation points specify where constraints are evaluated; this forward formulation requires no interior solution labels.

For point sets $\mathcal{X}_k = \{\mathbf{x}_{k,i}\}_{i=1}^{N_k}$, the basic training objective is

$$\min_\theta \mathcal{L}(\theta), \qquad \mathcal{L}(\theta) = \sum_{k \in \mathcal{K}} \lambda_k \mathcal{L}_k(\theta), \tag{3}$$

with

$$\mathcal{L}_k(\theta) = \frac{1}{N_k} \sum_{i=1}^{N_k} \|r_k(\mathbf{x}_{k,i};\theta)\|_2^2 . \tag{4}$$

Here, $\mathcal{K} \subseteq \{r, b, 0\}$ indexes the applicable residual, boundary, and initial terms, and $\lambda_k > 0$ are loss weights. A condition satisfied exactly by the parameterization needs no penalty (Sukumar & Srivastava, 2022). The sampling and weighting strategies can be chosen for the individual PDE problem (Wang et al., 2023; Wu et al., 2023; Wang et al., 2021a).

This objective penalizes constraint violations at sampled locations, while solution accuracy must be assessed separately. Recent studies examine ill-conditioning associated with differential operators (Rathore et al., 2024) and conflicts between the gradients of different loss terms (Wang et al., 2025). Within this framework, our method specifies u_θ through learnable Legendre expansions; the governing PDE and its auxiliary conditions continue to define the training constraints.

3 From Sinusoidal Composition to Legendre Representations

Sinusoidal representation networks (SIRENs) represent a function through compositions of affine transformations and sinusoidal activations (Sitzmann et al., 2020). With fixed frequency scales absorbed into the affine parameters, a scalar-valued SIREN takes the form

$$\mathbf{h}^{(0)} = \mathbf{x}, \tag{5}$$

$$\mathbf{h}^{(\ell)} = \sin\Big(\mathbf{W}_\ell \mathbf{h}^{(\ell-1)} + \mathbf{b}_\ell\Big), \tag{6}$$

$$f_\theta(\mathbf{x}) = \mathbf{a}^\top \mathbf{h}^{(L)} + b_{\text{out}}. \tag{7}$$

TUNER relates this compositional representation to a Fourier-type expansion (Novello et al., 2025). In particular, a neuron in the second hidden layer admits the exact representation

$$H_i(\mathbf{x}) = \sin\left(\sum_{j=1}^{m} W_{ij} \sin(\boldsymbol{\omega}_j^\top \mathbf{x} + \varphi_j) + b_i\right) \tag{8}$$

$$= \sum_{\mathbf{k}\in\mathbb{Z}^m} A_{i,\mathbf{k}} \sin\big(\boldsymbol{\xi}_\mathbf{k}^\top \mathbf{x} + \eta_{i,\mathbf{k}}\big), \tag{9}$$

$$\boldsymbol{\xi}_\mathbf{k} = \sum_{j=1}^{m} k_j \boldsymbol{\omega}_j. \tag{10}$$

Here, the coefficients $A_{i,\mathbf{k}}$ depend on the hidden weights and $\eta_{i,\mathbf{k}} = b_i + \sum_j k_j \varphi_j$. A finite truncation yields a Fourier-type spectral approximation. This interpretation exposes a property of sinusoidal composition: low input frequencies can combine to generate higher-frequency components. A sinusoidal network can therefore be viewed as an implicit spectral representation whose frequencies are integer combinations of the input frequencies. This motivates us to examine whether other spectral approximation bases admit a similar mechanism for generating higher-order modes.

Legendre polynomials exhibit such a compositional property. Since P_n has degree n and $\{P_0, \ldots, P_N\}$ spans all polynomials of degree at most N, for $m, n \geq 2$ we have

$$P_m(P_n(q)) = \sum_{r=0}^{mn} \gamma_r^{m,n} P_r(q), \qquad \gamma_{mn}^{m,n} \neq 0. \tag{11}$$

The leading term is nonzero because the composition has degree exactly mn. Thus, composing lower-degree Legendre modes introduces higher-degree components. Direct substitution gives

$$P_2(P_2(q)) = -\frac{1}{5}P_0(q) + \frac{3}{7}P_2(q) + \frac{27}{35}P_4(q). \tag{12}$$

Two quadratic components generate a fourth-degree mode. This provides a polynomial counterpart to the frequency-generation mechanism in sinusoidal networks.

Based on this observation, we construct Legendre representation networks (LERENs) by composing learnable Legendre expansions. Figure 1 shows the architecture. Each hidden layer evaluates a finite Legendre expansion in learned bounded coordinates. Starting from $\mathbf{h}^{(0)} = \mathbf{x}$, we define

$$\mathbf{q}^{(\ell)} = \tanh\Big[\alpha_\ell \left(\mathbf{W}_\ell \mathbf{h}^{(\ell-1)} + \mathbf{b}_\ell\right)\Big], \tag{13}$$

$$\mathbf{h}^{(\ell)} = \sum_{n \in \mathcal{S}_\ell} c_{\ell n} P_n\Big(\mathbf{q}^{(\ell)}\Big), \tag{14}$$

where P_n is applied elementwise, $\mathcal{S}_\ell$ specifies the selected degrees, and $c_{\ell n}$ and α_ℓ are trainable scalars. The coefficients determine the modal mixture within each layer, while composition allows subsequent layers to act on the representations constructed by preceding layers. The $\tanh$ map places each expansion coordinate in $(-1, 1)$. Equation (11) motivates compositional mode generation. We sum the equal-width hidden outputs and apply a shared linear readout,

$$u_\theta(\mathbf{x}) = b_{\text{out}} + \mathbf{v}^\top \sum_{\ell=1}^{L} \mathbf{h}^{(\ell)}(\mathbf{x}). \tag{15}$$

Here, $\mathbf{v} \in \mathbb{R}^m$ and $b_{\text{out}} \in \mathbb{R}$ are the shared readout weights and output bias, respectively, and m denotes the hidden width. The hidden representations propagate sequentially through the network and jointly contribute to the final readout. Our default architecture contains four hidden layers of width 256, with P_0–P_9 in the first layer and P_0, P_1, P_8, P_9 in the remaining three layers. This configuration introduces 22 trainable expansion coefficients, which are jointly optimized with the remaining network parameters using the physics-informed objective defined in Section 2.

