Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

From Soft to Hard Constraints in Physics-Informed Extreme Learning Machines: Analysis and Algorithms

Ruonan Zhai
Shijiazhuang University
Xiaoyuan Shi
Shijiazhuang University
Xiong Xiong
xiongxiongnwpu@mail.nwpu.edu.cn
Northwestern Polytechnical University
Zhuo Zhang
National University of Defense Technology
Maowei You
National University of Singapore

Article

Keywords: Physics-informed extreme learning machine, Hard constraint, Distance function, Dirichlet boundary condition, Gram-matrix analysis, Random-feature least squares

Posted Date: October 22nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11169466/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

FROM SOFT TO HARD CONSTRAINTS IN PHYSICS-INFORMED EXTREME LEARNING MACHINES: ANALYSIS AND ALGORITHMS

Ruonan Zhai [1] **and Xiaoyuan Shi** [2] **and Xiong Xiong** [3,*] **and Zhuo Zhang** [4] **and Maowei You** [5]

[1]Science College, Shijiazhuang University, Shijiazhuang, 050035, P. R. China
[2]School of Music,
Shijiazhuang University, Shijiazhuang, 050035, P. R. China
[3]School of Mathematics and Statistics,
Northwestern Polytechnical University, Xi'an, 710072, P. R. China
[4]College of Computer Science and Technology,
National University of Defense Technology, Changsha 410073, Hunan, P. R. China
[5]Yong Loo Lin School of Medicine, National University of Singapore,
Singapore 117597, Singapore

[*]Corresponding author(s). E-mail(s):
xiongxiongnwpu@mail.nwpu.edu.cn

ABSTRACT

Physics-informed extreme learning machines (PIELM) cast the numerical solution of partial differential equations (PDEs) as a single linear least-squares problem by fixing the hidden-layer weights at random, yet the standard soft-constraint formulation augments the PDE residual system with boundary equations, requiring a single coefficient vector to reconcile two competing objectives. A Gram-matrix eigenvalue analysis reveals that this competition suppresses boundary influence along directions where the PDE block dominates, and that uniform row scaling cannot reduce the resulting spectral spread. To eliminate this structural imbalance, we embed Dirichlet boundary conditions directly into the trial function via a lifting term and a boundary-vanishing distance function, so that the resulting least-squares system involves only the interior PDE residual while the one-shot closed-form structure is preserved. Boundary conditions are thereby satisfied to machine precision without entering the coefficient solve. Across seven Dirichlet benchmarks covering Helmholtz, Poisson, Kovasznay Navier–Stokes, wave, and nonlinear reaction equations on convex, non-convex, and curved domains, the hard-constrained formulation yields identically zero boundary residuals and reduces relative L^2 errors by up to seven orders of magnitude under identical random bases and collocation layouts, at comparable computational cost. The code and data accompanying this manuscript will be made publicly available at `https://github.com/xgxgnpu/Physics-informed-vibe-coding/tree/main/HC-PIELM`.

Keywords: Physics-informed extreme learning machine; Hard constraint; Distance function; Dirichlet boundary condition; Gram-matrix analysis; Random-feature least squares

Article Highlights

- Gram-matrix eigenvalue analysis reveals a structural imbalance that suppresses boundary fidelity in soft-constrained PIELM.
- A distance-function embedding enforces Dirichlet data exactly while preserving the one-shot least-squares framework.
- The hard-constrained formulation reduces relative errors by up to seven orders of magnitude at comparable cost.

1 Introduction

Partial differential equations (PDEs) underpin the modeling of physical and engineering phenomena. Classical numerical methods—finite element, finite difference, and spectral methods—are mature but remain constrained by mesh generation for complex geometries [1] and by the curse of dimensionality. Embedding PDE residuals into neural-network loss functions, first explored by Dissanayake and Phan-Thien [2], has been developed over the past decade into Physics-Informed Neural Networks (PINNs) [3] and related mesh-free paradigms such as the Deep Ritz method [4] and the Deep Galerkin Method [5]—implemented in open-source libraries such as DeepXDE [6]. These methods have been applied to the incompressible Navier–Stokes equations [7, 8] and a wide range of forward and inverse problems. However, PINNs rely on nonlinear optimization and are therefore sensitive to hyperparameters and prone to gradient pathologies [9], spectral bias [10–12], and failure on stiff multi-scale dynamics [13–15]. Adaptive loss weighting [16] and adaptive activation functions [17, 18] alleviate but do not remove the underlying optimization difficulty [19, 20].

Extreme Learning Machines (ELM) [21, 22], grounded in the universal approximation capability of single-hidden-layer networks [23] and the random-feature framework [24], provide an alternative that bypasses nonlinear optimization entirely: the hidden-layer weights and biases are randomly generated and fixed, so only the output-layer coefficients need to be determined, reducing the learning problem to a single linear least-squares solve. Random-feature methods have since been applied directly to the numerical solution of PDEs with spectral accuracy [25], theoretical convergence guarantees free from the curse of dimensionality [26], and efficient collocation-based implementations [27–29]. Physics-Informed ELM (PIELM) [30] combines this idea with PDE residual embedding, constructing system matrices from PDE operators evaluated at interior collocation points and boundary operators evaluated at boundary collocation points, then solving the stacked over-determined system in one step. PIELM trains orders of magnitude faster than PINNs and avoids gradient vanishing, local minima, and learning rate tuning altogether. Subsequent work has extended ELM-based PDE solvers to elliptic problems with sharp gradients [31], nonlinear bifurcation problems [32], domain decomposition settings [33, 34], pre-trained ELM bases [35], high-dimensional PDEs with A-TFC constructions [36], hidden-layer-concatenated ELM architectures [37, 38], eigenvalue problems [39], and various other extensions reviewed in [40].

Most PIELM implementations enforce boundary conditions by a *soft-constraint* strategy: BC rows are appended below the PDE rows in a single least-squares system. This couples two objectives with different row counts, operator orders, and preferred coefficient directions. Under homogeneous Dirichlet conditions, the target solution can often be written as a distance factor times a smoother interior field; the ELM basis must then approximate this product directly, capturing both the boundary zero structure and the interior variation. The task grows harder for high-degree distance factors (e.g., degree ten on a pentagram). In nonlinear Picard iterations, boundary residuals can also enter subsequent linearized solves through the nonlinear coupling.

Hard-constraint strategies embed boundary conditions into the trial function construction and thereby avoid boundary fitting as a competing least-squares objective. Lagaris et al. [41, 42] proposed early hard-constraint neural network formulations for rectangular and irregular domains, and McFall and Mahan [43] introduced length factors for exact BC satisfaction on general boundaries. Berg and Nyström [44] and Sheng and Yang [45] developed domain-aware trial-function ansatze for complex geometries, while Dong and Ni [46] addressed periodic boundary constraints via hard embedding. Sukumar and Srivastava [47] developed distance-function methods for exact BC imposition in deep neural networks, including R-functions and TFI-based treatments of non-homogeneous data. Liu et al. [48] proposed a unified hard-constraint framework that transforms Dirichlet, Neumann, and Robin conditions into analytically solvable linear equations via extra fields, and Berrone et al. [49–51] provided a systematic comparison of boundary-enforcement strategies in variational and collocation PINN formulations.

Leake and Mortari [52] introduced the Theory of Functional Connections (TFC) [53] for constrained function approximation, and Schiassi et al. [54] combined TFC with ELM in the Extreme Theory of Functional Connections (X-TFC) framework. Lu et al. [55] implemented hard constraints in the PINN setting by composing a distance function with a neural network output plus a boundary-data function. More recently, De Falco, Schiassi, and Calabrò [56] proposed LSE-ELM, which enforces boundary constraints through equality-constrained least squares and extends the X-TFC viewpoint to general-shaped domains, and Dong and Zhang [57] extended transfinite interpolation with TFC and ELM to enforce Dirichlet, Neumann, and Robin conditions exactly on curved quadrilateral domains. These works show that the ansatz $L + D\,N$ itself is a well-established hard-constraint construction; from a complementary theoretical angle, Xie et al. [58] recently provided a spectral analysis showing that boundary functions act as spatial modulators reshaping the neural tangent kernel eigenspectrum. The present paper focuses on a different PIELM route: continuous distance-function embedding within a closed-form random-feature least-squares solver, explicit constructions for non-convex polygonal and curved domains, and a Gram-matrix diagnosis of the soft-constraint system.

This paper makes three contributions. **(i)** We give a Gram-matrix diagnosis of Soft-PIELM: the joint least-squares solution is expressed as a matrix-weighted combination of the PDE-only and BC-only optima, and the associated generalized eigenvalues identify directions in which boundary influence is weak. As a corollary of this analysis, uniform row scaling is shown to preserve the spectral spread, motivating a structural rather than parametric remedy.

(ii) We formulate a distance-function Hard-PIELM workflow that embeds Dirichlet data continuously on $\partial\Omega$ while preserving the closed-form random-feature least-squares structure. The implementation uses four analytic constructions—polygonal edge-distance products, implicit level sets, TFC spatiotemporal lifting, and TFI multi-field lifting—covering the geometries and constraint types used in the experiments. **(iii)** We connect the construction to numerical evidence: seven Dirichlet-type benchmarks use identical ELM bases and collocation sets for Soft and Hard PIELM, showing lower boundary residuals and relative L^2 error reductions ranging from several-fold to seven orders of magnitude for the hard formulation at comparable cost.

Section 2 analyzes the structural imbalance of Soft-PIELM from the Gram-matrix perspective. Section 3 presents the Hard-PIELM framework, distance-function construction strategies, and approximation properties. Section 4 reports numerical experiments on seven benchmark problems. Section 5 concludes with a discussion of limitations and future directions.

2 Structural Imbalance of Soft-Constrained PIELM

2.1 Problem Formulation

Consider a boundary value problem on a domain $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$:

$$\mathcal{L}[u] = f \quad \text{in } \Omega, \qquad \mathcal{B}[u] = g \quad \text{on } \partial\Omega, \tag{2.1}$$

where $\mathcal{L}$ is a p-th order (possibly nonlinear) differential operator and $\mathcal{B}$ is the boundary operator (Dirichlet type throughout this paper). In the ELM framework [21], N hidden neurons generate the basis functions

$$\phi_j(\mathbf{x}) = \sigma(W_j \cdot \mathbf{x} + b_j), \qquad j = 1, \ldots, N, \tag{2.2}$$

where $W_j \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$ are randomly sampled and then fixed, and σ denotes the activation function ($\sigma = \tanh$ throughout). PIELM [30] adopts the trial function

$$u_S = \sum_{j=1}^{N} c_j \, \phi_j \tag{2.3}$$

and discretizes (2.1) at m_P interior and m_B boundary collocation points, yielding the stacked least-squares system

$$\min_{\mathbf{c}\in\mathbb{R}^N} \left\| \begin{pmatrix} A_P \\ A_B \end{pmatrix} \mathbf{c} - \begin{pmatrix} r_P \\ r_B \end{pmatrix} \right\|_2^2, \tag{2.4}$$

where $A_P \in \mathbb{R}^{m_P \times N}$ has entries $[\mathcal{L}[\phi_j]](\mathbf{x}_i^{\text{int}})$, $A_B \in \mathbb{R}^{m_B \times N}$ has entries $[\mathcal{B}[\phi_j]](\mathbf{x}_i^{\text{bdy}})$, and r_P, r_B collect the right-hand-side values. Define the PDE and BC Gram matrices

$$G_P := A_P^\top A_P, \qquad G_B := A_B^\top A_B. \tag{2.5}$$

The normal equation associated with (2.4) takes the form

$$(G_P + G_B)\, \mathbf{c}^* = A_P^\top r_P + A_B^\top r_B. \tag{2.6}$$

Equation (2.6) makes the bi-objective nature of the system transparent: the coefficient vector $\mathbf{c}$ must simultaneously minimize the PDE residual $\|A_P\mathbf{c} - r_P\|^2$ and the BC residual $\|A_B\mathbf{c} - r_B\|^2$, and the least-squares formulation sums these two terms without any weighting. The following theorem rewrites this competition in a form that separates the PDE and boundary influences.

