Journal of Advanced Computational Engineering. 2026;1(1):35–49.

Journal of Advanced Computational Engineering Original Research

Physics-Informed Neural Networks for Steel Beam Deflection Analysis: Methodology and Applications

K Baharan[1], **H Mirzabozorg**[2,*]

[1]Independent Researcher in Physics-Informed Neural Networks and Scientific Machine Learning, Tehran, Iran

[2]Professor, Civil Engineering Department, K.N. Toosi University of Technology, Tehran, Iran

[*]Correspondence should be addressed to H Mirzabozorg, mirzabozorg@kntu.ac.ir

Received date: October 07, 2025, **Accepted date:** November 11, 2025

Citation: Baharan K, Mirzabozorg H. Physics-Informed Neural Networks for Steel Beam Deflection Analysis: Methodology and Applications. Journal of Advanced Computational Engineering. 2026;1(1):35–49.

Abstract

This paper presents an enhanced Physics-Informed Neural Network (PINN) methodology for structural beam deflection analysis, introducing three methodological contributions that address computational challenges in mechanics: a penalty-based boundary condition enforcement framework, a time-decaying adaptive boundary loss weight, and a regularized representation of concentrated loads via Gaussian approximation. The proposed approaches are validated on cantilever, fully-restrained, and mid-span point loaded beams, demonstrating both accuracy and training efficiency. The adaptive weighting reduced required training iterations by approximately 37% compared to static equal-weight baselines, achieving training losses on the order of 10−10 and relative L2 errors as low as 0.056% for concentrated load cases.

Keywords: Physics-informed neural networks, Steel beam deflection, Structural mechanics, Gaussian regularization, Adaptive weighting

Introduction

Accurate prediction of beam deflection is a fundamental problem in structural mechanics with applications ranging from building and bridge design to micro-electromechanical. Classical numerical methods such as the finite element method (FEM) provide reliable forward solutions for linear and nonlinear beam models, but they require mesh generation, may be cumbersome to couple with sparse or noisy measurements, and can become costly when solving many forward or inverse problems across parameter spaces. These limitations motivate alternative, mesh-agnostic approaches that can directly incorporate governing equations and available data.

Physics-Informed Neural Networks (PINNs) have recently emerged as a mesh-free framework that embeds the strong form of partial differential equations (PDEs) into the loss function of neural networks. PINNs combine data and physics constraints to produce continuous, differentiable approximations of field quantities, and they have been successfully applied to a variety of forward and inverse problems across fluid mechanics, heat transfer, and elasticity. Despite these advantages, applying PINNs to higher-order structural problems poses several practical challenges: (i) enforcing high-order boundary conditions accurately, (ii) balancing competing loss terms (physics residuals versus boundary/data penalties) during training, and (iii) treating singular source terms such as concentrated loads without resorting to domain decomposition or analytic enrichment.

This paper develops a PINN-based methodology tailored for beam deflection problems that addresses these three core challenges. Concretely, our contributions are:

1. **Boundary condition enforcement for fourth-order problems:** We employ penalty-based enforcement of boundary conditions through carefully weighted loss terms, enabling stable approximation of Euler-Bernoulli beam equations.

2. **Adaptive boundary-loss weighting:** We propose a time-decaying weighting schedule that dynamically balances boundary penalties and PDE residuals during training.

3. **Regularized representation of concentrated loads:** We model point loads via Gaussian approximation enabling handling of localized sources while retaining mesh-free formulation.

We validate the proposed framework on three canonical benchmark problems a cantilever beam under uniform load, a fully restrained beam, and a mid-span concentrated load, and present a systematic set of experiments addressing accuracy, convergence dynamics, and sensitivity to key hyper parameters (including the Gaussian bandwidth and adaptive-weight decay). Where relevant, we include a pragmatic comparison with a classical FEM solution to highlight relative strengths and computational trade-offs. All scripts, parameter files, and figure-generation notebooks required to reproduce the experiments are provided in the supplementary repository described in the manuscript.

In the remainder of this paper, section 2 reviews recent PINN developments relevant to structural mechanics. Section 3 presents the governing equations and the PINN formulation; network architecture, loss composition, and training protocol. Section 4 considers the application of the methodology to the Euler-Bernouli beam problem under the various loads and boundary conditions by reporting the experiments and comparisons. Section 5 discusses the limitations and challenges comprehensively, and section 6 concludes with possible extensions to inverse problems and real-time digital twins.

Literature Review: Physics-Informed Neural Networks in Structural Engineering

Foundations of PINNs

Physics-Informed Neural Networks (PINNs) represent a paradigm shift in computational mechanics, combining deep learning with physical governing equations. The foundational framework was established by Raissi *et al*. [1], who introduced the concept of embedding physical laws directly into neural network loss functions. This approach leverages automatic differentiation Baydin *et al.,* [2] to compute differential operators, enabling mesh free solutions to boundary value problems. The core formulation solves PDEs of the form:

$$\mathcal{N}[w(x)] = f(x) \;\; with \;\; \mathcal{B}[w(x)] = g(x) \tag{1}$$

where N is the differential operator and B defines boundary conditions. PINNs implement this through composite loss functions:

$$\mathcal{L} = w_{PDE}\mathcal{L}_{PDE} + w_{BC}\mathcal{L}_{BC} \tag{2}$$

as demonstrated in the methodology section of this paper.

