LPINNs: First-Layer Gated Localization for Physics-Informed Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Physics-informed neural networks (PINNs) use one shared representation over the computational domain, which can become difficult to optimize on long domains and for high-order operators. We study a minimal alternative: multiply the first hidden activation of an otherwise unchanged dense PINN by input-dependent localization functions, giving first-layer units receptive fields without partitioning the domain or adding interface losses. We screen 13 families of localization functions, in up to three parameterizations each, on a nonlinear harmonic oscillator (HO), a heat equation on a long spatial interval, and a manufactured four-dimensional (4D) fourth-order problem, with ten paired seeds throughout. Three configurations give large reductions in solution error at matched budgets: (i) Fixed Gaussian localization functions on the 2π HO domain cut mean solution RMSE from 4.8369×10^{-1} to 8.83×10^{-3} at 3k epochs. (ii) The inverse-quadratic family with learnable centers and widths cuts it from 2.896×10^{-1} to 3.06×10^{-2} on the 8π heat domain at 10k epochs. (iii) Fixed bump localization functions cut it from 1.75947×10^{1} to 2.260×10^{-1} on the 4π 4D domain at 10k epochs. Every paired seed improves in these three comparisons. The screen also shows that the mechanism is not a free win: on HO only 2 of 13 families beat the baseline, and 10 of the remaining 11 are 9 to 23 times worse; on 4D four families are non-finite and five are more than three orders of magnitude worse than the baseline. The inverse-quadratic family is the only one that beats the baseline on all three equations. Overall, these results show that first-layer localization can provide measurable improvements to baseline PINNs on long-domain and high-order problems. We present this as an initial controlled proof-of-concept study: the comparisons are epoch-matched against a fully connected baseline to isolate whether first-layer localization helps and which families help, while computational-cost parity and comparisons against localized or domain-decomposition methods remain future work.

1 Introduction

A physics-informed neural network replaces a mesh-based solver with a neural approximation trained to reduce a differential-equation residual at collocation points (Lagaris et al., 1998; Raissi et al., 2019). Automatic differentiation makes this attractive, but training can stall when the loss is poorly conditioned, the target contains several length scales, or the residual contains high-order derivatives (Wang et al., 2021a; Krishnapriyan et al., 2021). A fully connected network also uses the same first-layer features across the entire coordinate domain. A feature useful near one end of a long interval is therefore active, and updated, at the other end.

One established response is domain decomposition: cPINNs, XPINNs, FBPINNs, and related local methods use several approximators and couple them through partitions, overlaps, or interface conditions (Jagtap et al., 2020b; Jagtap & Karniadakis, 2020; Moseley et al., 2023; Dong & Li, 2021). These methods add useful structure, but also add networks and coupling terms. This work explores whether part of the benefit can be obtained within a single network.

Our intervention is to multiply the first hidden representation by a bank of input-dependent localization functions. Each first-layer unit then has a receptive field, while all later layers remain dense and global. The residual, optimizer, collocation points, and boundary-condition transform are un-

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.

changed, and no subdomain partition or interface term is introduced, so the network still represents one smooth global function.

We report two kinds of evidence. Section 5 gives three detailed paired-seed comparisons, one per equation, in which localization reduces solution error by one to two orders of magnitude. Section 6 gives the screen over all 13 families of localization functions at the same settings, which is what makes the detailed comparisons interpretable: most families do not help, several are far worse than the baseline, and four are numerically non-finite on the 4D problem. Taken together, the two views support the following conclusions

- First-layer localization can substantially improve recovered solution accuracy in the tested long-domain and high-order settings, with the improvement holding in all ten paired seeds of each detailed comparison.
- The benefits of localization are equation-dependent, with different function families providing the strongest gains on different problems.

This is an initial controlled proof-of-concept study on one canonical ODE, one canonical PDE, and one manufactured high-order problem. Its comparisons are matched on training budget rather than on computational cost, and the only baseline is the matched fully connected PINN. That design is deliberate: holding the residual, constraint transform, optimizer, collocation scheme, budget, and seeds fixed and varying only the first-layer localization functions isolates whether localization helps at all and which families help on which equation. It does not establish equal-cost superiority or determine how the method compares with localized or domain-decomposition solvers. Direct comparisons with FBPINNs and related local methods, single-network feature embeddings, and standardized PINN benchmark suites are left to future work.