3.1 Controlling the Initial Response of Legendre Layers

The coefficients of a Legendre expansion determine both the shape and the local sensitivity of the resulting activation. In the first layer, we seek to retain multiple polynomial degrees while matching the local response of the underlying $\tanh$ map at the origin. We therefore choose the initial scalar activation to vanish at the origin and have unit slope there. This provides a reference scale for the first-layer response while retaining higher-degree components from the start of training.

To construct such an activation, we initialize the even-degree coefficients to zero and assign a common value c to the five odd-degree coefficients, denoting coefficient initial values by $c_{\ell n}^{(0)}$. With $\alpha_1^{(0)} = 1$, the resulting activation is

$$\phi_{1,c}^{(0)}(z) = c \sum_{k=0}^{4} P_{2k+1}(\tanh z), \tag{16}$$

where z denotes the scalar input to the activation after the affine transformation. Odd symmetry gives $\phi_{1,c}^{(0)}(0) = 0$. Since

$$\big(\phi_{1,c}^{(0)}\big)'(0) = c \sum_{k=0}^{4} P'_{2k+1}(0) = c\frac{211}{128}, \tag{17}$$

the unit-slope condition yields $c = 128/211$. The first-layer coefficients are therefore initialized as

$$c_{1n}^{(0)} = \begin{cases} 128/211, & n \in \{1, 3, 5, 7, 9\}, \\ 0, & n \in \{0, 2, 4, 6, 8\}. \end{cases} \tag{18}$$

Consequently, $\phi_1^{(0)}(z) = z + \mathcal{O}(z^3)$ near the origin. The first-layer activation thus has the same first-order response as $\tanh$, while its higher-degree terms remain present in the initial representation.

For the remaining layers, we use an equal-weight mixture of P_1 and P_9, initializing the coefficients of P_0, P_1, P_8, P_9 to $(0, 1/2, 0, 1/2)$. This yields

$$\phi_\ell^{(0)}(z) = \frac{1}{2} P_1(\tanh z) + \frac{1}{2} P_9(\tanh z), \qquad \ell = 2, 3, 4. \tag{19}$$

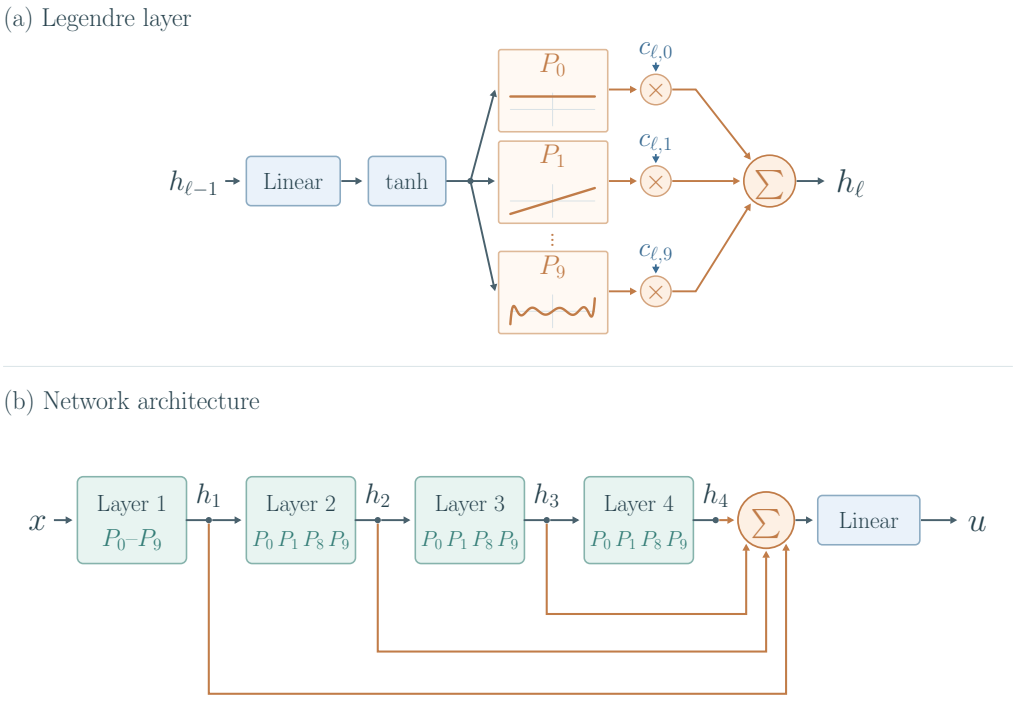


Figure 1: Architecture of a Legendre representation network (LEREN). (a) Each layer applies an affine transformation and $\tanh$, then combines elementwise Legendre polynomial evaluations using trainable coefficients $c_{\ell n}$. (b) The first layer uses P_0–P_9, and the remaining three layers use P_0, P_1, P_8, P_9. The layers are connected sequentially, and all four hidden outputs are summed before a shared linear readout produces u.

This gives each layer both low- and high-degree components at initialization. All selected coefficients, including those initialized to zero, remain trainable, allowing the initial mixture to adapt during optimization. The initialization ablation on the convection benchmark in Figure 5 provides empirical support for the first-layer coefficient choice.