2.2 Weighted Decomposition Theorem

Theorem 2.1 (Weighted Decomposition)**.** *Suppose $G_P + G_B$ is invertible. Let $\mathbf{c}_P := G_P^\dagger A_P^\top r_P$ and $\mathbf{c}_B := G_B^\dagger A_B^\top r_B$ denote the PDE-only and BC-only least-squares solutions, respectively. Then the joint minimizer $\mathbf{c}^*$ of* (2.4) *satisfies*

$$\mathbf{c}^* = \Gamma_P\, \mathbf{c}_P + \Gamma_B\, \mathbf{c}_B, \tag{2.7}$$

where the influence matrices

$$\Gamma_P := (G_P + G_B)^{-1} G_P, \qquad \Gamma_B := (G_P + G_B)^{-1} G_B \tag{2.8}$$

satisfy $\Gamma_P + \Gamma_B = I$. Let $\{(\mu_i, q_i)\}_{i=1}^N$ be the generalized eigenpairs of (G_P, G_B), i.e., $G_P q_i = \mu_i G_B q_i$ with $\{q_i\}$ G_B-orthonormal. Then

$$\Gamma_P\, q_i = \frac{\mu_i}{1+\mu_i}\, q_i, \qquad \Gamma_B\, q_i = \frac{1}{1+\mu_i}\, q_i. \tag{2.9}$$

Proof. By definition, $A_P^\top r_P = G_P \mathbf{c}_P$ and $A_B^\top r_B = G_B \mathbf{c}_B$. Substituting into the normal equation gives $(G_P + G_B)\mathbf{c}^* = G_P \mathbf{c}_P + G_B \mathbf{c}_B$. Left-multiplying by $(G_P + G_B)^{-1}$ yields (2.7)–(2.8). For the spectral characterization, from $G_P q_i = \mu_i G_B q_i$ we obtain $(G_P + G_B)q_i = (1 + \mu_i) G_B q_i$, whence $\Gamma_B q_i = (G_P + G_B)^{-1} G_B q_i = \frac{1}{1+\mu_i} q_i$. The identity $\Gamma_P = I - \Gamma_B$ gives the first relation. □

Theorem 2.1 shows that the joint solution $\mathbf{c}^*$ is a matrix-weighted average of $\mathbf{c}_P$ and $\mathbf{c}_B$, with weights governed by the generalized eigenvalues μ_i. In directions where $\mu_i \gg 1$, the BC influence weight $1/(1 + \mu_i)$ is small. The influence matrices Γ_P and Γ_B depend only on A_P and A_B (the operator structure and collocation geometry), not on the right-hand-side data r_P, r_B.

2.3 Three-Factor Driving Mechanism

Remark 1 (Three-factor mechanism for $\mu_i \gg 1$)**.** *In typical PIELM configurations, μ_i is driven well above unity by three multiplicative factors. The following scaling is a heuristic order-of-magnitude estimate rather than a rigorous bound:*

$$\mu_i \;\sim\; \frac{m_P}{m_B} \cdot \frac{\bar{\sigma}_P^2}{\bar{\sigma}_B^2} \cdot \rho_i, \tag{2.10}$$

where m_P/m_B is the row-count ratio (Factor I), $\bar{\sigma}_P^2/\bar{\sigma}_B^2$ is the operator-order amplification (Factor II), and ρ_i is the spectral anisotropy (Factor III). Here $\bar{\sigma}_P$ and $\bar{\sigma}_B$ should be read as representative row scales of the PDE and boundary blocks, for example root-mean-square row norms after applying the corresponding operators to the same ELM basis. Factor I reflects the row-count ratio: standard collocation schemes place $m_P \gg m_B$ interior points, giving PDE rows a numerical majority. For instance, with $m_P = 10{,}000$ and $m_B = 256$ (the Poisson 2D benchmark), this factor alone contributes a ratio of approximately 39. Factor II captures operator-order amplification: for a p-th order operator, the PDE rows involve p-th derivatives of ϕ_j whose magnitudes scale as $O(s^{2p})$ with the ELM weight scale s, while Dirichlet BC rows involve ϕ_j itself at $O(1)$. For the Laplacian ($p = 2$) with $s = 5$, this factor is of order $s^4 = 625$. Factor III encodes the spectral anisotropy: the eigenvalue distribution $\{\mu_i\}$ can be highly non-uniform, with $\mu_{\max}/\mu_{\min} \gg 1$ in high-frequency directions.

The three factors compound multiplicatively, suggesting large values of μ_i in practice. In the L-shape Poisson benchmark ($n_e = 5$), the condition numbers of the Soft and Hard systems are both of order 10^{17}, yet the Soft L^2 error at $N = 1000$ is 3.46×10^{-12} compared with Hard's 2.42×10^{-15}. In the wave benchmark, five constraint types (PDE, two spatial BCs, displacement IC, velocity IC) contribute five blocks to the stacked system, and the combined m_P/m_B ratio across all constraint rows exceeds 50. The resulting BC residual at $N = 1000$ remains at 5.56×10^{-3}, qualitatively in line with the heuristic expectation of reduced boundary influence in the stacked fit.

2.4 Irreparability of Row Scaling

The following observation shows that the simplest remedy—uniformly rescaling the PDE rows—cannot resolve the spectral spread. The result is an elementary consequence of the homogeneity of generalized eigenvalues under scalar multiplication; we state it explicitly to motivate the hard-constraint approach developed in Section 3.

Remark 2 (Uniform row scaling preserves spectral spread)**.** *If A_P is replaced by αA_P for any $\alpha > 0$, then:* (i) *$\mu_i \to \alpha^2 \mu_i$ for all i;* (ii) *$\mu_{\max}/\mu_{\min}$ is strictly invariant;* (iii) *$\ker(G_B)$ is unchanged. Consequently, no $\alpha > 0$ can make all μ_i simultaneously close to unity.*

Proof. After scaling, $G_P \to \alpha^2 G_P$ while G_B is unaffected. The generalized eigenvalue equation $\alpha^2 G_P q_i = \mu_i' G_B q_i$ gives $\mu_i' = \alpha^2 \mu_i$ with identical eigenvectors, so $\mu'_{\max}/\mu'_{\min} = \mu_{\max}/\mu_{\min}$. The null space $\ker(G_B) = \ker(A_B)$ depends only on A_B and is independent of A_P or α. □

Remark 2 establishes that uniform scaling shifts all eigenvalues by the same multiplicative factor but leaves the relative spread intact. Directions in $\ker(G_B)$ (where $\mu_i = \infty$) receive zero BC influence regardless of scaling. This result is deliberately limited to scalar row-block weights; more elaborate non-uniform or adaptive weighting strategies change the algebraic problem and are outside the present analysis. The hard-constraint approach developed next removes the boundary objective rather than attempting to tune its relative weight.

3 Hard-Constrained PIELM

3.1 Unified Hard-Constrained Ansatz

To remove boundary fitting from the least-squares objective identified in Section 2, we embed the boundary conditions directly into the trial function. Define the Hard-PIELM ansatz

$$u_H(\mathbf{x}) = L(\mathbf{x}) + D(\mathbf{x}) \cdot \sum_{j=1}^{N} c_j \, \phi_j(\mathbf{x}), \tag{3.1}$$

where $L\colon \overline{\Omega} \to \mathbb{R}$ is a *lifting function* satisfying $\mathcal{B}[L] = g$ on $\partial\Omega$, and $D\colon \overline{\Omega} \to \mathbb{R}$ is a *distance function* satisfying $D(\mathbf{x}) = 0$ for all $\mathbf{x} \in \partial\Omega$ and $D(\mathbf{x}) > 0$ for all $\mathbf{x} \in \Omega$. By construction, $\mathcal{B}[u_H]\big|_{\partial\Omega} = g$ for *any* coefficient vector $\mathbf{c}$, so the Dirichlet data are built into the trial space rather than fitted by optimization.

Substituting (3.1) into the PDE $\mathcal{L}[u_H] = f$ and evaluating at m_P interior collocation points yields an interior-residual least-squares system

$$\min_{\mathbf{c}\in\mathbb{R}^N} \left\| A_P^{\mathrm{hard}}\,\mathbf{c} - \tilde{f} \right\|_2^2, \tag{3.2}$$

where $[A_P^{\mathrm{hard}}]_{ij} = \mathcal{L}[D\phi_j](\mathbf{x}_i)$ and $\tilde{f}_i = f(\mathbf{x}_i) - \mathcal{L}[L](\mathbf{x}_i)$. Constructing A_P^{hard} requires the differential operator applied to the product $D\phi_j$. For $\mathcal{L} = -\Delta$, the Leibniz rule gives

$$\Delta(D\,\phi_j) = (\Delta D)\,\phi_j + 2\,\nabla D \cdot \nabla\phi_j + D\,\Delta\phi_j, \tag{3.3}$$

where all three terms are computed from closed-form expressions of D and the $\tanh$ activation derivatives

$$\phi_j' = 1 - \phi_j^2, \qquad \phi_j'' = -2\,\phi_j\,(1 - \phi_j^2). \tag{3.4}$$

Because every quantity in (3.3)–(3.4) admits a closed-form expression, no numerical differentiation or automatic differentiation is required. For nonlinear PDEs of the form $-\Delta u + h(u) = f$, we employ Picard iteration. At step k, the nonlinear term is linearized around the previous iterate u_k, yielding the linear system

$$-\Delta u_{k+1} + h'(u_k)\,u_{k+1} = f - h(u_k) + h'(u_k)\,u_k, \tag{3.5}$$

which is solved using ansatz (3.1). Because $u_k\big|_{\partial\Omega} = g$ exactly at every iteration, the boundary residual does not propagate into the nonlinear coupling.

The complete Hard-PIELM workflow is summarized in Algorithm 1.

Algorithm 1 Hard-PIELM solver

Require: PDE operator $\mathcal{L}$, source f, BC data g, domain Ω
Require: Neuron count N, collocation points m_P, activation σ
Construct distance function D and lifting function L (Strategies I–IV)
Precompute $D_{\max}$; normalize $D \leftarrow D/D_{\max}$ if needed
Generate random weights $W_j \sim \mathcal{U}(-s, s)$, biases $b_j \sim \mathcal{U}(-s, s)$
Compute ELM basis $\phi_j(\mathbf{x}_i) = \sigma(W_j \cdot \mathbf{x}_i + b_j)$ at all collocation points
Compute derivatives $\nabla\phi_j$, $\Delta\phi_j$ via (3.4)
Compute ∇D, ΔD analytically from the closed-form expression of D
Assemble $[A_P^{\mathrm{hard}}]_{ij} = \mathcal{L}[D\,\phi_j](\mathbf{x}_i)$ via Leibniz rule (3.3)
Assemble $\tilde{f}_i = f(\mathbf{x}_i) - \mathcal{L}[L](\mathbf{x}_i)$
if PDE is nonlinear **then**
    Initialize $u_0 = L$
    for $k = 0, 1, \ldots, K-1$ **do**
        Linearize via (3.5); update A_P^{hard} and $\tilde{f}$
        Solve $\mathbf{c}^{(k+1)} = \arg\min \|A_P^{\mathrm{hard}}\,\mathbf{c} - \tilde{f}\|_2^2$
        Set $u_{k+1} = L + D \cdot \sum c_j^{(k+1)}\phi_j$
    end for
else
    Solve $\mathbf{c}^* = \arg\min \|A_P^{\mathrm{hard}}\,\mathbf{c} - \tilde{f}\|_2^2$
end if
return $u_H = L + D \cdot \sum c_j\phi_j$

3.2 Distance-Function Construction Strategies

The distance function D and lifting function L must satisfy: **(C1)** $D \in C^p(\overline{\Omega})$; **(C2)** $D > 0$ in Ω and $D = 0$ on $\partial\Omega$; **(C3)** all required derivatives of D admit closed-form expressions; **(C4)** $\mathcal{B}[L] = g$ on $\partial\Omega$. We develop four construction strategies.

Strategy I: Edge-distance products for polygonal domains. Let Ω be a polygon with n_e edges, the i-th edge described by the linear equation $\ell_i(\mathbf{x}) = a_i x + b_i y + c_i = 0$ (oriented so that $\ell_i > 0$ inside Ω). Then

$$D(\mathbf{x}) = \prod_{i=1}^{n_e} \ell_i(\mathbf{x}) \tag{3.6}$$

is a polynomial of degree n_e vanishing on every edge. Since each ℓ_i is linear, $\Delta\ell_i = 0$ and $\nabla\ell_i = (a_i, b_i)$ is constant. The Laplacian of D can therefore be expanded as

$$\Delta D = \sum_{i\neq k} (\nabla\ell_i \cdot \nabla\ell_k) \prod_{j\neq i,k} \ell_j \,, \tag{3.7}$$

which contains only constant inner products $\nabla\ell_i \cdot \nabla\ell_k = a_i a_k + b_i b_k$ multiplied by products of linear functions, all computable in closed form. The gradient of D follows from the product rule as

$$\nabla D = \sum_{i=1}^{n_e} \nabla\ell_i \prod_{j\neq i} \ell_j \,. \tag{3.8}$$

This strategy covers rectangles ($n_e = 4$), L-shaped domains ($n_e \leq 6$), and pentagrams ($n_e = 10$). For homogeneous Dirichlet conditions, $L = 0$.