2 Related work

PINNs were introduced as neural approximations trained through differential-equation residuals and boundary or initial conditions (Lagaris et al., 1998; Raissi et al., 2019). A broad survey of the area is given by Karniadakis et al. (2021). Their optimization difficulties include imbalanced gradients (Wang et al., 2021a), stiff training dynamics explained through the neural tangent kernel (Wang et al., 2022), spectral bias on oscillatory targets (Rahaman et al., 2019; Wang et al., 2021b), and residual minima that do not correspond to accurate solutions (Krishnapriyan et al., 2021).

Local PINN methods address these difficulties by using multiple subnetworks or local basis functions. cPINNs enforce conservation across subdomains, XPINNs generalize the decomposition to space-time, and FBPINNs combine overlapping local networks through a partition of unity (Jagtap et al., 2020b; Jagtap & Karniadakis, 2020; Moseley et al., 2023). Local extreme learning machines and compactly supported variational bases follow a related motivation (Dong & Li, 2021; Kharazmi et al., 2021), and Heinlein et al. (2021) review the line of work. Our method keeps one network and introduces locality only through input-dependent modulation of its first hidden layer.

The construction is related to localized units in radial-basis networks (Broomhead & Lowe, 1988; Park & Sandberg, 1991) and to neuron-wise adaptive activations (Jagtap et al., 2020a), but differs from both: our localization functions depend on the input and are inserted inside a standard multilayer PINN rather than used as the output basis or as input-independent slopes. A second single-network line of work modifies the input representation through Fourier features or other positional embeddings (Tancik et al., 2020; Wang et al., 2021b); radial-basis embeddings provide a localized alternative. These methods and ours both act near the input, but an embedding supplies features that the first linear layer mixes, whereas our construction leaves that layer intact and gates its learned activation. We do not compare against these methods empirically here; the present study is scoped to the matched baseline PINN.

3 FIRST-LAYER LOCALIZATION

Let $x \in \mathbb{R}^d$ be the physical coordinates and $\widetilde{x} \in [0,1]^d$ their coordinate-wise normalization. A standard multilayer perceptron is

$$h^{(1)} = \phi(W_0 \widetilde{x} + b_0), \tag{1}$$

$$h^{(\ell+1)} = \phi(W_\ell h^{(\ell)} + b_\ell), \qquad \ell \geq 1, \tag{2}$$

followed by a scalar linear output N_θ. Localization replaces only the first hidden representation with

$$h^{(1)}_{\mathrm{loc}} = h^{(1)} \odot g(\widetilde{x}; \theta_g), \tag{3}$$

where g returns one value per first-layer unit and $\odot$ is elementwise multiplication. We refer to g as a localization function: it multiplicatively modulates the first hidden-layer representation and gives each unit an input-dependent receptive field. The substitution is complete: $h^{(1)}_{\mathrm{loc}}$ takes the place of $h^{(1)}$ downstream, so the second hidden layer is $h^{(2)} = \phi(W_1 h^{(1)}_{\mathrm{loc}} + b_1)$ and Eq. 2 applies unchanged for $\ell \geq 2$. For the Gaussian family, the localization function g_j associated with first-layer unit j has centers μ_j and widths σ_j giving

$$g_j(\widetilde{x}) = \exp\left[-\frac{1}{2}\sum_{k=1}^{d}\left(\frac{\widetilde{x}_k - \mu_{kj}}{\sigma_{kj}}\right)^2\right] \tag{4}$$

with centers initialized along the diagonal of $[0,1]^d$: for a first layer of m units, μ_{kj} is the j-th of m values evenly spaced from 0.05 to 0.95, reused in every coordinate. This makes the construction well defined for the 4D problem, where 32 units do not form a regular Cartesian grid. The widths are isotropic and initialized to $\sigma_{kj} = \max(0.08, 1.5/\max(m, 4))$, which equals 0.08 for both the $m = 64$ and $m = 32$ first layers used here. Every family uses the separable form

$$g_j(\widetilde{x}) = \prod_{k=1}^{d} \psi\left(\frac{\widetilde{x}_k - \mu_{kj}}{\sigma_{kj}}\right), \tag{5}$$

where ψ is the one-dimensional profile of the family; Eq. 4 is the corresponding Gaussian product. The Supplementary Material gives the profiles and parameterizations.