3.2 Universal Approximation

Proposition 1 (Universal approximation)**.** *Let $K \subset \mathbb{R}^d$ be compact. Fix a depth $L \geq 1$ and degree sets with $1 \in \mathcal{S}_1$. For LEREN defined by Equations (13)–(15) with unrestricted trainable coefficients, every $f \in C(K)$ and $\varepsilon > 0$ admit a common hidden width m and parameters θ such that*

$$\sup_{\mathbf{x}\in K} |u_\theta(\mathbf{x}) - f(\mathbf{x})| < \varepsilon. \tag{20}$$

Proof. Set $\alpha_1 = 1$, $c_{11} = 1$, and all other expansion coefficients to zero. Since $P_1(t) = t$, this gives $\mathbf{h}^{(1)}(\mathbf{x}) = \tanh(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$ and $\mathbf{h}^{(\ell)}(\mathbf{x}) = \mathbf{0}$ for $\ell \geq 2$. The shared readout therefore becomes

$$u_\theta(\mathbf{x}) = b_{\text{out}} + \sum_{j=1}^{m} v_j \tanh\left(\mathbf{w}_j^\top \mathbf{x} + b_{1j}\right), \tag{21}$$

where $\mathbf{w}_j^\top$ is the jth row of $\mathbf{W}_1$. This is an arbitrary single-hidden-layer $\tanh$ network. Since $\tanh$ is continuous, bounded, and nonconstant, these networks are dense in $C(K)$ under the uniform norm (Hornik, 1991, Theorem 2). Choosing a suitable width and parameters proves the claim. □

The depth and selected polynomial degrees can remain fixed as width increases. The proof's coefficient choice embeds the $\tanh$ model class; the coefficients remain trainable under our initialization.

4 EXPERIMENTAL EVALUATION

Our experiments examine two aspects of the proposed method. First, we evaluate its competitiveness on PDEs with sinusoidal or periodic structure, where Fourier representations are a natural choice but can still be difficult to optimize. Second, we assess whether its effectiveness extends to problems with non-sinusoidal solutions, including high-dimensional heat conduction and nonlinear diffusion.

We evaluate prediction accuracy using the relative L_2 error (RL2), defined as

$$\mathrm{RL2} = \frac{\|\widehat{\mathbf{u}} - \mathbf{u}_{\mathrm{ref}}\|_2}{\|\mathbf{u}_{\mathrm{ref}}\|_2} = \left(\frac{\sum_{i=1}^{N} |\widehat{u}_i - u_{\mathrm{ref},i}|^2}{\sum_{i=1}^{N} |u_{\mathrm{ref},i}|^2} \right)^{1/2}, \tag{22}$$

where $\widehat{\mathbf{u}}$ and $\mathbf{u}_{\mathrm{ref}}$ contain predicted and reference values at the same N evaluation points, respectively, with entries $\widehat{u}_i$ and $u_{\mathrm{ref},i}$ at point i. Benchmark-specific model variants, PDE definitions, compared methods, and computational costs are detailed in Appendices A and C–E.

Our experiments use auxiliary training components from the JAX-PI framework (Wang et al., 2023). Random weight factorization introduces trainable scaling factors for neuron weights to facilitate optimization (Wang et al., 2022). Gradient-norm-based adaptive loss balancing rescales individual loss terms to balance their contributions to parameter updates. Causal weighting prioritizes earlier-time residual minimization, helping training respect the solution's temporal order (Wang et al., 2024b).

4.1 COMPETITIVE PERFORMANCE ON FOURIER-FRIENDLY BENCHMARKS

We consider three challenging benchmarks: convection, Burgers, and high-frequency Helmholtz. These problems have sinusoidal solutions or periodic structure that makes Fourier-based representations a natural choice. We evaluate whether Legendre representations can achieve competitive accuracy without knowing the solution frequencies or substantially changing the MLP architecture.

The baselines include JAX-PI configurations with Fourier and periodic embeddings, SIREN, IFeF-PINN, and PirateNet. The convection and Burgers problems follow the JAX-PI benchmark formulations, while the high-frequency Helmholtz problem follows the setting studied in IFeF-PINN.

Convection and Burgers equations. The convection benchmark has the analytical solution $u(x,t) = \sin(x - ct)$. Following JAX-PI, we use a convection speed of $c = 80$ and the initial condition $u(x,0) = \sin x$. The temporal derivative is scaled by c relative to the spatial derivative, creating a substantial difference between the temporal and spatial variation of the solution. This makes the problem challenging to optimize despite its simple sinusoidal form. The viscous Burgers equation provides a complementary test: nonlinear advection and low viscosity produce steep solution gradients that are difficult to resolve accurately. The following table compares combinations of Fourier embeddings, periodic embeddings, and the Modified MLP architecture.

Table 1: Relative L_2 errors on the convection and Burgers benchmarks for different combinations of Fourier embeddings, periodic embeddings, and the Modified MLP architecture. A check mark indicates that a component is used, and a cross indicates that it is absent. Lower errors indicate closer agreement between predictions and reference solutions.

Method	Fourier embedding	Periodic embedding	Modified MLP	Convection RL2	Burgers RL2
JAX-PI (1)	×	✓	✓	4.30×10^{-1}	7.42×10^{-4}
JAX-PI (2)	✓	×	✓	7.38×10^{-1}	8.50×10^{-4}
JAX-PI (3)	✓	✓	×	1.02×10^{-2}	1.21×10^{-3}
JAX-PI (4)	✓	✓	✓	6.88×10^{-4}	3.58×10^{-4}
JAX-PI + SIREN	×	✓	×	1.88×10^{-1}	1.09×10^{-3}
PirateNet	✓	✓	✓	5.48×10^{-4}	8.20×10^{-5}
Ours	×	×	×	$\mathbf{9.33 \times 10^{-5}}$	$\mathbf{4.49 \times 10^{-5}}$

Our method achieves the lowest error among the compared methods on both problems. It obtains relative L_2 errors of 9.33×10^{-5} on convection and 4.49×10^{-5} on Burgers, outperforming the

full JAX-PI configuration and PirateNet. These results show that an MLP equipped with Legendre representations can accurately solve both convection and Burgers without using either Fourier embeddings or periodic embeddings at the input.