Strategy II: Implicit level sets for curved domains. When $\partial\Omega$ is defined by a smooth implicit function $F(\mathbf{x}) = 0$ with $F > 0$ inside Ω, we set $D = F$ directly. For instance, the five-petal flower domain used in Section 4.8 has

$$D = 1 - r^6 + \mathrm{Re}[(x+iy)^5], \qquad r = \sqrt{x^2+y^2}, \qquad \Delta D = -36\, r^4, \tag{3.9}$$

where the last identity holds because $\mathrm{Re}[(x+iy)^5]$ is harmonic. All derivatives are available in closed form.

Strategy III: Spatiotemporal TFC framework. For time-dependent PDEs with initial conditions (ICs) and boundary conditions, D and L are constructed to satisfy all constraints simultaneously. For the wave equation on $[0,1]\times[0,T]$ with $u(x,0) = u_0(x)$, $u_t(x,0) = 0$, and $u(0,t) = u(1,t) = 0$, we set

$$L(x,t) = u_0(x), \qquad D(x,t) = x(1-x)\, t^2. \tag{3.10}$$

The factor $x(1-x)$ enforces the spatial BCs, while t^2 ensures both $D|_{t=0} = 0$ and $D_t|_{t=0} = 0$, satisfying the displacement and velocity initial conditions. All partial derivatives of D required by the wave operator are available analytically:

$$\begin{aligned} &D_x = (1-2x)\, t^2, \quad D_{xx} = -2\, t^2, \quad D_t = 2\, x(1-x)\, t, \\ &D_{tt} = 2\, x(1-x), \quad D_{xt} = 2\,(1-2x)\, t. \end{aligned} \tag{3.11}$$

Strategy IV: Transfinite Interpolation (TFI) lifting. For non-homogeneous Dirichlet conditions on rectangular domains $[x_a, x_b]\times[y_a, y_b]$, the lifting function is constructed by Gordon–Hall TFI:

$$\begin{aligned} L(x,y) = {} & (1-\xi)\, g(x_a, y) + \xi\, g(x_b, y) + (1-\eta)\, g(x, y_a) + \eta\, g(x, y_b) \\ & - (1-\xi)(1-\eta)\, g(x_a, y_a) - \xi(1-\eta)\, g(x_b, y_a) \\ & - (1-\xi)\eta\, g(x_a, y_b) - \xi\,\eta\, g(x_b, y_b), \end{aligned} \tag{3.12}$$

where $\xi = (x - x_a)/(x_b - x_a)$ and $\eta = (y - y_a)/(y_b - y_a)$. After TFI lifting, D is constructed as in Strategy I for the homogeneous residual. For multi-field systems such as Navier–Stokes (fields u, v, p), each field receives its own lifting L_u, L_v, L_p.

When n_e is large, $D_{\max} := \max_{\mathbf{x}\in\Omega} |D(\mathbf{x})|$ can be very small (e.g., $D_{\max} = 7.94\times 10^{-6}$ for the pentagram with $n_e = 10$). To avoid ill-conditioning, we normalize $D \to D/D_{\max}$ so that $\max|D| = 1$; $D_{\max}$ is precomputed on a dense evaluation grid before assembling the system matrix.

3.3 Theoretical Advantages

Proposition 1 (Elimination of multi-objective competition)**.** *In the Hard-PIELM system* (3.2)*, the boundary Gram matrix satisfies $G_B = 0$, and consequently $\Gamma_P = I$, $\Gamma_B = 0$. All N degrees of freedom are devoted to minimizing the PDE residual, and the BC residual is identically zero for any* $\mathbf{c}$*.*

This follows immediately from the absence of boundary rows in (3.2): A_B does not exist, so $G_B = A_B^\top A_B = 0$, and the influence matrix decomposition of Theorem 2.1 degenerates to $\mathbf{c}^* = \mathbf{c}_P$. No generalized eigenvalue μ_i arises, eliminating the spectral imbalance at the algebraic level.

Proposition 2 (Conditional approximation-space decoupling)**.** *Assume that $D \in C^p(\overline{\Omega})$, $D = 0$ on $\partial\Omega$, $D > 0$ in Ω, and that the quotient*

$$g_D := (u^* - L)/D$$

admits an $L^2(\Omega)$ extension with the regularity required by the ELM approximation space. Then the Hard-PIELM approximation error satisfies

$$\|u_H - u^*\|_{L^2(\Omega)} \leq \|D\|_{L^\infty(\Omega)} \cdot \inf_{\mathbf{c}\in\mathbb{R}^N} \left\| \sum_{j=1}^{N} c_j \phi_j - g_D \right\|_{L^2(\Omega)}, \tag{3.13}$$

so the ELM basis approximates the quotient field g_D rather than the full product Dg_D.

Proof. For any coefficient vector $\mathbf{c}$, write $\hat{g}_D = \sum_{j=1}^{N} c_j \phi_j$. Since $u_H = L + D\hat{g}_D$ and $u^* = L + Dg_D$ almost everywhere in Ω, we have

$$u_H - u^* = D(\hat{g}_D - g_D).$$

Taking the $L^2(\Omega)$ norm and using $\|D(\hat{g}_D - g_D)\|_{L^2} \le \|D\|_{L^\infty} \|\hat{g}_D - g_D\|_{L^2}$ gives (3.13); minimizing over $\mathbf{c}$ completes the argument. □

The quotient condition is a genuine regularity assumption: if D has higher-order zeros at corners or edge intersections, the numerator $u^* - L$ must vanish to a compatible order for g_D to remain smooth. Under this condition, Soft-PIELM must approximate the product $D \cdot g_D$ directly, simultaneously capturing both the polynomial zero structure of D and the physical content of g_D. In Hard-PIELM, the network approximates g_D, with D supplied analytically by the trial function.

Hard-PIELM also provides two stability benefits. First, *D-weighted boundary suppression*: the pointwise error satisfies

$$|u_H(\mathbf{x}) - u^*(\mathbf{x})| = |D(\mathbf{x})| \cdot |\hat{g}_D(\mathbf{x}) - g_D(\mathbf{x})| \;\to\; 0 \quad \text{as } \mathbf{x} \to \partial\Omega, \tag{3.14}$$

so errors are automatically damped near the boundary. Second, *Picard iteration boundary consistency*: since $u_k\big|_{\partial\Omega} = g$ exactly at every iteration k, boundary mismatch is not carried into the nonlinear coupling through the iterates, reducing one route by which Soft-PIELM errors can accumulate over K Picard steps.

Table 1 contrasts the structural properties of Soft-PIELM and Hard-PIELM.

Table 1: Structural comparison of Soft-PIELM and Hard-PIELM.

Soft-PIELM limitation	Hard-PIELM resolution	Supporting result
Multi-objective: $\mu_i \gg 1$, BC suppressed	$G_B = 0$, single PDE objective	Prop. 1
Uniform row scaling cannot reduce Soft spectral spread	Boundary rows removed from the Hard system	Remark 2; Eq. (3.2)
Approximation mismatch when n_e is large	ELM approximates smooth g_D only	Prop. 2
No boundary suppression; Picard error $O(C^K)$	$D \to 0$ suppression; exact BC each step	Eq. (3.1)

4 Numerical Experiments

4.1 Experimental Setup

We use a matched protocol throughout. The random seed is fixed at `seed` $= 42$ for every case. At each neuron count N, Soft-PIELM and Hard-PIELM use the identical random weight matrix W and bias vector b (and hence the same ELM basis functions), the same interior collocation set, and the same boundary collocation set. The sole difference is the constraint strategy: Soft appends BC rows to the least-squares system, while Hard embeds BCs through the ansatz (3.1). Two sweep protocols are employed: (i) neuron sweep—fix the number of interior collocation points N_{int} and vary N; (ii) collocation sweep—fix N and vary N_{int}. Errors are evaluated on independent test grids distinct from the training collocation set. The four metrics reported throughout this section are defined as follows. The relative L^2 error measures the overall accuracy of the predicted solution:

$$e_{L^2} \;=\; \frac{\|u_{\text{pred}} - u_{\text{ref}}\|_2}{\|u_{\text{ref}}\|_2}, \tag{4.1}$$

where the norms are computed over the test grid. The PDE residual and BC residual quantify the point-wise satisfaction of the governing equation and boundary conditions, respectively:

$$r_{\text{PDE}} \;=\; \sqrt{\frac{1}{N_{\text{test}}^{\text{int}}} \sum_{i=1}^{N_{\text{test}}^{\text{int}}} \big|\mathcal{L}[u_{\text{pred}}](\mathbf{x}_i) - f(\mathbf{x}_i)\big|^2}, \qquad r_{\text{BC}} \;=\; \sqrt{\frac{1}{N_{\text{test}}^{\text{bd}}} \sum_{j=1}^{N_{\text{test}}^{\text{bd}}} \big|u_{\text{pred}}(\mathbf{x}_j) - g(\mathbf{x}_j)\big|^2}. \tag{4.2}$$

The error ratio compares the relative L^2 errors of Soft-PIELM and Hard-PIELM:

$$R \;=\; \frac{e_{L^2}^{\text{Soft}}}{e_{L^2}^{\text{Hard}}}. \tag{4.3}$$

Table 2: Summary of seven benchmark cases at $N = 1000$ (Kovasznay at $N = 500$; see text). PDE-res and BC-res ratios are Soft/Hard (>1 means Hard is better). L^2 ratio is Soft/Hard.

Benchmark	Domain	D-strategy	n_e	PDE-res ratio	BC-res ratio	Best Hard L^2	L^2 ratio
Helmholtz 1D	$(0, 2)$	Lifting + edge	2	2.1	∞	3.95×10^{-13}	7.3
Poisson 2D	$[-1, 1]^2$	Edge product	4	122	1.4×10^{5}	5.35×10^{-14}	147
L-shape Poisson	L-shape	Polynomial	5	85	4.4×10^{4}	2.42×10^{-15}	1430
Kovasznay flow	Rectangle	TFI + edge	4	46	1.2×10^{11}	7.46×10^{-8}	206
Wave 1D	Spatiotemporal	TFC	4	52	2.3×10^{13}	2.01×10^{-8}	1.2×10^{5}
Pentagram NL react.	Pentagram	Edge product	10	1.2×10^{5}	1.6×10^{35}	9.02×10^{-15}	1.0×10^{7}
Flower NL react.	Five-petal flower	Level set	6	112	8.0×10^{15}	3.30×10^{-14}	3720

A value $R > 1$ indicates a lower Hard-PIELM error in the reported metric. The implementation is written in JAX, and all computations use float64 arithmetic on a single NVIDIA RTX 4090 GPU.

Table 2 provides an overview of the seven benchmark problems and their key metrics at the listed neuron configurations.

The seven cases exercise edge products on convex and non-convex polygons, implicit level sets on a curved domain, and TFC/TFI liftings for spatiotemporal and multi-field constraints. Larger L^2 ratios occur in cases with higher-degree distance functions, multiple constraint types, or Picard linearization. The following subsections describe each case in detail.

4.2 Helmholtz 1D — Non-Homogeneous Dirichlet

Consider the one-dimensional Helmholtz equation

$$u'' - k^2 u = f, \quad x \in (0, 2), \quad k^2 = 10, \quad u(0) = g_0,\ u(2) = g_1, \tag{4.4}$$

with reference solution $u(x) = \sin(3\pi x + 3\pi/20)\cos(2\pi x + \pi/10) + 2$ and the source term f computed by substitution. The boundary values $g_0 = u(0)$ and $g_1 = u(2)$ are non-homogeneous. For the Hard ansatz, the lifting and distance functions are

$$L(x) = (1 - x/2)\, g_0 + (x/2)\, g_1, \qquad D(x) = x(2 - x), \tag{4.5}$$

giving $n_e = 2$. Interior collocation uses $N_{\text{int}} = 10{,}000$ uniform random points; the ELM weight scale is $s = 5$.

Table 3: Helmholtz 1D neuron sweep ($N_{\text{int}} = 10{,}000$).