What the localization functions change, and what they do not. The localization field is a smooth function of the input, so the network output remains one globally defined smooth function; there is no partition, no interface penalty, and no continuity constraint to satisfy. What changes is which parameters a given collocation point can move. Differentiating Eq. 3 w.r.t. $\widetilde{x}_k$ gives

$$\partial_k h^{(1)}_{\mathrm{loc},j} = g_j \, \partial_k h^{(1)}_j + h^{(1)}_j \, \partial_k g_j \tag{6}$$

and both terms are small for units whose localization function is negligible at that point. A residual evaluated near one end of the domain, therefore, produces gradients concentrated on the first-layer units whose receptive fields cover that region, rather than on all of them, while the later layers still mix features from every unit. This is a weaker form of locality than domain decomposition and it is correspondingly cheaper: one bank of localization functions, no extra networks, no interface terms, and no per-subdomain hyperparameters.

Why one layer. We place the localization functions at the first layer for two practical reasons. (i) It is the smallest change that can create receptive fields, and later layers already mix features globally. (ii) Localizing at every layer would also add spatially varying parameters to every derivative order in the residual, including four orders in the 4D problem. The reported experiments therefore characterize a single bank of localization functions; they are not evidence that other placements cannot work.

Localization-function families and parameterizations. We consider 13 families of localization functions: Gaussian, activated, super-Gaussian, Ricker, boxcar, Laplace, Cauchy, raised-cosine, bump, Gabor, Morlet, inverse-quadratic, and triangular. Each family is tested with fixed centers and widths (F), learnable widths (S), and learnable centers and widths (M), where the implementation supports the variant numerically. All 13 families are screened at the primary setting of each equation in Section 6, and the three configurations examined in detail in Section 5 are members of this same set.

4 EXPERIMENTAL SETUP

Problems and constraint transforms. We study three model problems, pairing each equation with a hard constraint transform for its solution.

(i) Harmonic Oscillator: The HO problem is the undamped nonlinear oscillator

$$u''(t) + \sin(u(t)) = 0, \qquad u(0) = 0, \quad u'(0) = 1, \tag{7}$$

on $[0, L]$, with $L \in \{\pi, 2\pi, 4\pi\}$ and a numerical reference trajectory. Lengthening L adds oscillations without changing the operator. We impose the initial conditions exactly using

$$u(t) = t + t^2 N_\theta(t). \tag{8}$$

(ii) Heat Equation: The heat problem is

$$u_t - u_{xx} = 0, \qquad u^*(x, t) = \sin(x)e^{-t}, \tag{9}$$

on $x \in [0, L]$, $t \in [0, 1]$, with $L \in \{\pi, 2\pi, 4\pi, 8\pi\}$. At $L = 8\pi$, the target contains four spatial periods. We enforce the initial and boundary conditions using

$$u(x, t) = \sin(x) + t\, x(L - x)N_\theta(x, t). \tag{10}$$

This transform is exact for the initial condition and the left boundary at any domain length, but at the right boundary it returns the time-independent value $\sin(L)$ whereas the target is $u^*(L, t) = \sin(L)e^{-t}$. It therefore enforces the correct boundary physics only when $\sin(L) = 0$, that is, when L is an integer multiple of π. Every domain used here, $L \in \{\pi, 2\pi, 4\pi, 8\pi\}$, satisfies this condition. For an arbitrary domain length this particular transform would impose a non-decaying value at $x = L$ and would need to be replaced; this limitation concerns the chosen transform, not the localization mechanism.

(iii) 4D manufactured problem: The manufactured fourth-order problem is

$$\frac{\partial^4 u}{\partial x_1 \partial x_2 \partial x_3 \partial x_4} - 10 = 0, \qquad u^*(x) = 10 \prod_{k=1}^{4} x_k, \tag{11}$$

on $[0, L]^4$, with $L \in \{\pi, 2\pi, 4\pi\}$ (Han et al., 2018; Kharazmi et al., 2021). Its target network output is the constant $N_\theta^* = 10$. We impose the boundary conditions using

$$u(x) = \left(\prod_{k=1}^{4} x_k \right) N_\theta(x). \tag{12}$$

In all three cases, the constraints are exact by construction, so the objective contains only the residual term and requires no loss weighting.