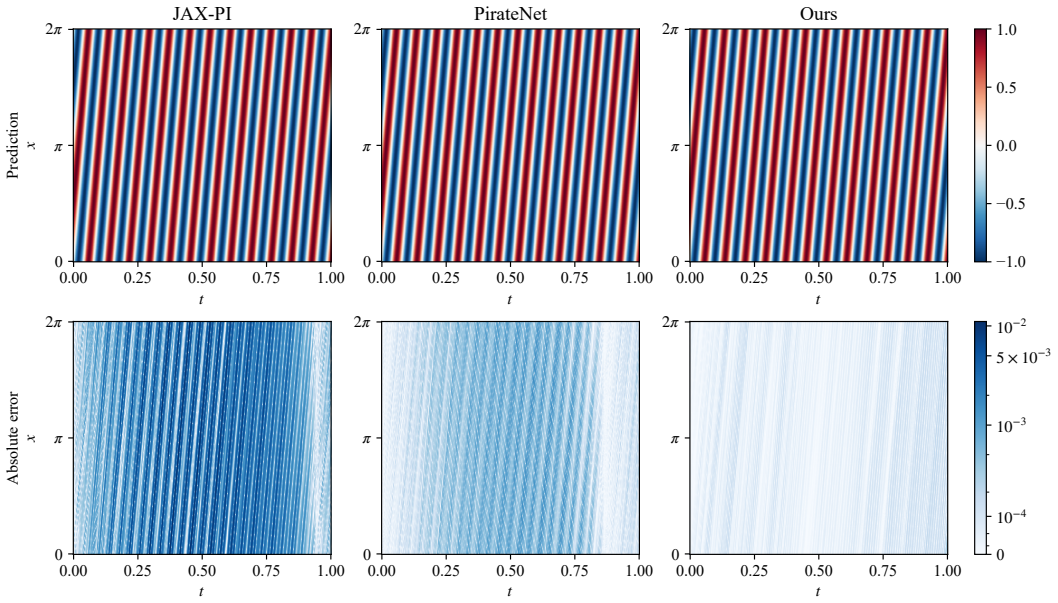


Figure 2: Predicted solutions (top) and pointwise absolute errors $|\widehat{u} - u_{\mathrm{ref}}|$ (bottom) for the convection equation with speed $c = 80$. Columns show JAX-PI, PirateNet, and our method from left to right. The horizontal and vertical axes denote time t and position x, respectively. The color scales are shared across methods within each row.

High-frequency Helmholtz equation. We consider the analytical solution

$$u(x, y) = \sin(a_1 \pi x) \sin(a_2 \pi y), \qquad a_1 = a_2 = 100. \tag{23}$$

The solution oscillates rapidly along both spatial coordinates, making it challenging for neural networks to approximate. Although its sinusoidal form is naturally compatible with Fourier representations, this compatibility does not guarantee successful optimization.

Table 2: Relative L_2 errors on the high-frequency Helmholtz benchmark with $a_1 = a_2 = 100$. Lower errors indicate better accuracy. "Did not converge" indicates convergence failure.

Method	Vanilla PINN	JAX-PI	SIREN	PirateNet	IFeF-PINN	Ours
RL2	Did not converge	1.04×10^{-2}	1.64×10^{-2}	Did not converge	9.20×10^{-3}	$\mathbf{7.10 \times 10^{-3}}$

IFeF-PINN uses alternating bi-level optimization to separate feature learning from output-coefficient estimation; its end-to-end ablation fails to converge in this setting. We jointly train all parameters in one Adam stage, with relative L_2 error 7.10×10^{-3} versus 9.20×10^{-3} for IFeF-PINN. We resolve the oscillatory solution without explicit Fourier features or separate coefficient fitting.

4.2 Effectiveness on non-sinusoidal benchmarks

We further evaluate our method on Heat-5D and nonlinear diffusion. These benchmarks test two different challenges: approximation in a higher-dimensional input domain and solution-dependent diffusion dynamics. They also allow us to assess whether the effectiveness of Legendre representations extends beyond explicitly sinusoidal solution profiles. The comparison includes vanilla PINNs, JAX-PI, PirateNet, IFeF-PINN, and ProPINN.

Heat-5D. Heat-5D is a heat-conduction benchmark from PINNacle, defined on the five-dimensional unit ball with an additional temporal coordinate. It tests approximation accuracy with five spatial

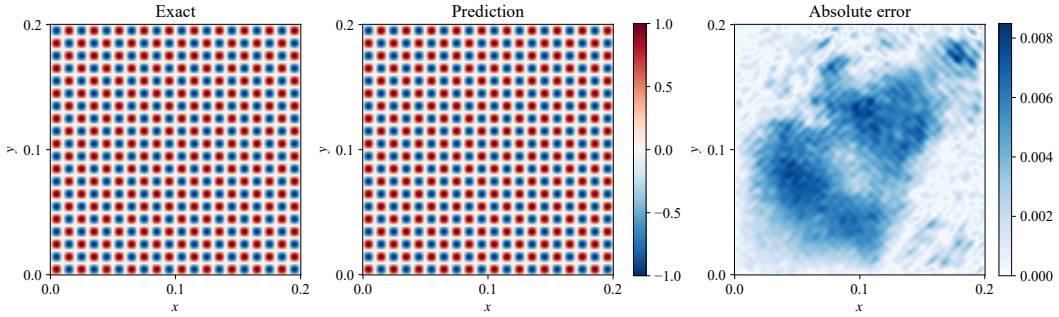


Figure 3: Exact solution (left), our prediction (middle), and pointwise absolute error (right) for the high-frequency Helmholtz benchmark with $a_1 = a_2 = 100$. The displayed domain is $[0, 0.2]^2$.

Table 3: Relative L_2 errors on Heat-5D and nonlinear diffusion. Lower errors indicate better accuracy. The nonnumeric ProPINN entry for Heat-5D indicates its neighborhood-size limitation.

Method	Year	Venue	Heat-5D RL2	Diffusion RL2
Vanilla PINN	2019	JCP	1.53×10^{-2}	6.58×10^{-2}
JAX-PI	2023	arXiv	2.09×10^{-4}	4.73×10^{-3}
PirateNet	2024	JMLR	1.31×10^{-3}	2.02×10^{-2}
IFeF-PINN	2026	ICLR	2.31×10^{-2}	2.08×10^{-2}
ProPINN	2026	TMLR	Curse of dimensionality	1.54×10^{-2}
Ours	—	—	$\mathbf{2.06 \times 10^{-4}}$	$\mathbf{3.20 \times 10^{-3}}$

inputs and one time input. The larger input dimension makes sampling the domain and computing the required derivatives more demanding. Because an analytical solution is available, reference values can be evaluated directly without a numerical PDE solver.