N	Soft PDE-res	Hard PDE-res	Soft BC-res	Hard BC-res	Soft L^2	Hard L^2	Ratio
10	8.12e+01	8.24e+01	9.27e+01	**0**	1.79e+01	2.35e−01	76
50	2.47e−02	9.24e−03	7.17e−05	**0**	1.65e−05	6.98e−06	2.4
200	2.75e−08	5.26e−08	1.64e−10	**0**	5.15e−11	1.52e−11	3.4
1000	9.48e−10	4.45e−10	4.80e−12	**0**	2.87e−12	3.95e−13	7.3

Table 3 reports the results of the neuron sweep. The lifting-plus-distance construction gives zero boundary residual at every N. At $N = 10$, the Soft system must fit the two non-homogeneous boundary values with very few degrees of freedom, producing an L^2 ratio of 76. For $N = 50$–1000, both methods converge rapidly and Hard remains lower in relative L^2 error by factors from 2.4 to 7.3. The condition numbers are both of order 10^{16} (Soft: 3.16×10^{16}, Hard: 2.20×10^{16}), indicating that the gain is not simply a conditioning effect. The lifting function L absorbs the non-homogeneous boundary data, so the ELM basis in the Hard system approximates only the residual interior structure. The monotonic growth of the L^2 ratio with N reflects the increasing dilution of the two boundary equations in a system whose PDE block grows with N. The pointwise error distribution is shown in Figure 1; Hard errors are roughly uniform across the domain, whereas Soft errors peak near the boundaries. Both methods achieve spectral-like decay (Figure 2), but the Hard curve lies consistently below the Soft curve.

4.3 Poisson 2D — Homogeneous Dirichlet

We solve $-\Delta u = f$ on $[-1, 1]^2$ with homogeneous Dirichlet BCs and reference solution $u = \sin(\pi x)\sin(\pi y)$. Here the distance function is a degree-four polynomial and the soft system contains many more interior than boundary rows. The Hard ansatz uses $D(x, y) = (1 - x^2)(1 - y^2)$ ($n_e = 4$) and $L = 0$. The collocation set contains $N_{\text{int}} = 8192$ interior and $4 \times 64 = 256$ boundary points.

The neuron sweep results are listed in Table 4. Hard-PIELM gives BC residuals of 8.66×10^{-17} across all N, while the Soft BC residual decreases from 2.77×10^{-1} at $N = 50$ to 1.17×10^{-11} at $N = 1000$. The PDE residual ratio grows from 1.8 to 122 over the same range, and the L^2 ratio is 147 at $N = 1000$. With $m_P/m_B = 8192/256 = 32$,

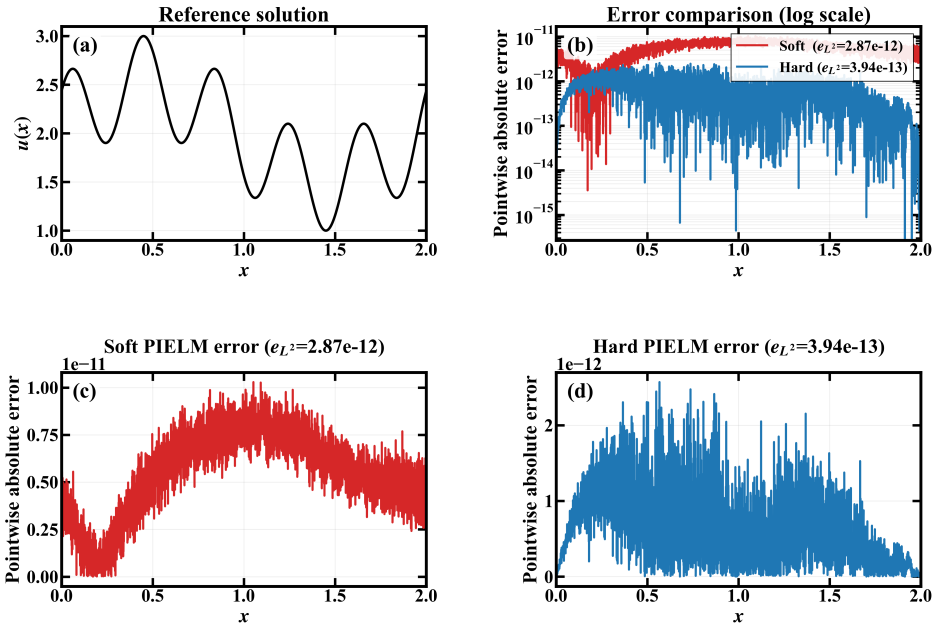
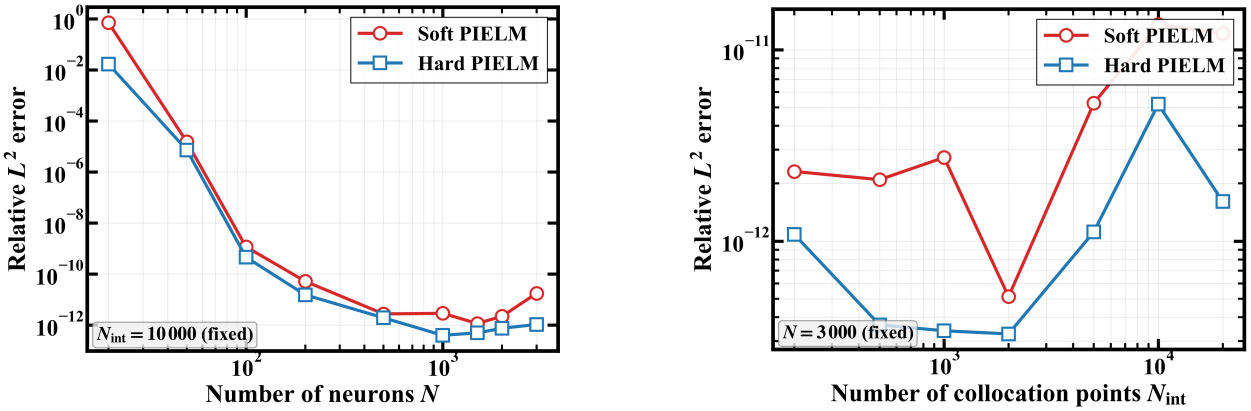


Figure 1: Helmholtz 1D benchmark ($N = 1000$, $N_{\text{int}} = 10{,}000$). (a) Reference solution. (b) Log-scale pointwise error comparison. (c,d) Soft and Hard pointwise errors on a linear scale.



(a) Neuron sweep: relative L^2 error vs. number of neurons N at fixed $N_{\text{int}} = 10{,}000$.

(b) Collocation sweep: relative L^2 error vs. N_{int} at fixed $N = 3000$.

Figure 2: Helmholtz 1D convergence studies.

Table 4: Poisson 2D neuron sweep ($N_{\text{int}} = 8192$).

N	Soft PDE	Hard PDE	PDE ratio	Soft BC	Hard BC	Soft L^2	Hard L^2	L^2 ratio
50	7.83e−01	4.41e−01	1.8	2.77e−01	8.66e−17	2.82e−01	7.40e−03	38
200	5.29e−03	3.51e−04	15	8.47e−04	8.66e−17	6.39e−04	2.45e−06	261
500	1.13e−06	3.56e−08	32	5.86e−08	8.66e−17	3.44e−08	1.26e−10	273
1000	1.79e−09	1.47e−11	122	1.17e−11	8.66e−17	7.86e−12	5.35e−14	147

the row-count ratio already pushes the generalized eigenvalues well above unity; combined with second-derivative amplification from the Laplacian, the boundary block carries limited weight in the Soft normal equation. The increasing PDE ratio with N reflects that additional neurons add degrees of freedom devoted entirely to interior accuracy in the Hard system, while the Soft system must allocate part of this capacity to boundary approximation. The log-ratio error map in Figure 3 shows that the Soft–Hard discrepancy is distributed throughout the domain interior rather than being confined to the boundary. Convergence behavior is reported in Figure 4.

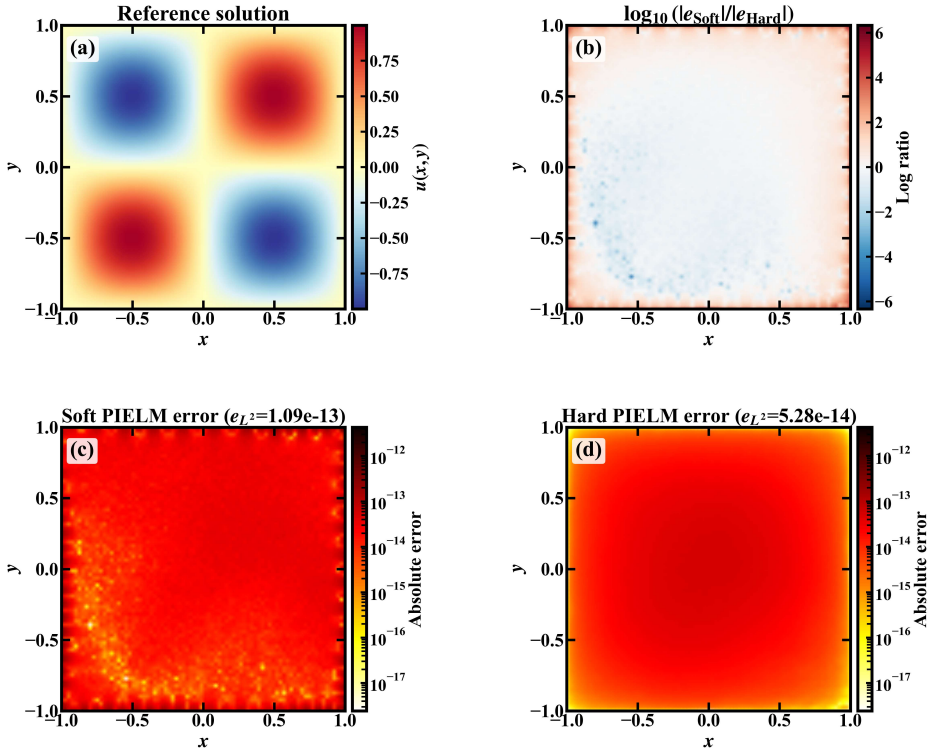
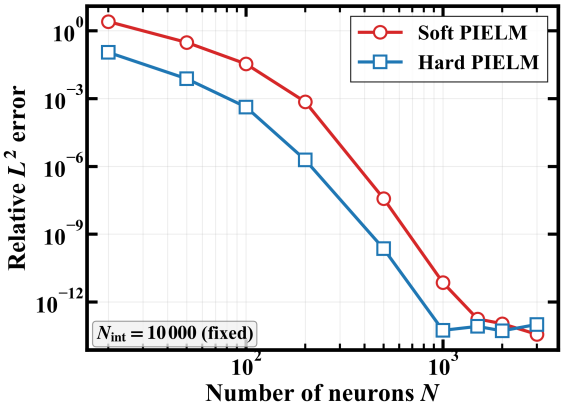
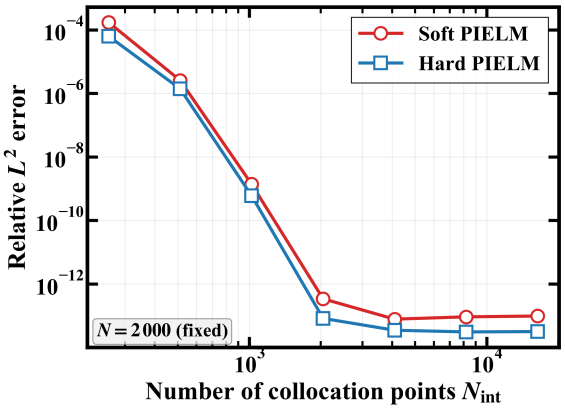


Figure 3: Poisson 2D benchmark ($N = 2000$, $N_{\text{int}} = 10{,}000$). (a) Reference solution. (b) Log-ratio map $\log_{10}(|e_{\text{Soft}}|/|e_{\text{Hard}}|)$. (c,d) Pointwise absolute errors on a common logarithmic color scale.



(a) Neuron sweep: relative L^2 error vs. N at fixed $N_{\text{int}} = 10{,}000$.



(b) Collocation sweep: relative L^2 error vs. N_{int} at fixed $N = 2000$.

Figure 4: Poisson 2D convergence studies.