Baseline. Throughout the matched comparisons, the baseline is the standard ungated fully connected PINN of Raissi et al. (2019). It uses the same architecture, residual, boundary-condition transform, collocation scheme, optimizer, training budget, and seeds as the corresponding proposed localized model, and differs from it only in Eq. 3, which for the baseline is $g \equiv 1$. Every comparison reported in Tables 2–5 is paired in this sense, thereby isolating the effect of the first-layer localization functions.

Models and setup. HO and Heat use two hidden layers of width 64; 4D uses width 32. HO uses sine activations, which suit an oscillatory trajectory (Sitzmann et al., 2020), and the other two problems use tanh. The complete study contains 24,960 configurations and 249,600 seed runs across the three problems. The domain and training-epoch settings for the results reported here are summarized in Table 1.

Metrics. We report four quantities. *Final residual loss* is the training objective at the last epoch, evaluated on that epoch's collocation batch. *Residual RMSE*, *solution RMSE*, and *solution MAE* are evaluated on a fixed per-problem evaluation set described in the Supplementary Material at the

Table 1: Domain sizes and epoch budgets used in the localization campaign.

Problem	Domain values	Epoch budgets
HO	$t \in [0, L],\ L \in \{\pi, 2\pi, 3\pi, 4\pi\}$ $x \in [0, L],\ t \in [0, 1]$	1k, 3k, 5k, 10k, 50k, 100k
Heat	$L \in \{\pi, 2\pi, 4\pi, 8\pi\}$	1k, 3k, 5k, 10k, 50k
4D	$x \in [0, L]^4,\ L \in \{\pi, 2\pi, 4\pi\}$	1k, 3k, 5k, 10k

selected checkpoint, which is the checkpoint with the lowest evaluated residual RMSE among those taken at the first epoch, at regular intervals thereafter, and at the last epoch. The first quantity and the remaining three are therefore measured on different point sets and at different points in training, so final residual loss is not the square of the reported residual RMSE. Solution errors are measured against the numerical HO reference or the exact Heat and 4D solutions. Reported values are means over the ten seeds. For the localization family screen we use mean solution MSE, the mean across seeds of each seed's squared solution RMSE. We keep residual and solution metrics separate throughout, because the residual is what training minimizes, whereas the solution error is what the method is for, and the two can be decoupled (Krishnapriyan et al., 2021).

5 Results

5.1 Harmonic oscillator (HO)

As reported in Table 2, at $L = 2\pi$ and 3k epochs, fixed Gaussian localization functions reduce mean solution RMSE by 98.2% and MAE by 98.3% relative to the baseline. Residual RMSE falls by 78.0% and the final residual loss by 96.3%. Figure 1 shows the same improvement visually: the localized trajectory stays close to the numerical reference across the interval, while the baseline deviates more strongly; the training-history and absolute-residual panels show the corresponding lower residuals. All ten paired seeds improve in solution RMSE and MAE. With ten independent pairs, a unanimous outcome corresponds to an exact one-sided sign-test $p = 2^{-10} \approx 0.001$. We use this test rather than an interval on the mean because the seed-to-seed spread of the baseline is large, as Section 6 quantifies.

Table 2: HO at $L = 2\pi$ with 64×64 networks and fixed Gaussian localization functions. Means over ten paired seeds. All metric entries are shown in scientific notation. Bold marks the localized solution RMSE and MAE at the larger budget.

Localization Function	Epochs	Final Residual Loss	Residual RMSE	Solution RMSE	Solution MAE
Baseline	3k	5.01×10^{-2}	1.20×10^{-1}	4.84×10^{-1}	4.08×10^{-1}
Gaussian	3k	1.85×10^{-3}	2.64×10^{-2}	$\mathbf{8.83 \times 10^{-3}}$	$\mathbf{7.06 \times 10^{-3}}$
Baseline	1k	6.23×10^{-2}	1.74×10^{-1}	6.76×10^{-1}	5.57×10^{-1}
Gaussian	1k	1.14×10^{-1}	3.02×10^{-1}	2.15×10^{0}	1.80×10^{0}

At 1k epochs the ordering reverses: the localized model is worse on all four metrics, with solution RMSE $3.2\times$ the baseline's. We report the short budget because this sensitivity is a property of the method rather than an artifact to be tuned away.