ProPINN aggregates information from local neighborhoods of input points. Directly extending its Cartesian perturbation grid to six input coordinates causes the neighborhood size to grow exponentially, resulting in the curse of dimensionality. Our method processes the input coordinates through an MLP and avoids this growth in neighborhood size. It achieves a relative L_2 error of 2.06×10^{-4}, the lowest among the compared methods, demonstrating its effectiveness on this benchmark.

Nonlinear diffusion. This benchmark describes diffusion with an effective diffusivity that depends on the solution itself. Consequently, the rate of diffusion varies across the domain as the solution evolves, making the problem more challenging than linear diffusion. Following the CoPINN benchmark, we use reference solutions generated numerically with the finite-element solver FEniCS.

Our method achieves a relative L_2 error of 3.20×10^{-3}, lower than all other methods listed in the table. It also outperforms the state-of-the-art CoPINN baseline, which achieves a relative L_2 error of 3.31×10^{-3}. These results demonstrate that the effectiveness of Legendre representations extends beyond sinusoidal solutions to nonlinear diffusion dynamics.

5 Ablation Study

We conduct ablation studies to examine how the maximum Legendre polynomial degree, network depth, and initialization parameter affect the accuracy of our method.

5.1 Polynomial degree and network depth

We jointly vary the maximum Legendre polynomial degree per layer and the network depth on the convection equation. We evaluate networks with 3, 4, and 5 layers, each with a maximum polynomial degree of 5, 7, 9, or 11, resulting in 12 configurations.

Figure 4 summarizes the results. For shallow networks with low polynomial degrees, increasing depth and degree improves accuracy. This suggests that the observed accuracy gains stem from the use of the Legendre orthogonal basis. However, these gains saturate at four layers and a maximum degree of nine. Further increases in depth or degree lead to a higher relative L_2 error. These results suggest that an appropriate depth and polynomial degree are needed to achieve low error.

5.2 Initialization parameter

We vary the first-layer initialization parameter c on the convection equation. As shown in Figure 5, $c = 128/211$ achieves the lowest relative L_2 error among the tested values. This result supports choosing c so that the first-layer scalar activation has unit slope at $z = 0$.

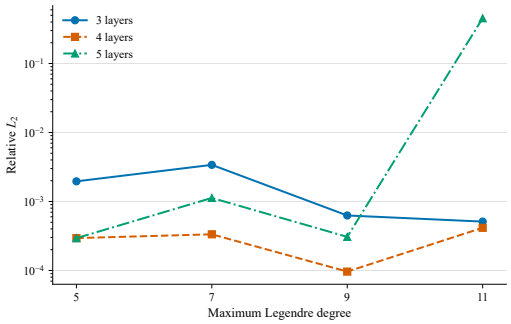


Figure 4: Relative L_2 error on convection as a function of maximum Legendre degree for networks with 3, 4, and 5 layers. Four layers and degree nine give the lowest error among the tested configurations.

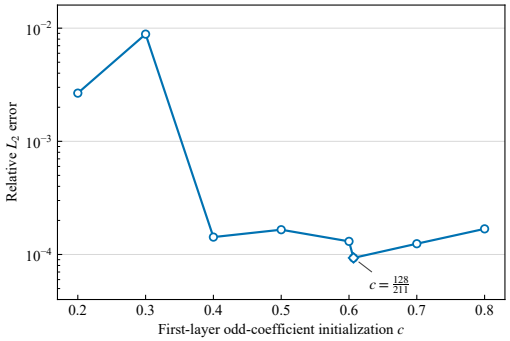


Figure 5: Effect of the first-layer odd-coefficient initialization c on convection accuracy. The diamond highlights $c = 128/211$, which yields the lowest relative L_2 error among the tested values.

6 Conclusion

In this work, we presented Legendre representation networks (LERENs) for learning PDE solutions without Fourier embeddings or sinusoidal activations. Motivated by the generation of higher-degree modes through polynomial composition, LERENs integrate learnable Legendre expansions into a multilayer architecture and combine representations across depths. Experiments on five PDE benchmarks demonstrate competitive accuracy on problems with sinusoidal or periodic structure, while also showing effectiveness on non-sinusoidal solutions in high-dimensional heat conduction and nonlinear diffusion. These results show that Legendre representations can support accurate high-frequency approximation in PINNs and remain useful beyond oscillatory solution profiles. More broadly, our findings suggest that the compositional properties of spectral bases offer a useful starting point for neural representation design. Extending this construction to other spectral bases, and understanding how their properties affect approximation and optimization, are natural directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools for debugging experimental code, preparing figures and tables, organizing mathematical arguments and drafting proof text, and interpreting experimental results. We also used it for manuscript drafting and LaTeX preparation. Code checks included parameter-count, initialization, derivative, and short-run execution checks. The authors directed the work and retain responsibility for the scientific claims, reported results, and the final manuscript.

REFERENCES

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic Differentiation in Machine Learning: a Survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.

Nuri Benbarka, Timon Höfer, Hamd ul-Moqeet Riaz, and Andreas Zell. Seeing Implicit Neural Representations As Fourier Series. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2041–2050, 2022.

Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, 1978.

Siyuan Duan, Wenyuan Wu, Peng Hu, Zhenwen Ren, Dezhong Peng, and Yuan Sun. CoPINN: Cognitive Physics-Informed Neural Networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 14759–14777. PMLR, 2025.

Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, and Jun Zhu. PINNacle: A Comprehensive Benchmark of Physics-Informed Neural Networks for Solving PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 76721–76774, 2024. doi: 10.52202/079017-2442.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4 (2):251–257, 1991. doi: 10.1016/0893-6080(91)90009-T.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

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.

Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. Society for Industrial and Applied Mathematics, 2007. doi: 10.1137/1.9780898717839.