4.4 L-Shape Poisson — Non-Convex Domain

The Poisson equation $-\Delta u = f$ is solved on the L-shaped domain $\Omega = [-1, 1]^2 \setminus [0, 1]^2$ with $u = 0$ on $\partial\Omega$ and reference solution $u = \sin(\pi x)\sin(\pi y)$. The L-shape has six boundary segments requiring 384 boundary collocation

points. The Hard distance function is $D(x, y) = x\, y\, (1 - x^2)(1 - y^2)$, a polynomial of degree $n_e = 5$ that vanishes on all boundary segments. The ELM weight scale is $s = 3$; $N_{\text{int}} = 10{,}000$.

Table 5: L-shape Poisson neuron sweep ($N_{\text{int}} = 10{,}000$).

N	Soft PDE	Hard PDE	PDE ratio	Soft BC	Hard BC	Soft L^2	Hard L^2	L^2 ratio
100	6.90e−02	1.94e−03	36	2.02e−02	7.04e−17	1.80e−02	1.04e−05	1730
500	1.13e−07	3.42e−10	330	5.12e−09	7.04e−17	4.86e−09	7.99e−13	6079
1000	6.32e−11	7.47e−13	85	3.10e−12	7.04e−17	3.46e−12	2.42e−15	1430

Table 5 presents the results. The non-convex geometry introduces 384 boundary collocation points across six segments in the Soft system, while the Hard system uses the polynomial distance factor $D = xy(1 - x^2)(1 - y^2)$. At $N = 100$, the L^2 ratio is 1730; at $N = 500$, it exceeds 6000. At $N = 1000$, the reported relative L^2 error of Hard-PIELM is 2.42×10^{-15} on the evaluation grid. Compared with the convex Poisson benchmark (L^2 ratio 147 at $n_e = 4$), the L-shape ratio increases to 1430 at the same $N = 1000$. Two factors contribute: the higher polynomial degree ($n_e = 5$ versus 4) makes the Soft basis work harder to represent the boundary zero structure, and the additional 128 boundary points further dilute the relative BC weight in the stacked normal equation. Near the re-entrant corner at the origin, D vanishes as $O(xy)$, creating a second-order zero; however, because the reference solution $\sin(\pi x)\sin(\pi y)$ also vanishes as $O(xy)$ at the origin, the quotient g_D remains smooth and Proposition 2 applies without requiring local enrichment. The pointwise error maps in Figure 5 show that the Soft error concentrates along the boundary segments and is elevated near the re-entrant corner, whereas the Hard error is uniformly small throughout the interior (see also Figure 6).

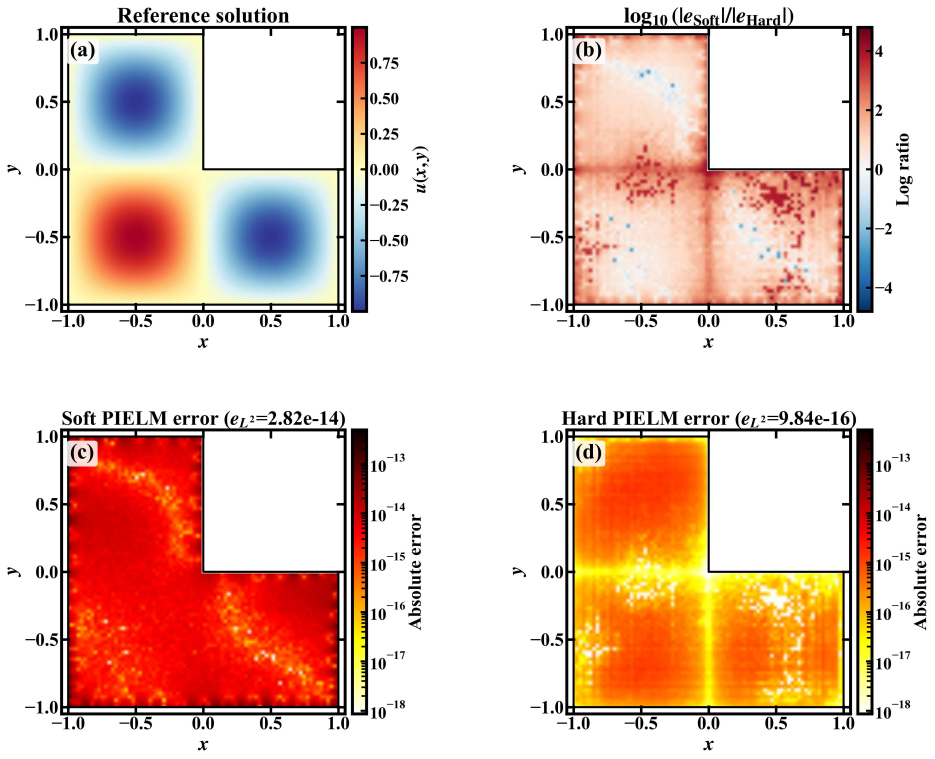
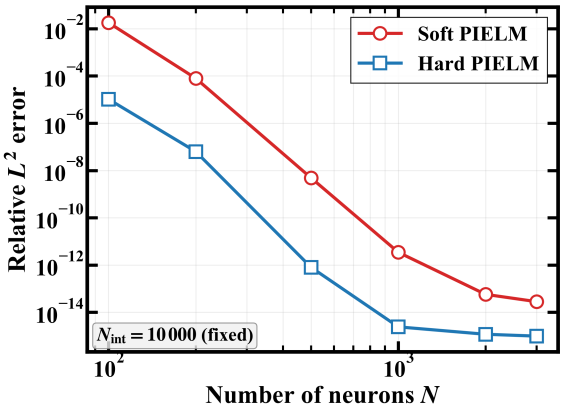


Figure 5: L-shape Poisson benchmark ($N = 3000$, $N_{\text{int}} = 10{,}000$). (a) Reference solution on $[-1, 1]^2 \setminus [0, 1]^2$. (b) Log-ratio map. (c,d) Pointwise absolute errors on a common logarithmic color scale.

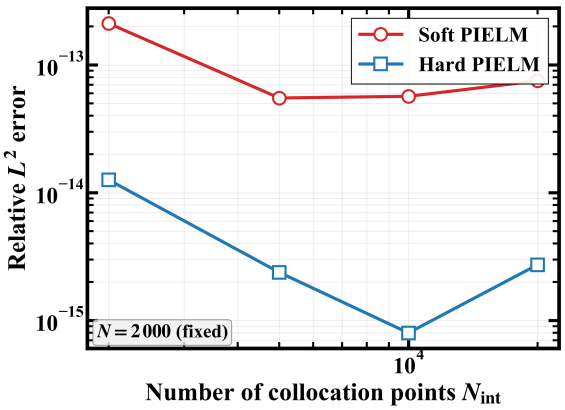
4.5 Navier–Stokes Kovasznay Flow

The steady incompressible Navier–Stokes equations are considered:

$$\mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nu\, \Delta \mathbf{u} = 0, \qquad \nabla \cdot \mathbf{u} = 0. \tag{4.6}$$



(a) Neuron sweep: relative L^2 error vs. N at fixed $N_{\text{int}} = 10{,}000$.



(b) Collocation sweep: relative L^2 error vs. N_{int} at fixed $N = 2000$.

Figure 6: L-shape Poisson convergence studies.

These equations are solved on $[-0.5, 1.0] \times [-0.5, 1.5]$ at Reynolds number $Re = 40$ ($\nu = 1/Re$). The Kovasznay reference solution is

$$u = 1 - e^{\lambda x}\cos(2\pi y), \quad v = \frac{\lambda}{2\pi} e^{\lambda x}\sin(2\pi y), \quad p = \frac{1}{2}(1 - e^{2\lambda x}), \tag{4.7}$$

where $\lambda = Re/2 - \sqrt{Re^2/4 + 4\pi^2}$. Non-homogeneous Dirichlet BCs for all three fields are imposed on all boundaries via TFI lifting (Strategy IV, Eq. (3.12)), with $D = (x - x_a)(x_b - x)(y - y_a)(y_b - y)$ ($n_e = 4$). Each field (u, v, p) receives its own lifting function L_u, L_v, L_p constructed from its boundary data. The nonlinearity is handled by Picard iteration (40 steps). The ELM weight scale is $s = 5$; $N_{\text{int}} = 5000$; 32 points per boundary edge.

Table 6: Navier–Stokes Kovasznay neuron sweep ($N_{\text{int}} = 5000$).

N	Soft PDE	Hard PDE	PDE ratio	Soft BC	Hard BC	Soft L^2	Hard L^2	L^2 ratio
20	2.95e−02	2.46e−01	0.12	5.11e−01	1.53e−16	6.78e−01	6.34e−02	11
100	3.86e−02	6.05e−03	6.4	2.19e−01	1.53e−16	3.07e−01	3.95e−04	777
200	9.43e−03	1.90e−04	50	3.61e−02	1.53e−16	6.99e−02	8.08e−06	8653
500	1.02e−04	2.23e−06	46	1.91e−05	1.53e−16	1.53e−05	7.46e−08	206

As shown in Table 6, TFI lifting keeps the combined boundary residual at 1.53×10^{-16} for the three-field Hard system. The Soft BC residual is 5.11×10^{-1} at $N = 20$ and remains 2.19×10^{-1} at $N = 100$. The L^2 ratio reaches 8653 at $N = 200$ and is 206 at $N = 500$. Each of u, v, and p receives its own TFI boundary extension before the homogeneous distance factor is applied. At $N = 20$, the Soft system has a smaller PDE residual but much larger boundary error—a direct illustration of the PDE–BC trade-off. Over 40 Picard iterations, boundary mismatch in the Soft iterates enters each subsequent right-hand side through the nonlinear advection term, compounding the residual; Hard-PIELM avoids this because $u_k|_{\partial\Omega} = g$ at every step. The three-field error maps in Figure 7 reveal that boundary mismatch propagates to all three fields through the Picard coupling. Figure 8 reports the convergence studies.

4.6 Wave Equation 1D — TFC Multi-Constraint

The one-dimensional wave equation

$$u_{tt} = c^2\, u_{xx}, \qquad (x, t) \in [0, 1] \times [0, 2], \qquad c^2 = \frac{1}{16\pi^2}, \tag{4.8}$$

is solved with initial conditions $u(x, 0) = \sin(4\pi x)$, $u_t(x, 0) = 0$ and homogeneous Dirichlet BCs $u(0, t) = u(1, t) = 0$. The Hard ansatz uses $L(x, t) = \sin(4\pi x)$ and $D(x, t) = x(1 - x)\, t^2$ (Strategy III, $n_e = 4$), which simultaneously satisfies both ICs and both BCs. The Soft system must accommodate five constraint types (PDE + 2 BCs + 2 ICs), making this case a direct test of the multi-constraint setting. $N_{\text{int}} = 10{,}000$.

Table 7 records the results. This case tests the TFC component because the ansatz must satisfy two spatial BCs, displacement IC, and velocity IC simultaneously. At $N = 1000$, the Soft BC residual is 5.56×10^{-3} and the Soft L^2 error is 2.36×10^{-3}, while Hard gives $L^2 = 2.01 \times 10^{-8}$ with IC and BC residuals at zero or roundoff level. The Hard