5.2 Heat equation on a long spatial domain

Table 3 reports that at $L = 8\pi$ and 10k epochs, inverse-quadratic localization functions with learnable centers and widths reduce solution RMSE by 89.4% and MAE by 90.3% relative to the baseline. Residual RMSE falls by 78.7% and the final residual loss by 95.2%. Figure 2 shows the same improvement visually: at $t = 0.5$, the localized solution follows the reference more closely than the baseline, while the training-history and absolute-residual panels show lower residuals for the localized model. All four reported metrics improve in all ten paired seeds. The two budgets are informative together: extending the baseline from 3k to 10k epochs yields only a 0.4% improvement,

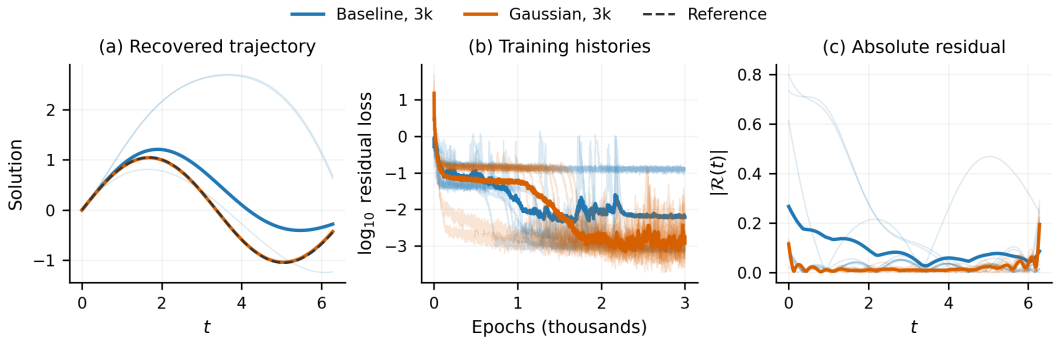

Figure 1: HO at $L = 2\pi$ and 3k epochs: recovered trajectories with the numerical reference (left), residual-loss histories (center), and absolute residuals (right). Thin curves are the ten paired seeds and thick curves are their means, with loss averaged in $\log_{10}$ space.

whereas the localized model improves by 69.7%; the localized model at 3k is already better than the baseline at 10k. Thus, the baseline's poorer solution accuracy is not explained simply by insufficient training.

Table 3: Heat at $L = 8\pi$ with 64×64 networks and inverse-quadratic localization functions with learnable centers and widths. Means over ten paired seeds. All metric entries are shown in scientific notation. Bold marks the localized solution RMSE and MAE at the larger budget.

Localization Function	Epochs	Final Residual Loss	Residual RMSE	Solution RMSE	Solution MAE
Baseline	3k	6.43×10^{-1}	7.07×10^{-1}	2.91×10^{-1}	2.33×10^{-1}
Inverse-quadratic	3k	1.21×10^{-1}	3.35×10^{-1}	1.01×10^{-1}	7.17×10^{-2}
Baseline	10k	5.21×10^{-1}	7.05×10^{-1}	2.90×10^{-1}	2.32×10^{-1}
Inverse-quadratic	10k	2.51×10^{-2}	1.50×10^{-1}	$\mathbf{3.06 \times 10^{-2}}$	$\mathbf{2.24 \times 10^{-2}}$

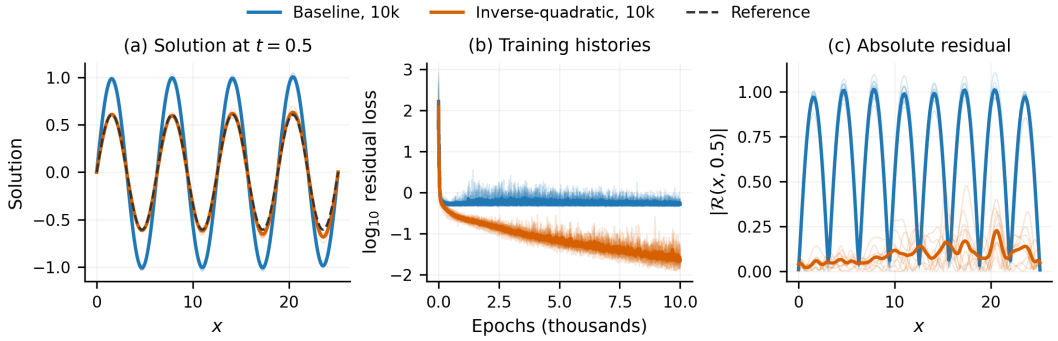

Figure 2: Heat equation at $L = 8\pi$ and 10k epochs for both models: solution at $t = 0.5$ (left), residual-loss histories (center), and absolute residuals at $t = 0.5$ (right). Thin curves are the ten paired seeds and thick curves are their means, with loss averaged in $\log_{10}$ space.