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*, 2021.

David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. BACON: Band-Limited Coordinate Networks for Multiscale Scene Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16252–16262, 2022.

Yarong Liu, Hong Gu, Xiangjun Yu, and Pan Qin. Diminishing spectral bias in physics-informed neural networks using spatially-adaptive Fourier feature encoding. *Neural Networks*, 182:106886, 2025. doi: 10.1016/j.neunet.2024.106886.

Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. FINER: Flexible Spectral-bias Tuning in Implicit NEural Representation by Variable-periodic Activation Functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2713–2722, 2024.

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*, 3:218–229, 2021. doi: 10.1038/s42256-021-00302-5.

Yuezhou Ma, Haixu Wu, Hang Zhou, Huikun Weng, Jianmin Wang, and Mingsheng Long. ProPINN: Demystifying Propagation Failures in Physics Informed Neural Networks. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision*, volume 12346 of *Lecture Notes in Computer Science*, pp. 405–421. Springer, 2020. doi: 10.1007/978-3-030-58452-8_24.

Tiago Novello, Diana Aldana, Andre Araujo, and Luiz Velho. Tuning the Frequencies: Robust Training for Sinusoidal Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3071–3080, 2025.

David Pfau, James S. Spencer, Alexander G. de G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Physical Review Research*, 2:033429, 2020. doi: 10.1103/PhysRevResearch.2.033429.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 5301–5310, 2019.

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. doi: 10.1016/j.jcp.2018.10.045.

Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020. doi: 10.1126/science.aaw4741.

Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in Training PINNs: A Loss Landscape Perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42159–42191. PMLR, 2024.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473, 2020.

Chao Song and Yanghua Wang. Simulating seismic multifrequency wavefields with the Fourier feature physics-informed neural network. *Geophysical Journal International*, 232(3):1503–1514, 2023. doi: 10.1093/gji/ggac399.

N. Sukumar and Ankit Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, 389:114333, 2022. doi: 10.1016/j.cma.2021.114333.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7537–7547, 2020.

Jean Virieux and Stephane Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009. doi: 10.1190/1.3238367.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, 43 (5):A3055–A3081, 2021a. doi: 10.1137/20M1318043.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021b. doi: 10.1016/j.cma.2021.113938.

Sifan Wang, Hanwen Wang, Jacob H. Seidman, and Paris Perdikaris. Random Weight Factorization Improves the Training of Continuous Neural Representations. *arXiv preprint arXiv:2210.01274*, 2022. doi: 10.48550/arXiv.2210.01274.

Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An Expert's Guide to Training Physics-informed Neural Networks. *arXiv preprint arXiv:2308.08468*, 2023. doi: 10.48550/arXiv.2308.08468.

Sifan Wang, Bowen Li, Yuhan Chen, and Paris Perdikaris. PirateNets: Physics-informed Deep Learning with Residual Adaptive Networks. *Journal of Machine Learning Research*, 25(402): 1–51, 2024a.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421: 116813, 2024b. doi: 10.1016/j.cma.2024.116813.

Sifan Wang, Ananyae Kumar Bhartari, Bowen Li, and Paris Perdikaris. Gradient Alignment in Physics-informed Neural Networks: A Second-Order Optimization Perspective. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023. doi: 10.1016/j.cma.2022.115671.

Yulun Wu, Miguel Aguiar, Karl H. Johansson, and Matthieu Barreau. Iterative Training of Physics-Informed Neural Networks with Fourier-enhanced Features. In *International Conference on Learning Representations*, 2026.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks. *Communications in Computational Physics*, 28(5):1746–1767, 2020. doi: 10.4208/cicp.OA-2020-0085.

A MATHEMATICAL NOTATION

Table 4 summarizes the notation used in the paper. For time-dependent problems, $\mathbf{x} = (t, \mathbf{s})$ collects time and spatial coordinates; all gradients and Laplacians in the PDE definitions act on space only.

Table 4: Summary of notation.

Symbol	Meaning
$\mathbf{s} \in \Omega \subset \mathbb{R}^d$, t	Spatial coordinate, spatial domain, spatial dimension, and time.
$\mathcal{D}$, Γ_b, Γ_0	Computational domain, boundary, and initial surface.
u_θ, $u_\star$	Neural prediction and reference solution.
θ	All trainable model parameters.
$\mathcal{N}$, $\mathcal{B}$	Governing differential and boundary operators.
f, g_b, g_0	Prescribed source, boundary data, and initial data.
$\partial_n u$	Outward normal derivative $\mathbf{n}^\top \nabla_{\mathbf{s}} u$.
r_k, $\mathcal{X}_k$, N_k	Residual, collocation set, and point count for constraint k.
$\mathcal{L}_k$, λ_k	Loss component and its weight in $\mathcal{L} = \sum_k \lambda_k \mathcal{L}_k$.
L, m, ℓ	Hidden depth, width, and layer index.
$\mathbf{W}_\ell$, $\mathbf{b}_\ell$, α_ℓ	Affine weights, bias, and trainable preactivation scale.
$\mathbf{q}^{(\ell)}$, $\mathbf{h}^{(\ell)}$	Bounded expansion coordinates and hidden output.
P_n, $\mathcal{S}_\ell$, $c_{\ell n}$	Standard Legendre polynomial, selected degrees, and trainable expansion coefficient.
$\mathbf{v}$, b_{out}	Shared linear readout weights and output bias.
β_ℓ	Layer-output multiplier in the specified benchmark variants.
$\boldsymbol{\omega}_j$, $\boldsymbol{\xi}_{\mathbf{k}}$	Input and combined frequency vectors in the sinusoidal expansion.
$A_{i,\mathbf{k}}$, $\eta_{i,\mathbf{k}}$	Harmonic amplitude and phase.
$\gamma_r^{m,n}$	Coefficient of P_r in $P_m \circ P_n$; m, n here denote degrees.