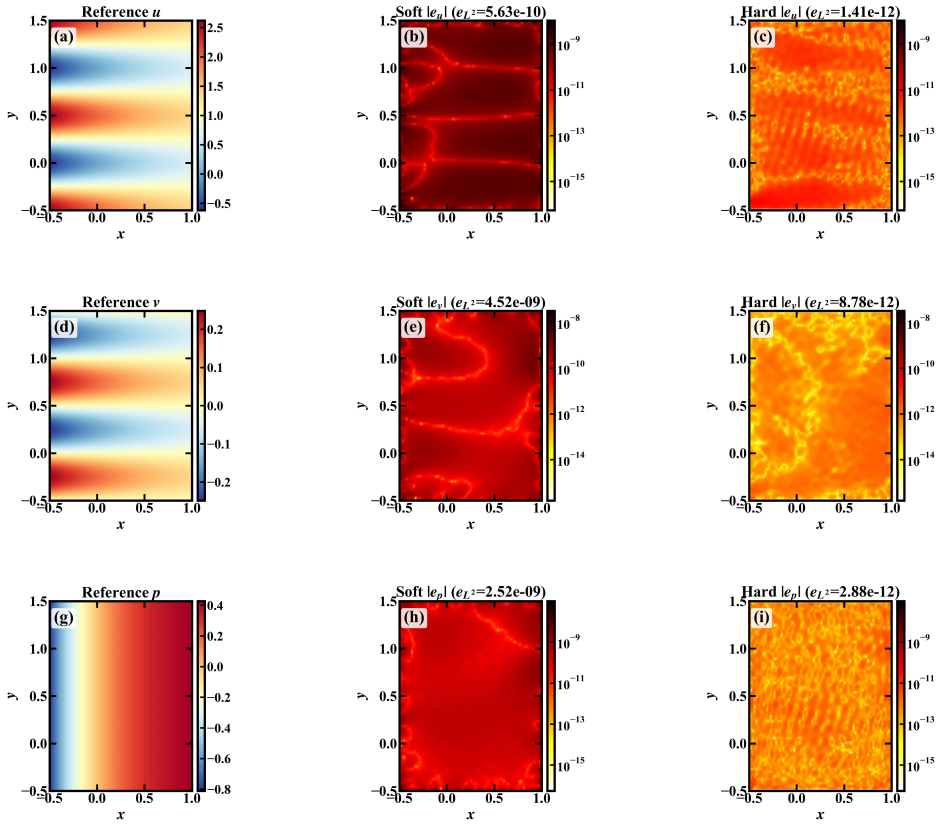
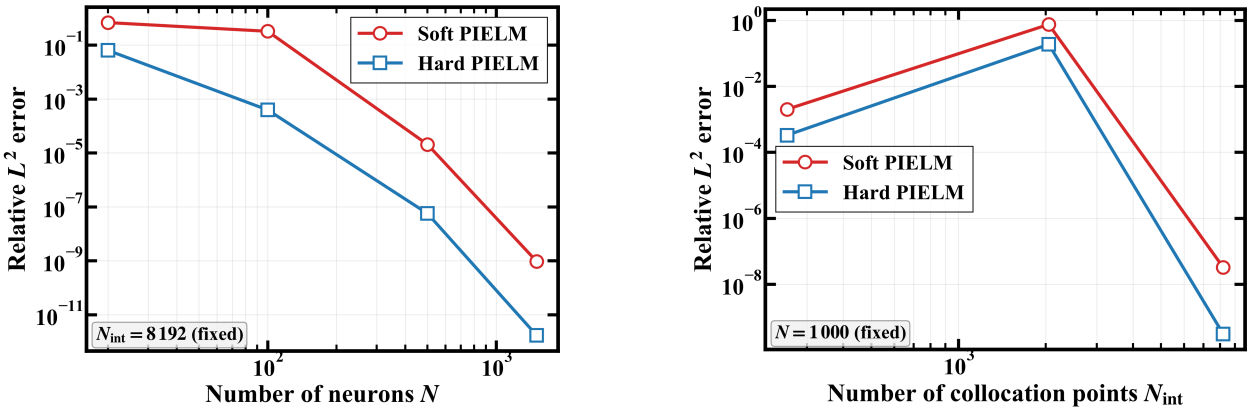


Figure 7: Navier–Stokes Kovasznay flow ($N = 1500$, $N_{\text{int}} = 8192$, 40 Picard iterations). Rows show u, v, and p; columns show the reference solution, Soft error, and Hard error. Error maps use a common logarithmic scale within each row.



(a) Neuron sweep: combined relative L^2 error vs. N at fixed $N_{\text{int}} = 8192$.

(b) Collocation sweep: combined relative L^2 error vs. N_{int} at fixed $N = 1000$.

Figure 8: Navier–Stokes Kovasznay convergence studies.

Table 7: Wave equation 1D neuron sweep ($N_{\text{int}} = 10{,}000$).

N	Soft PDE	Hard PDE	Soft BC	Hard BC	Soft IC	Hard IC	Soft L^2	Hard L^2	L^2 ratio
200	3.12e−02	1.92e−02	5.96e−01	2.45e−16	8.00e−03	**0**	3.38e−01	7.72e−03	44
500	2.42e−04	1.44e−05	1.77e−02	2.45e−16	4.34e−05	**0**	6.84e−03	3.99e−06	1713
1000	9.68e−06	1.87e−07	5.56e−03	2.45e−16	6.02e−07	**0**	2.36e−03	2.01e−08	1.2×10^5

IC residual is zero because the distance function and its time derivative both vanish at $t = 0$:

$$\begin{aligned} D(x,0) = x(1-x)\cdot 0^2 = 0 \quad &\Longrightarrow \quad u_H(x,0) = L(x,0) = u_0(x), \\ D_t(x,0) = 2\,x(1-x)\cdot 0 = 0 \quad &\Longrightarrow \quad \partial_t u_H(x,0) = \partial_t L(x,0) = 0. \end{aligned} \tag{4.9}$$

The wave equation shares $n_e = 4$ with the Poisson 2D benchmark, but the additional IC constraints make the Soft system a five-block least-squares fit rather than a simple PDE–BC fit. Consequently, the L^2 ratio at $N = 1000$ is 1.2×10^5—three orders of magnitude larger than the Poisson 2D ratio of 147 at the same n_e—because the IC rows further inflate the total row count $m_P + m_B$ while contributing objectives that are structurally different from the PDE residual. The spatiotemporal error map in Figure 9 shows that the Soft error grows with increasing t: at $t = 0$ both methods start from the prescribed IC, but as t advances the Soft solution accumulates phase and amplitude errors that the Hard solution avoids because its constraint structure is enforced analytically for all t. The neuron sweep and collocation sweep in Figure 10 confirm that the Hard curve converges monotonically, whereas the Soft curve plateaus at large N due to the persistent BC residual floor.

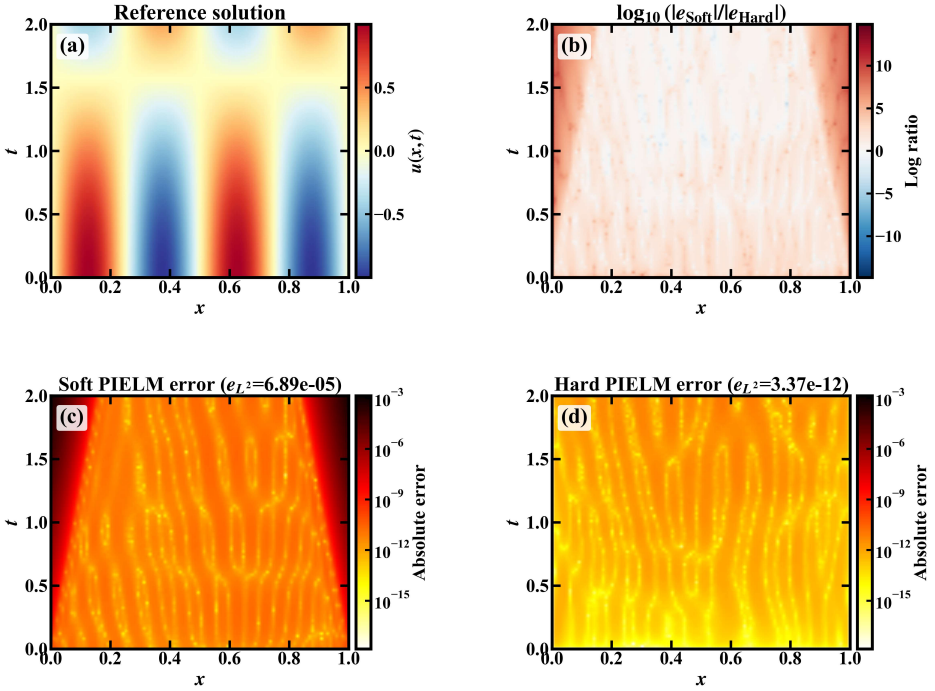


Figure 9: Wave 1D benchmark ($N = 3000$, $N_{\text{int}} = 10{,}000$). (a) Reference solution on $[0,1] \times [0,2]$. (b) Log-ratio map. (c,d) Pointwise absolute errors on a common logarithmic color scale.

4.7 Pentagram Nonlinear Reaction

We solve the nonlinear reaction–diffusion equation

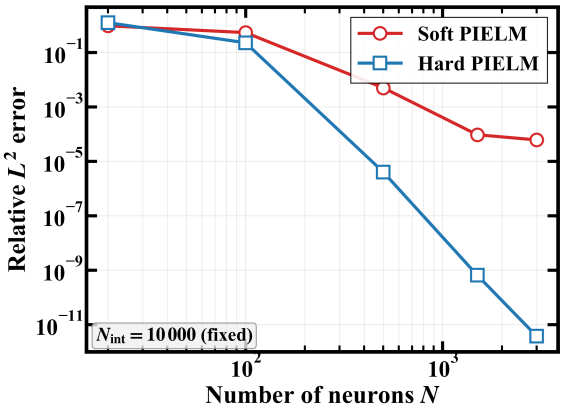
$$-\Delta u + u^3 = f \tag{4.10}$$

on a regular pentagram domain $\Omega_\star$ with 10 edges and homogeneous Dirichlet BCs. The distance function is the product of signed distances to all ten edges,

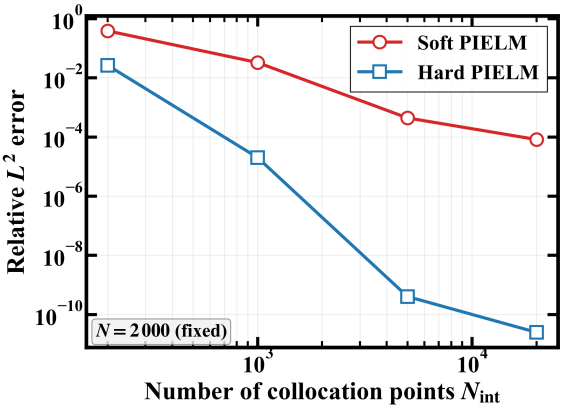
$$D = \prod_{i=0}^{9} \ell_i, \qquad D_{\text{norm}} = D/D_{\max}, \qquad n_e = 10, \tag{4.11}$$

and the reference solution is $u = D_{\text{norm}} \cdot e^{x+y}$. The $D_{\max}$ for this geometry is 7.94×10^{-6}, necessitating normalization. The cubic nonlinearity is handled by 30 Picard iterations. The ELM weight scale is $s = 3$; $N_{\text{int}} = 10{,}000$.

Table 8 gives the largest separation among the reported benchmarks. The Soft basis must represent the degree-ten boundary zero structure directly, whereas Hard-PIELM inserts this structure analytically through D. At $N = 50$, the Soft L^2 error is 3.15×10^1, while Hard gives 5.57×10^{-5}. At $N = 200$, the L^2 ratio is 3.7×10^8; at $N = 1000$, it remains 1.0×10^7. The PDE residual ratio exceeds 10^5 across all listed N, and the Picard iteration provides an additional pathway for Soft boundary error to affect the interior solve. The extreme separation here has a specific



(a) Neuron sweep: relative L^2 error vs. N at fixed $N_{\text{int}} = 10{,}000$.



(b) Collocation sweep: relative L^2 error vs. N_{int} at fixed $N = 2000$.

Figure 10: Wave 1D convergence studies.

Table 8: Pentagram nonlinear reaction neuron sweep ($N_{\text{int}} = 10{,}000$).

N	Soft PDE	Hard PDE	PDE ratio	Soft BC	Hard BC	Soft L^2	Hard L^2	L^2 ratio
50	2.43e+04	3.38e−02	7.2×10^5	2.10e+01	5.67e−34	3.15e+01	5.57e−05	5.7×10^5
200	2.37e+00	1.56e−06	1.5×10^6	6.37e−01	3.81e−38	5.18e−01	1.40e−09	3.7×10^8
500	1.16e−04	6.71e−11	1.7×10^6	1.01e−05	2.12e−42	7.17e−06	2.76e−14	2.6×10^8
1000	3.32e−06	2.81e−11	1.2×10^5	1.25e−07	7.80e−43	9.15e−08	9.02e−15	1.0×10^7

algebraic origin: with $D_{\max} = 7.94 \times 10^{-6}$, the unnormalized D values inside the pentagram are tiny, so the Soft basis must approximate a function that is $O(10^{-6})$ in the interior while simultaneously vanishing on the boundary—a task that random tanh features are poorly suited for without explicit polynomial structure. After normalization, the Hard quotient $g_D = u^*/D_{\text{norm}}$ is simply e^{x+y}, a smooth function that the ELM basis approximates efficiently. Moreover, because the cubic nonlinearity couples each Picard iterate through the linearized term $h'(u_k)\, u_{k+1}$, boundary errors in the Soft iterates propagate into subsequent right-hand sides, providing an additional amplification route that is absent in the linear benchmarks. The pointwise error map in Figure 11 confirms that the Soft error is concentrated along the narrow arms of the pentagram where D is smallest. Figure 12 shows the convergence trends.