5.3 Four-dimensional fourth-order problem

Table 4 shows that at $L = 4\pi$ and 10k epochs, fixed bump localization functions reduce solution RMSE by 98.7% and MAE by 99.1% relative to the baseline. Residual RMSE falls by 95.3%, and these three evaluation metrics improve in all ten paired seeds.

This comparison also separates residual accuracy from solution accuracy. The final residual loss differs by only 11.6%, whereas the mean solution RMSE differs by a factor of 78. A useful intuition

is that recovering u from this mixed fourth-order residual involves four integrations, so residual errors can accumulate over the long domain even when the residual itself is small. Figure 3 shows the same contrast visually: on the diagonal slice $x_1 = x_2 = x_3 = x_4$, the localized solution stays close to exact recovery, while the pointwise relative-error panel shows a larger fraction of localized evaluations at smaller errors. The training histories are much closer together than the solution-error results, illustrating why residual loss alone is not sufficient here. Where no reference solution is available, residual loss is therefore a weak selection signal, and a usable selection rule for choosing localization families without ground truth remains future work.

Table 4: 4D at $L = 4\pi$ with 32×32 networks and fixed bump localization functions. Means over ten paired seeds. All metric entries are shown in scientific notation. Bold marks the localized solution RMSE and MAE at the larger budget.

Localization Function	Epochs	Final Residual Loss	Residual RMSE	Solution RMSE	Solution MAE
Baseline	3k	2.62×10^{-3}	5.35×10^{-2}	1.42×10^{2}	1.11×10^{2}
Bump	3k	2.52×10^{-3}	1.11×10^{-3}	9.22×10^{-1}	5.29×10^{-1}
Baseline	10k	5.34×10^{-5}	7.32×10^{-3}	1.76×10^{1}	1.41×10^{1}
Bump	10k	4.72×10^{-5}	3.44×10^{-4}	$\mathbf{2.26 \times 10^{-1}}$	$\mathbf{1.30 \times 10^{-1}}$

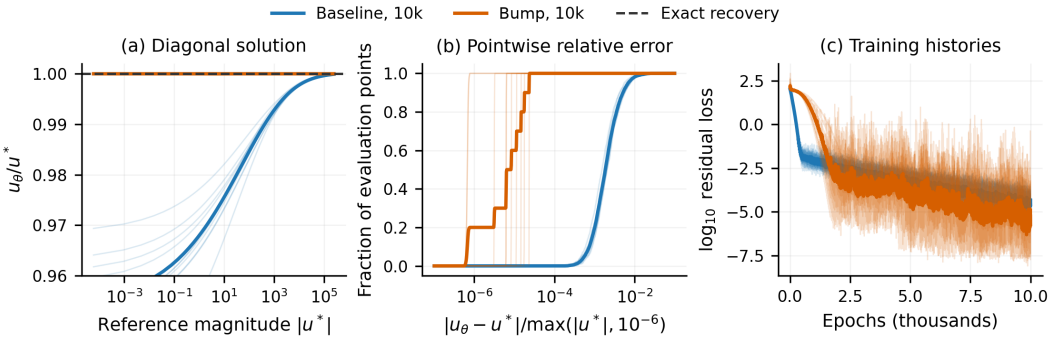

Figure 3: 4D problem at $L = 4\pi$ and 10k epochs for both models: normalized solution on the diagonal slice (left), empirical distribution of relative error (center), and residual-loss histories (right). Thin curves are the ten paired seeds and thick curves are their means, with loss averaged in $\log_{10}$ space.

6 Localization-Function Family Screen

Table 5 reports mean solution MSE for all 13 families of localization functions at the primary setting of each equation. Three properties of the table govern how it should be read. First, each family entry is the minimum over the parameterizations that were finite, whereas the baseline row involves no such minimum, so the family entries are biased low relative to the baseline. Second, mean MSE averages squared errors, so a single diverged seed can dominate a cell. Third, the three columns have different solution scales and are not comparable across columns. We therefore treat only order-of-magnitude differences within a column as evidence, and rely on the paired-seed win counts of Section 5 for the detailed comparisons.