B ADDITIONAL ABLATION EXPERIMENTS

We separate the contributions of the Legendre expansion and coefficient learning on convection and high-frequency Helmholtz. All three variants retain four affine hidden layers of width 256, trainable activation scales α_ℓ, and one shared linear readout applied to the sum of all four hidden-layer outputs. The $\tanh$ variant removes the Legendre expansion and directly uses $\tanh(\alpha_\ell z)$. The fixed-coefficient variant retains the selected Legendre bases but freezes all 22 expansion coefficients at initialization: the first-layer odd coefficients are $128/211$ and the even coefficients are zero, while the subsequent layers use $(c_{\ell 0}, c_{\ell 1}, c_{\ell 8}, c_{\ell 9}) = (0, 1/2, 0, 1/2)$. Full LEREN learns these coefficients along with the remaining network parameters.

Table 5 compares final errors and trainable parameter counts under the respective task protocols, using initialization seed 42. Convection and Helmholtz use 200,000 and 120,000 Adam updates, respectively. Full LEREN results are reused from the main text.

Table 5: Contributions of Legendre expansion and coefficient learning on convection and high-frequency Helmholtz. All variants share the affine backbone, trainable activation scales, and readout structure. Relative L_2 errors (RL2) are obtained with initialization seed 42 at the final training budget; lower is better, and the best result for each task is bold.

Variant	Trainable parameters	Convection RL2 ↓	High-frequency Helmholtz RL2 ↓
$\tanh$ with the same backbone	199,430	9.12×10^{-1}	2.47×10^{-1}
LEREN with fixed coefficients	199,430	2.09×10^{-4}	2.35×10^{-2}
Full LEREN	199,452	$\mathbf{9.33 \times 10^{-5}}$	$\mathbf{7.10 \times 10^{-3}}$

With fixed coefficients, the Legendre expansion reduces the convection and Helmholtz errors by factors of approximately 4.4×10^3 and 10.5, respectively, relative to the $\tanh$ variant. These two variants have identical trainable parameter counts and retain the same backbone, activation scales,

and readout. The improvement therefore supports a contribution from the nonlinear representation formed by the expansion, beyond changes in parameter count or readout structure.

Learning the expansion coefficients further reduces the errors by approximately 55.3% on convection and 69.7% on Helmholtz. The fixed-coefficient to full LEREN error ratios are 2.24 and 3.30. This adds only 22 trainable parameters, from $199{,}430$ to $199{,}452$, or about 0.011%.

Under these training protocols, the Legendre expansion provides the larger accuracy gain, and adapting its coefficients provides a further improvement.

C Computational cost on diffusion

Table 6 reports parameter counts, training time, and memory use. Adam runs use 50000 updates; ProPINN uses 1000 outer L-BFGS calls. IFeF-PINN uses $3000 + 15000 + 2000$ epochs across its three stages. L-BFGS calls can contain multiple loss/gradient evaluations, so these budgets are not interchangeable.

Table 6: Computational cost and final relative L_2 error on Diffusion. Time is the final logged training time. GPU memory is the observed peak of a separate bounded profile.

Method	Parameters	Time (s)	GPU (MiB)	Final RL2
Vanilla PINN	198,657	213.74	922	6.580×10^{-2}
JAX-PI	202,242	235.93	1,434	4.731×10^{-3}
PirateNet	727,171	600.64	2,926	2.022×10^{-2}
ProPINN (adapted)	15,038	3,224.55	968	1.537×10^{-2}
IFeF-PINN	4,641	973.02	1,538	2.079×10^{-2}
Ours ($c_{\text{init}} = 0.2$)	199,712	478.66	1,438	3.199×10^{-3}

Logged time includes scheduled evaluation and checkpoint work completed before the final log; IFeF-PINN includes all three stages. Historical hardware and concurrent workloads were not fully recorded, so the times do not establish a controlled speed ranking. JAX implementations are used for vanilla PINN, JAX-PI, PirateNet, and LEREN; the other implementations use PyTorch. IFeF-PINN's count includes all registered model parameters and excludes its fixed Fourier buffer.

Memory profiles use one job per RTX 4090 (24 GiB), with JAX preallocation disabled and full point counts retained: 1100 Adam updates, 25 L-BFGS calls, or 110 epochs per IFeF stage. Sampled GPU process-memory peaks reflect short runs, not full training. Our run has lower final error but longer training time than JAX-PI; these are single-run results under the stated protocols.

D Equations and experimental settings

D.1 Convection

Following the JAX-PI benchmark (Wang et al., 2023), we consider

$$\begin{aligned} \frac{\partial u}{\partial t} + 80\frac{\partial u}{\partial x} &= 0, && (x,t) \in (0, 2\pi) \times (0, 1], \\ u(x, 0) &= \sin x, && x \in [0, 2\pi], \\ u(0, t) &= u(2\pi, t), && t \in [0, 1]. \end{aligned} \tag{24}$$

The exact solution is $u_\star(x, t) = \sin(x - 80t)$. Evaluation uses a uniform 200×128 time–space grid. Our model uses raw coordinates and an explicit periodic boundary loss.

D.2 BURGERS

The viscous Burgers problem uses viscosity $\nu = 0.01/\pi$ and periodic conditions:

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0, &\qquad (x,t) \in (-1,1)\times(0,1], \\ u(x,0) = -\sin(\pi x), &\qquad x \in [-1,1], \\ u(-1,t) = u(1,t), &\qquad t \in [0,1], \\ \frac{\partial u}{\partial x}(-1,t) = \frac{\partial u}{\partial x}(1,t), &\qquad t \in [0,1]. \end{aligned} \tag{25}$$

We use the JAX-PI reference solution on its 201×512 time–space grid (Wang et al., 2023). Both periodic conditions are penalized explicitly.

D.3 HELMHOLTZ

The high-frequency problem follows the IFeF-PINN setting (Wu et al., 2026) on $\Omega_H = (0, 0.2)^2$:

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + u = f_H(x,y), &\qquad (x,y) \in \Omega_H, \\ u(x,y) = 0, &\qquad (x,y) \in \partial\Omega_H. \end{aligned} \tag{26}$$

The exact solution and source are

$$\begin{aligned} u_\star(x,y) &= \sin(100\pi x)\sin(100\pi y), \\ f_H(x,y) &= (1 - 20000\pi^2)u_\star(x,y). \end{aligned} \tag{27}$$

We use 23000 fixed centered Latin-hypercube interior points, 750 points per edge, and a 200×200 evaluation grid. The PDE residual is not frequency-normalized.