4.8 Flower Domain Nonlinear Reaction

The nonlinear equation

$$-\varepsilon\, \Delta u + u^2 = f, \qquad \varepsilon = 1, \tag{4.12}$$

is solved on a five-petal flower domain Ω_{flower} with curved boundary and homogeneous Dirichlet BCs. The reference solution is $u = D_{\text{norm}} \cdot e^{x+y}$, where $D = 1 - r^6 + \text{Re}[(x+iy)^5]$ is a level-set function (Strategy II, effective $n_e = 6$). Picard iteration (30 steps) handles the u^2 nonlinearity. $N_{\text{int}} = 10{,}000$; $s = 3$.

Table 9: Flower domain neuron sweep ($N_{\text{int}} = 10{,}000$).

N	Soft PDE	Hard PDE	PDE ratio	Soft BC	Hard BC	Soft L^2	Hard L^2	L^2 ratio
100	7.53e−01	5.07e−03	149	3.45e−01	1.14e−17	1.22e−01	5.85e−06	2.1×10^4
500	4.39e−06	8.75e−10	5020	4.96e−07	1.25e−24	1.11e−07	6.01e−13	1.8×10^5
1000	6.66e−09	5.97e−11	112	7.01e−10	8.75e−26	1.23e−10	3.30e−14	3720

Table 9 complements the pentagram benchmark by replacing edge products with the level-set function $D = 1 - r^6 + \text{Re}[(x+iy)^5]$, for which $\Delta D = -36r^4$ because the harmonic term vanishes under Δ. The Hard BC residuals range from 1.14×10^{-17} at $N = 100$ to 8.75×10^{-26} at $N = 1000$, and the L^2 ratio ranges from 3720 to 1.8×10^5. The pentagram and flower cases have very different $D_{\max}$ values (7.94×10^{-6} and 1.067, respectively), but both are handled by the same normalization step and quotient approximation viewpoint. This comparison reveals that the dominant factor in the Soft–Hard gap is the complexity of the boundary zero structure rather than the magnitude of D: the flower has $D_{\max} \approx 1$ yet still shows a four-order ratio, because the Soft basis must capture the five-fold rotational symmetry of D near the petals. A distinctive feature of the level-set construction is that the Leibniz cross

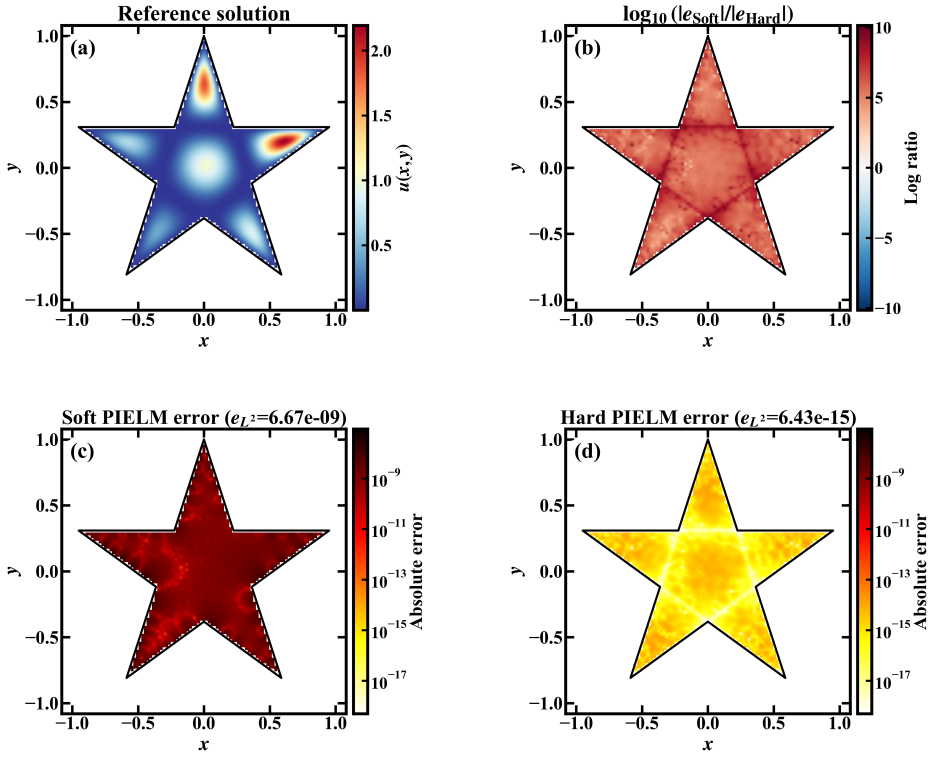
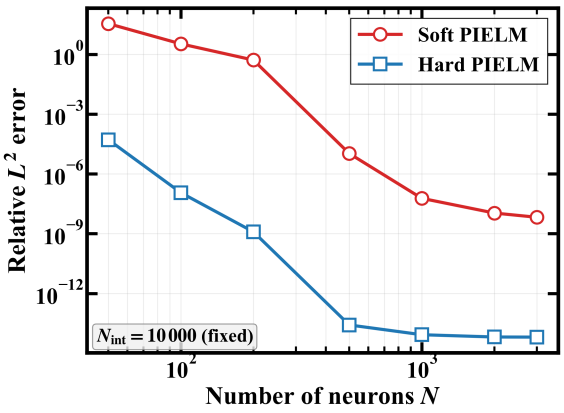
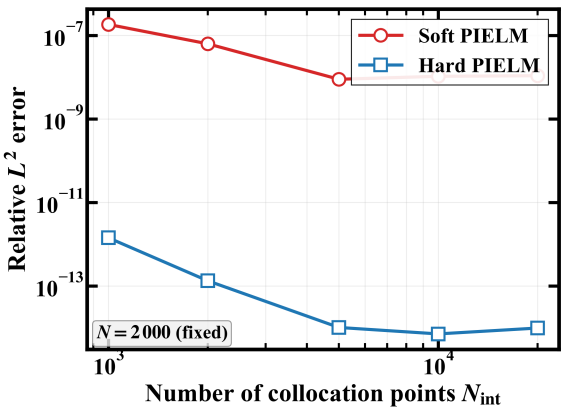


Figure 11: Pentagram nonlinear reaction benchmark ($N = 3000$, $N_{\text{int}} = 10{,}000$, 30 Picard iterations). (a) Reference solution on the pentagram domain. (b) Log-ratio map. (c,d) Pointwise absolute errors on a common logarithmic color scale.



(a) Neuron sweep: relative L^2 error vs. N at fixed $N_{\text{int}} = 10{,}000$.



(b) Collocation sweep: relative L^2 error vs. N_{int} at fixed $N = 2000$.

Figure 12: Pentagram nonlinear reaction convergence studies.

term $2\nabla D \cdot \nabla \phi_j$ contributes anisotropically—the gradient ∇D rotates around the boundary curve, probing different directional components of each basis function. Figure 13 shows that the Soft error peaks at the petal tips where D varies most steeply, while the Hard error remains below 10^{-13} uniformly. Convergence curves are given in Figure 14.

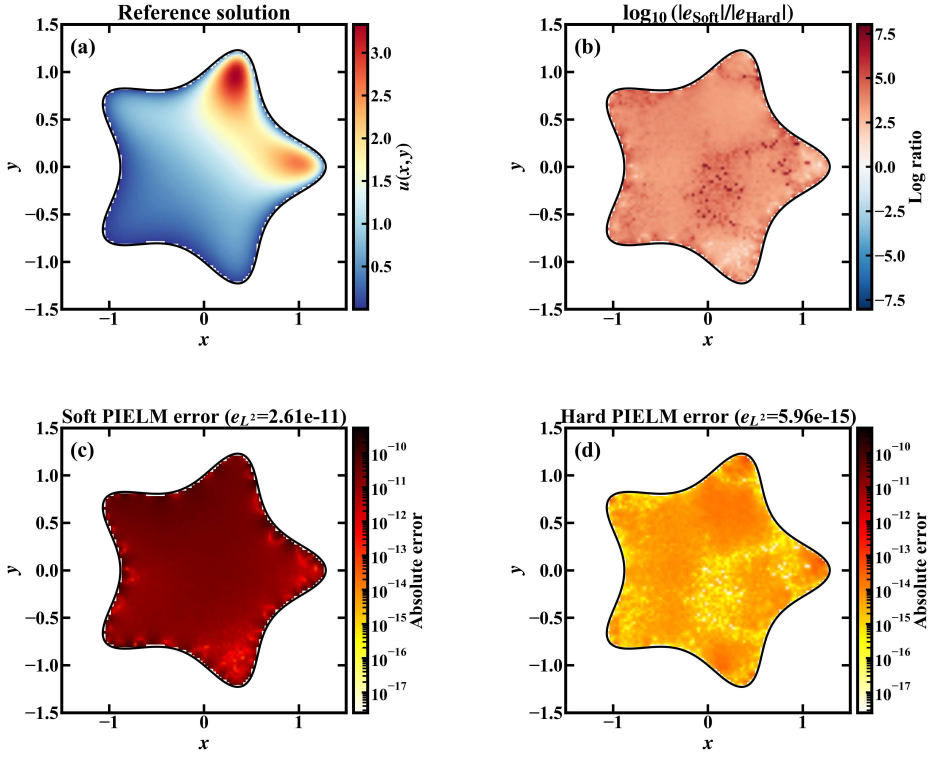
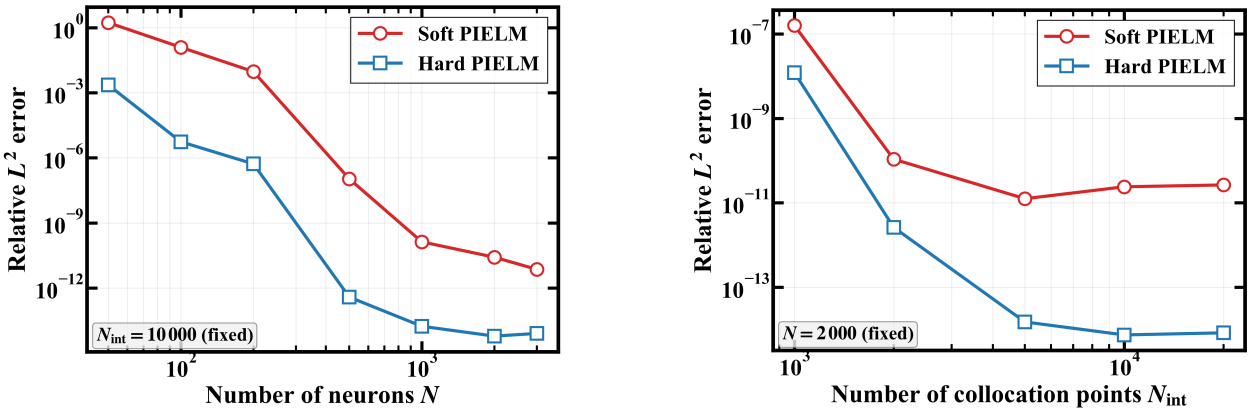


Figure 13: Flower domain nonlinear reaction benchmark ($N = 2000$, $N_{\text{int}} = 10{,}000$, 30 Picard iterations). (a) Reference solution on the five-petal flower domain. (b) Log-ratio map. (c,d) Pointwise absolute errors on a common logarithmic color scale.



(a) Neuron sweep: relative L^2 error vs. N at fixed $N_{\text{int}} = 10{,}000$.

(b) Collocation sweep: relative L^2 error vs. N_{int} at fixed $N = 2000$.

Figure 14: Flower domain nonlinear reaction convergence studies.

4.9 Computational Cost

Table 10 reports the wall-clock times for the listed neuron-sweep configuration in each benchmark.

The Hard-to-Soft time ratios range from 0.88 (Poisson 2D) to 1.15 (wave equation, where the $D\phi_j$ product expansion adds marginal cost to each Picard-free linear solve). In five of the seven cases, Hard is faster than Soft, because Soft's

Table 10: Computational cost comparison at the listed neuron-sweep configuration.