This ablation study highlights three main findings. First, localization benefits depend on the function family and the equation, with several families providing substantial gains over the baseline. On HO only Gaussian and inverse-quadratic beat the baseline; of the remaining 11, ten are 9 to 23 times worse and Cauchy is 1.4 times worse. On Heat 11 of 13 families are nominally better, but eight of those lie within 30% of the baseline, which the selection bias above is enough to explain, leaving Cauchy, inverse-quadratic, and triangular as the only substantial gains. On the 4D problem, four families are non-finite for every parameterization tested, five land between 8×10^3 and 6×10^4 times

the baseline error, and four beat it. The variation across families is more pronounced for a higher-order problem, making the selection of a localization function suited to the equation important.

Second, inverse-quadratic localization is the most consistent family across the three equations. It beats the baseline on all three equations and gives the lowest entry for HO and Heat. On 4D, bump performs best, while the inverse-quadratic entry is 17.6 times the bump entry and still 315 times below the baseline. These results make inverse-quadratic localization functions with learnable centers and widths a strong default candidate for future experiments.

Third, the detailed comparisons in Section 5 provide representative examples from the broader family screen rather than selecting its single best entry. The screen also contains stronger configurations: its lowest HO entry is inverse-quadratic at 2×10^{-5}, which is 4.3 times below the fixed Gaussian configuration featured in Section 5.1, while its lowest 4D bump entry uses learnable widths rather than the fixed parameters of Section 5.3. The featured settings were chosen because they produce large and unanimous improvements across seeds, while the broader screen identifies further configurations for future study.

The non-finite entries are recorded rather than replaced by a large finite value. For the Gaussian, Ricker, Gabor, and Morlet families on the 4D problem, the saved records contain non-finite values from the first recorded training loss, so no finite training state was captured for those runs; this is a missing measurement, not a small or zero loss. All four families are finite on HO and Heat and fail only here, where the residual differentiates the localization functions once in each of four coordinates. The records establish the numerical failure, but not its precise mechanism. We hypothesize that the Gaussian-envelope profiles can combine very small gate values with rapidly growing derivative factors under repeated differentiation, producing underflow, overflow, or an indeterminate product in the fourth-order automatic-differentiation graph. This is only a hypothesis: super-Gaussian and Laplace profiles also have exponential envelopes and remained finite, and we did not run the precision, width-scaling, or derivative-instrumentation studies needed to distinguish this explanation from cancellation or implementation-specific effects. No equation-specific stabilization was added for the fourth-order residual, because the study was intended to keep one generalized implementation across all three equations.

Table 5: Screen of localization-function families at the primary settings: HO at 2π/3k, Heat at 8π/10k, and 4D at 4π/10k. Entries are mean solution MSE over ten seeds; lower is better. F, S, and M denote fixed, learnable-width, and learnable-center/width parameterizations; each cell gives the lowest finite parameterization for that family, with the winning variant in parentheses. n/a means that no tested parameterization produced a finite solution metric for that problem. Bold marks the lowest valid entry in each problem column. Columns have different solution scales and should be compared only within a column.

Localization Function	HO	Heat	4D
Baseline	9.95×10^{-1}	8.39×10^{-2}	3.13×10^{2}
Gaussian	8.40×10^{-5} (F)	7.96×10^{-2} (F)	n/a
Activated	1.79×10^{1} (S)	7.50×10^{-2} (S)	1.78×10^{7} (F)
Super-Gaussian	1.54×10^{1} (S)	7.47×10^{-2} (M)	1.62×10^{7} (S)
Ricker	1.15×10^{1} (F)	8.34×10^{-2} (F)	n/a
Boxcar	9.14×10^{0} (S)	7.24×10^{-2} (S)	1.35×10^{7} (S)
Laplace	1.51×10^{1} (M)	7.90×10^{-2} (M)	7.89×10^{1} (F)
Cauchy	1.44×10^{0} (M)	1.64×10^{-3} (M)	3.61×10^{0} (F)
Raised-cosine	9.17×10^{0} (M)	7.16×10^{-2} (S)	7.23×10^{6} (M)
Bump	1.73×10^{1} (F)	6.56×10^{-2} (S)	$\mathbf{5.65 \times 10^{-2}}$ **(S)**
Gabor	1.86×10^{1} (F)	8.49×10^{-2} (F)	n/a
Morlet	2.28×10^{1} (F)	8.41×10^{-2} (F)	n/a
Inverse-quadratic	$\mathbf{1.96 \times 10^{-5}}$ **(M)**	$\mathbf{9.78 \times 10^{-4}}$ **(M)**	9.92×10^{-1} (M)
Triangular	1.39×10^{1} (M)	1.18×10^{-2} (S)	2.60×10^{6} (S)