D.4 HEAT-5D

Let $B_5 = \{\mathbf{s} \in \mathbb{R}^5 : \|\mathbf{s}\|_2 < 1\}$. We use the five-dimensional heat problem (Hao et al., 2024):

$$\begin{aligned} \frac{\partial u}{\partial t} = \frac{1}{5}\Delta_{\mathbf{s}} u - \frac{\|\mathbf{s}\|_2^2}{5}u, &\qquad (\mathbf{s},t) \in B_5 \times (0,1], \\ u(\mathbf{s},0) = \exp\left(\frac{\|\mathbf{s}\|_2^2}{2}\right), &\qquad \mathbf{s} \in \overline{B}_5, \\ \frac{\partial u}{\partial n}(\mathbf{s},t) = \exp\left(\frac{1}{2} + t\right), &\qquad (\mathbf{s},t) \in \partial B_5 \times [0,1]. \end{aligned} \tag{28}$$

Here $\mathbf{n} = \mathbf{s}$ on the unit sphere and $u_\star(\mathbf{s},t) = \exp(\|\mathbf{s}\|_2^2/2 + t)$. Points are sampled uniformly in the ball, on its boundary, and in time. Evaluation uses 49152 fixed space–time points, with data seed 43.

D.5 NONLINEAR DIFFUSION

The CoPINN benchmark (Duan et al., 2025) has two spatial coordinates, $\mathbf{s} = (x,y) \in \Omega_D = (-1,1)^2$, and one time coordinate:

$$\begin{aligned} \frac{\partial u}{\partial t} = 0.05\nabla_{\mathbf{s}} \cdot (u\nabla_{\mathbf{s}} u), &\qquad (\mathbf{s},t) \in \Omega_D \times (0,1], \\ u(\mathbf{s},0) = u_0(\mathbf{s}), &\qquad \mathbf{s} \in \Omega_D, \\ u(\mathbf{s},t) = 0, &\qquad (\mathbf{s},t) \in \partial\Omega_D \times (0,1]. \end{aligned} \tag{29}$$

The initial profile is

$$\begin{aligned} u_0(x,y) = 0.25\exp\left[-\frac{(x-0.3)^2 + (y-0.2)^2}{0.1}\right] \\ + 0.4\exp\left[-15\left((x+0.5)^2 + (y+0.1)^2\right)\right] \\ + 0.3\exp\left[-20\left(x^2 + (y+0.5)^2\right)\right]. \end{aligned} \tag{30}$$

This follows the released code's coordinate convention, with x and y exchanged relative to the paper, without truncating the Gaussian boundary tails. Initial data apply in the interior and boundary data at positive times. We use the authors' FEniCS reference on a 101^3 space–time grid. The loss sums the PDE and initial MSEs and four separate boundary-edge MSEs.

D.6 Training settings

Table 7 summarizes the reported LEREN runs. All network parameters and Legendre coefficients use Adam (Kingma & Ba, 2015) with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.999, 10^{-8})$.

Table 7: LEREN training budgets and collocation counts. For periodic problems, the boundary count denotes paired boundary times; otherwise it denotes boundary points.

Problem	Adam updates	Interior	Initial	Boundary	Seed
Convection	200,000	4,096	128	200	42
Burgers	200,000	4,096	512	201	42
Helmholtz	120,000	23,000	—	3,000	42
Heat-5D	20,000	32,768	8,192	8,192	42
Diffusion	50,000	4,096	256	1,024	113

The learning rate is $10^{-3}0.9^{s/2000}$ at update s, except for Diffusion, which uses a constant 10^{-3}. Convection and Burgers resample interior points every step; Diffusion resamples all points every 100 steps; Helmholtz and Heat-5D use fixed full batches.

Convection and Burgers use 32 causal time chunks with tolerance 1. Both, together with Helmholtz, use gradient-norm balancing with EMA 0.9, updated after the first step and every 1000 steps. Heat-5D uses equal PDE, initial, and Neumann loss weights. Reported errors use the final training budget. All experiment results use first-layer odd-coefficient initialization $128/211$.

E Compared methods

Vanilla PINN. A coordinate MLP minimizes PDE residuals and initial/boundary losses, with derivatives computed by automatic differentiation (Raissi et al., 2019). Our baseline uses $\tanh$.

JAX-PI. It combines Fourier and periodic embeddings, gated ModifiedMLPs, random weight factorization, gradient-norm loss balancing, and causal weighting (Wang et al., 2023). Components vary by task; the Diffusion baseline uses ModifiedMLP without Fourier or periodic embeddings.

SIREN. SIREN uses sine activations and an initialization designed for periodic nonlinearities to represent functions and their derivatives (Sitzmann et al., 2020).

PirateNet. PirateNet combines coordinate embeddings with gated residual blocks and adaptive skip connections. Its initially zero residual coefficients enable a physics-informed least-squares initialization of the linear readout (Wang et al., 2024a).

IFeF-PINN. IFeF-PINN expands learned latent features with Fourier features and alternates feature learning with coefficient estimation (Wu et al., 2026). Our Diffusion implementation uses pretraining, end-to-end training, and alternating updates. The cited Helmholtz value 9.20×10^{-3} is the published mean for its primal-dual variant, IFeF-PD.

ProPINN. ProPINN aggregates differentiable multiscale neighborhoods to improve propagation via correlated parameter gradients (Ma et al., 2026). Our Diffusion variant uses $9/25/49$ Sobol perturbations. The Heat-5D limitation is direct Cartesian neighborhood expansion to six inputs.

CoPINN. CoPINN combines separable coordinate subnetworks with a scheduler based on coordinate gradients of the PDE-residual loss (Duan et al., 2025). It supplies a published Diffusion accuracy reference; we have no corresponding in-house runtime measurement.