Benchmark	Soft time (s)	Hard time (s)	Hard/Soft
Helmholtz 1D	1.80	1.61	0.89
Poisson 2D	3.42	3.01	0.88
L-shape Poisson	4.39	4.34	0.99
Kovasznay flow	820.9	835.0	1.02
Wave 1D	6.75	7.75	1.15
Pentagram NL react.	160.4	154.0	0.96
Flower NL react.	33.65	30.66	0.91

system matrix has m_B additional rows. The dominant expense is the least-squares solve via SVD, whose complexity is

$$\text{Soft: } O\big((m_P + m_B)\, N^2\big), \qquad \text{Hard: } O\big(m_P\, N^2\big). \tag{4.13}$$

For the nonlinear cases, the Picard iteration cost dominates: the Kovasznay benchmark requires 40 iterations of a three-field system (~ 835 s), while the pentagram requires 30 iterations with a single field (~ 154 s). Once D and its analytic derivatives are available, the hard formulation has a comparable wall-clock cost to the soft formulation in these benchmarks.

5 Conclusions

This paper has developed a hard-constrained formulation for physics-informed extreme learning machines in which Dirichlet boundary data are incorporated directly into the trial function through a lifting term and a boundary-vanishing distance function. The analysis identifies a structural imbalance in the conventional soft-constrained least-squares system: boundary rows and interior residual rows compete within a single coefficient solve, and this competition can weaken boundary influence along directions dominated by the PDE block. The proposed formulation removes this competition while retaining the one-shot random-feature least-squares structure of PIELM. It also provides practical distance-function constructions for polygonal, curved, spatiotemporal, and multi-field Dirichlet settings. Across seven benchmark problems with identical random bases and collocation layouts, the hard-constrained solver achieves zero or roundoff-level boundary residuals, consistently reduces relative errors, and maintains comparable computational cost.

The present framework is nevertheless limited to Dirichlet-type constraints. Extensions to Neumann or Robin conditions would require additional constraint mechanisms, such as mixed hard–soft formulations or factors designed to control derivative traces. The effectiveness of the method also depends on the regularity of the quotient field induced by the distance-function representation. Corner singularities, boundary layers, non-smooth boundary data, or non-smooth distance functions at edge intersections may reduce the achievable approximation quality unless local enrichment or partition-of-unity corrections are incorporated. In higher spatial dimensions, analytic distance-function construction can become difficult because edge-product representations grow in algebraic complexity and closed-form implicit descriptions may not be available.

Future work will focus on extending the formulation to more general boundary conditions, developing automated distance-function construction for complex geometries, and combining the hard-constrained PIELM framework with adaptive collocation and rank-revealing linear solvers for higher-dimensional and singular problems.

Declarations

Data availability

The experimental data and source code generated during this study are available from the corresponding author upon reasonable request.

Funding

This work was supported by the S&T Program of Hebei (252Q0701D), the Research Projects of Shijiazhuang University (24XJ010), and the Shijiazhuang University Doctoral Research Fund Project (18BS013).

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author contribution

Ruonan Zhai: Methodology, Software, Validation, Writing. **Xiaoyuan Shi**: Methodology, Software. **Xiong Xiong**: Conceptualization, Methodology, Software, Supervision, Writing. **Zhuo Zhang**: Investigation, Writing. **Maowei You**: Investigation, Validation.

References

[1] Steven Owen, Nathan Brown, Nikos Chrisochoides, Rao Garimella, Xianfeng Gu, Franck Ledoux, Na Lei, Roshan Quadros, Navamita Ray, Nicolas Winovich, et al. A survey of AI methods for geometry preparation and mesh generation in engineering simulation. *arXiv preprint arXiv:2512.23719*, 2025.

[2] M. W. M. G. Dissanayake and Nhan Phan-Thien. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 10(3):195–201, 1994.

[3] 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.

[4] Weinan E and Bing Yu. The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[5] Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

[6] 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, 2021.

[7] Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. NSFnets (Navier–Stokes flow nets): Physics-informed neural networks for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 426:109951, 2021.

[8] Peng Shu Ng, Elisa Y. M. Ang, Peng Cheng Wang, and Teng Yong Ng. Physics-informed neural network modelling for deep-water linear wave propagation. *Engineering with Computers*, 42(1):27, 2026.

[9] 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, 2021.

[10] 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.

[11] Xiong Xiong, Kang Lu, Zhuo Zhang, Zheng Zeng, Sheng Zhou, Rongchun Hu, and Zichen Deng. High-frequency flow field super-resolution via physics-informed hierarchical adaptive fourier feature networks. *Physics of Fluids*, 37(9), 2025.

[12] Xiong Xiong, Kang Lu, Zhuo Zhang, Zheng Zeng, Sheng Zhou, Zichen Deng, and Rongchun Hu. J-pikan: A physics-informed kan network based on jacobi orthogonal polynomials for solving fluid dynamics. *Communications in Nonlinear Science and Numerical Simulation*, 152:109414, 2025.

[13] 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*, volume 34, pages 26548–26560, 2021.

[14] Xiong Xiong, Zhuo Zhang, Rongchun Hu, Chen Gao, and Zichen Deng. Separated-variable spectral neural networks: a physics-informed learning approach for high-frequency pdes. *arXiv preprint arXiv:2508.00628*, 2025.

[15] Hang Zhou, Zhuhong Wang, Geping Qi, and Yisheng Wang. IPIRNNs: Integration-based physics informed randomized neural networks for solving stiff ODEs. *Neurocomputing*, 662:131911, 2026.

[16] Levi D. McClenny and Ulisses M. Braga-Neto. Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474:111722, 2023.

[17] Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2020.

[18] Yixin Huang, Haizhou Liu, Yang Zhao, and Min Fei. Chebyshev spectral approximation-based physics-informed neural network for solving higher-order nonlinear differential equations. *Engineering with Computers*, 41(2):1191–1210, 2025.

[19] 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.

[20] Subarna Khanra, Vijay Kumar Kukreja, and Indu Bala. Physics-informed neural networks for differential equation solutions: A comprehensive review. *Neurocomputing*, 680:133317, 2026.

[21] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3):489–501, 2006.

[22] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(2):513–529, 2012.

[23] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[24] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, pages 1177–1184, 2007.

[25] Jingrun Chen, Xurong Chi, Weinan E, and Zhouwang Yang. Bridging traditional and machine learning-based algorithms for solving PDEs: The random feature method. *Journal of Machine Learning*, 1(3):268–298, 2022.

[26] Lukas Gonon. Random feature neural networks learn Black–Scholes type PDEs without curse of dimensionality. *Journal of Machine Learning Research*, 24(189):1–51, 2023.

[27] Jingrun Chen, Rui Du, Ruijie Li, and Zhe Wang. Solving partial differential equations with random feature models. *arXiv preprint arXiv:2501.00288*, 2025.

[28] Vedant Puri, Aviral Prakash, Levent Burak Kara, and Yongjie Jessica Zhang. SNF-ROM: Projection-based nonlinear reduced order modeling with smooth neural fields. *Journal of Computational Physics*, 532:113957, 2025.

[29] Vedant Puri, Aditya Joglekar, Sri Datta Ganesh Bandreddi, Kevin Ferguson, Yu-hsuan Chen, Yongjie Jessica Zhang, and Levent Burak Kara. FLARE: Fast low-rank attention routing engine. *arXiv preprint arXiv:2508.12594*, 2025.

[30] Vikas Dwivedi and Balaji Srinivasan. Physics informed extreme learning machine (PIELM)—A rapid method for the numerical solution of partial differential equations. *Neurocomputing*, 391:96–118, 2020.

[31] Francesco Calabrò, Gianluca Fabiani, and Constantinos Siettos. Extreme learning machine collocation for the numerical solution of elliptic PDEs with sharp gradients. *Computer Methods in Applied Mechanics and Engineering*, 387:114188, 2021.

[32] Gianluca Fabiani, Francesco Calabrò, Lucia Russo, and Constantinos Siettos. Numerical solution and bifurcation analysis of nonlinear partial differential equations with extreme learning machines. *Journal of Scientific Computing*, 89:44, 2021.

[33] 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.

[34] 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.

[35] Suchuan Dong and Zongze Li. A modified batch intrinsic plasticity method for pre-training the random coefficients of extreme learning machines. *Journal of Computational Physics*, 445:110585, 2021.

[36] Yiran Wang and Suchuan Dong. An extreme learning machine-based method for computational PDEs in higher dimensions. *Computer Methods in Applied Mechanics and Engineering*, 418:116578, 2024.

[37] Naxian Ni and Suchuan Dong. Numerical computation of partial differential equations by hidden-layer concatenated extreme learning machine. *Journal of Scientific Computing*, 95:35, 2023.

[38] Daiwei Dong, Jiaqing Kou, Wei Suo, and Weiwei Zhang. Residual cascade extreme learning machine for solving partial differential equations. *Acta Mechanica Sinica*, 42(5):725023, 2026.

[39] Rishi Mishra, Smriti, Ganapathy Krishnamurthi, Balaji Srinivasan, and Sundararajan Natarajan. Eig-PIELM: An efficient mesh-free method for eigenvalue problems using physics-informed extreme learning machines. *Computer Methods in Applied Mechanics and Engineering*, 451:118674, 2026.

[40] He Yang, Fei Ren, Francesco Calabrò, Hai-Sui Yu, Xiaohui Chen, and Pei-Zhi Zhuang. Physics-informed extreme learning machine (PIELM): Opportunities and challenges. *arXiv preprint arXiv:2510.24577*, 2025.

[41] 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.

[42] Isaac E. Lagaris, Aristidis C. Likas, and Dimitrios G. Papageorgiou. Neural-network methods for boundary value problems with irregular boundaries. *IEEE Transactions on Neural Networks*, 11(5):1041–1049, 2000.

[43] Kevin Stanley McFall and James Robert Mahan. Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions. *IEEE Transactions on Neural Networks*, 20(8):1221–1233, 2009.

[44] Jens Berg and Kaj Nyström. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing*, 317:28–41, 2018.

[45] Hailong Sheng and Chao Yang. PFNN: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries. *Journal of Computational Physics*, 428:110085, 2021.

[46] Suchuan Dong and Naxian Ni. A method for representing periodic functions and enforcing exactly periodic boundary conditions with deep neural networks. *Journal of Computational Physics*, 435:110242, 2021.

[47] Natarajan 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.

[48] Songming Liu, Zhongkai Hao, Chengyang Ying, Hang Su, Jun Zhu, and Ze Cheng. A unified hard-constraint framework for solving geometrically complex PDEs. In *Advances in Neural Information Processing Systems*, volume 35, pages 20287–20299, 2022.

[49] Stefano Berrone, Claudio Canuto, Moreno Pintore, and Natarajan Sukumar. Enforcing Dirichlet boundary conditions in physics-informed neural networks and variational physics-informed neural networks. *Heliyon*, 9(8):e18820, 2023.

[50] Peng Lan, Jing-Jing Su, Xin-Yan Ma, and Sheng Zhang. Application of improved physics-informed neural networks for nonlinear consolidation problems with continuous drainage boundary conditions. *Acta Geotechnica*, 19(1):495–508, 2024.

[51] Hossein Vasheghani Farahani, Amir Mahdi Tahsini, Ali Nematollahi, and Milad Ghiasi Afzal. A novel physics-informed neural network architecture for solving nonlinear PDEs in fluid mechanics without labeled data. *Engineering with Computers*, 42(2):46, 2026.

[52] Carl Leake and Daniele Mortari. Deep theory of functional connections: A new method for estimating the solutions of partial differential equations. *Machine Learning and Knowledge Extraction*, 2(1):37–55, 2020.

[53] Daniele Mortari. The Theory of Connections: Connecting points. *Mathematics*, 5(4):57, 2017.

[54] Enrico Schiassi, Roberto Furfaro, Carl Leake, Mario De Florio, Hunter Johnston, and Daniele Mortari. Extreme Theory of Functional Connections: A fast physics-informed neural network method for solving ordinary and partial differential equations. *Neurocomputing*, 457:334–356, 2021.

[55] Lu Lu, Raphael Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G Johnson. Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing*, 43(6):B1105–B1132, 2021.

[56] Davide Elia De Falco, Enrico Schiassi, and Francesco Calabrò. Least squares with equality constraints extreme learning machines for the resolution of PDEs. *Journal of Computational Physics*, 547:114553, 2026.

[57] Suchuan Dong and Yuchuan Zhang. Exact enforcement of Dirichlet, Neumann, and Robin boundary conditions on curved quadrilateral domains via transfinite interpolation and the theory of functional connections with extreme learning machines. *arXiv preprint arXiv:2603.21909*, 2026.

[58] Yuchen Xie, Honghang Chi, Haopeng Quan, Yahui Wang, Wei Wang, and Yu Ma. Spectral analysis of hard-constraint PINNs: The spatial modulation mechanism of boundary functions. *arXiv preprint arXiv:2512.23295*, 2025.