Finally, the results together recover the across-seed spread that neither shows on its own, because mean MSE and mean RMSE are the first two moments of the same ten values. For the HO baseline

they imply a standard deviation of 8.7×10^{-1} around a mean solution RMSE of 4.8369×10^{-1}, against 2.5×10^{-3} around 8.83×10^{-3} for the fixed Gaussian model. The baseline's seed-to-seed variation is thus larger than its own mean error, which is the reason we report unanimity across pairs instead of an interval on the mean.

7 Conclusion and future work

Across the three tested problems, first-layer localization improves recovered solution accuracy while leaving the PINN, residual, optimizer, and boundary transform unchanged, and the improvement is reproduced in every paired seed of the detailed comparisons. The 13-family screen shows that localization is a flexible design choice whose benefit depends on matching the family to the equation and derivative order: inverse-quadratic localization with learnable centers and widths is the most consistent configuration across the three equations, while the broader screen also identifies additional family–problem combinations worth exploring. Because the study uses matched comparisons and controlled analytic benchmarks, it isolates the optimization effect of localization and provides a clear foundation for testing the approach on broader equations and more realistic scientific settings. These conclusions are bounded by the study's scope: comparisons are epoch-matched rather than compute-matched, the evaluation uses three controlled problems rather than a standardized benchmark suite, and the only baseline is the matched ungated network.

Future work should extend the study to a broader range of equations and more realistic scientific applications, moving beyond the controlled benchmark problems considered here. Direct comparisons against domain-decomposition methods and single-network feature embeddings, together with standardized benchmarks, will test whether the benefits of first-layer localization persist for nonseparable targets, coupled systems, complex geometries, and real-world problem settings. A selection rule that does not require a reference solution, and diagnostic experiments for the fourth-order nonfinite runs, are also left for future work.

Reproducibility statement

Sections 3 and 4 define the problems, transforms, architectures, and matched comparison protocol. The Supplementary Material records the localization profiles, parameterizations, optimizer, learning rates, collocation and evaluation sets, initialization, seeds, and checkpoint rule used for every reported number. Baseline and localized models share all settings except the bank of localization functions in Eq. 3. The experiments were run on a 16-inch Apple M3 Max MacBook Pro with 36 GB RAM, using CPU-only multiprocessing. The source code is available on the Git repository https://anonymous.4open.science/r/LPINNs-1E31/

AI use statement

Generative AI tools were used for code-assisted editing, formatting, figure preparation, and language revision. The authors reviewed and verified the AI-assisted text, equations, citations, reported results, figures, and code. Generative AI was not used to generate experimental data, to select which results to report, or to replace author verification of the reported results. The authors take responsibility for the final content.

References

David S. Broomhead and David Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2(3):321–355, 1988.

Suchuan Dong and Zongwei Li. Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 387:114129, 2021.

Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

Alexander Heinlein, Axel Klawonn, Martin Lanser, and Janine Weber. Combining machine learning and domain decomposition methods for the solution of partial differential equations—a review. *GAMM-Mitteilungen*, 44(1):e202100001, 2021.

Ameya D. Jagtap and George Em Karniadakis. Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5):2002–2041, 2020.

Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks. *Proceedings of the Royal Society A*, 476(2239):20200334, 2020a.

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, 2020b.

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.

Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, 2021.

Isaac E. Lagaris, Aristidis Likas, and Dimitrios I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (FBPINNs): A scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, 49(4):62, 2023.

Jooyoung Park and Irwin W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257, 1991.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, 2019.

Maziar Raissi, Paris Perdikaris, and George Em 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.

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*, 2020.

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*, 2020.

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–A3083, 2021a.

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.

Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022.