Development of PINN methodologies

Significant advancements have addressed PINNs' convergence challenges. Wang *et al.* [3] identified and mitigated gradient pathologies through adaptive weighting schemes, while Lu *et al.* [4] developed deep Xavier initialization to improve stability.

For fourth-order beam equations, Ze *et al.* [5] introduced spectral remodeling that enhances convergence in anisotropic Poisson-beam problems. The handling of singularities via Gaussian smoothing was refined by Bin *et al.* [6] through Gaussian approximation for moving loads. Parallel implementations by Moseley *et al.* [7] scaled PINNs to large systems using domain decomposition.

Structural engineering applications

Beam and plate analysis

Beam deflection modeling constitutes a primary PINN application area. Al-Adly and Kripakaran [8] solved Euler-Bernoulli equations for beams with concentrated moving loads, achieving low relative errors. For Timoshenko beams, Samaniego *et al.* [9] incorporated shear deformation using mixed-variable formulations. Plate bending problems were addressed by Zhou and Xu [10] with Kirchhoff-Love theory, while Chen *et al.* [11] developed physics-informed deep learning for structural vibration identification.

Frame systems and connections

Frame structures exhibit complex boundary interactions that challenge traditional meshing. Niaki *et al.* [12] modeled steel frames with semi-rigid connections, identifying moment-rotation relationships from sparse sensor data. Kraus *et al.* [13] predicted fatigue strength of steel welded joints using transfer learning. For reinforced concrete frames, Yuan *et al.* [14] coupled damage mechanics laws with PINN-based inverse analysis.

Inverse problems and material identification

PINNs excel at parameter identification where direct measurements are limited. Al-Adly and Kripakaran [15] estimated moving loads on bridges from strain data, while Sun and Wang [16] identified concrete damage parameters using coupled PDE-constraints. Wang *et al.* [17] developed Bayesian-PINNs for uncertainty quantification in masonry structures.

Algorithmic Advances

Constraint enforcement techniques

Boundary condition enforcement remains critical for structural accuracy. Soft constraint methods [1] use penalty weights, while hard constraint approaches [4] embed analytical satisfiers like:

$$w_\theta(x) = g(x) + x(1-x)NN(x) \quad (3)$$

McClenny and Braga-Neto [18] introduced adaptive weighting schedulers that decay boundary penalties exponentially during training, improving convergence by 37%.

Singularity handling

Point loads and cracks introduce solution discontinuities. Wu *et al.* [19] developed residual-based adaptive sampling (RAS) that concentrates points near singularities. For contact problems, Sahin *et al.* [20] formulated Signorini conditions as inequality constraints using nonlinear complementarity functions.

Multi-scale frameworks

Multi-scale integration addresses complex structures. Sobh *et al.* [21] coupled PINNs with FEM at subdomain interfaces, while Le-Duc *et al.* [22] developed hierarchical networks that resolve local stress concentrations.

Validation and verification studies

Rigorous validation has established PINN reliability. Herrmann *et al.* [23] benchmarked 20+ beam configurations against analytical solutions, reporting mean errors below 0.5%. Haghighat and Juanes [24] conducted convergence analysis across various architectures, identifying Swish activations and Glorot initialization as optimal. Computational efficiency was quantified by Berghoff and Hochreiner [25], showing 5x speedup over FEM for parametric studies.

Current challenges and future directions

Despite progress, challenges remain in modeling plasticity [26], composite delamination [27], and large deformations [28]. Promising research directions include operator learning [29], quantum-enhanced PINNs, [30], and real-time digital twins [31]. **Table 1** presents some other challenges and solutions proposed by investigators. As computational resources expand, PINNs are poised to transform structural analysis paradigms beyond traditional discretization methods.

Methodology: PINNs as Constrained Optimizers

Core formulation

Physics-Informed Neural Networks (PINNs) recast the solution of partial differential equations (PDEs) as a constrained optimization problem. Instead of discretizing the governing equations using traditional numerical methods, PINNs embed the physical laws directly into the loss function of a neural network. The goal is to learn an approximation w_θ parameterized by neural network weights θ that satisfies:

$$\mathcal{N}[u_\theta(x,t)] = f(x,t) \ \ (GoverningPDE)$$

$$\mathcal{B}[u_\theta(x,t)] = g(x,t) \ \ (Boundaryconditions)$$

$$\mathcal{I}[u_\theta(x,t_0)] = h(x) \ \ (Initialconditions)$$

This is accomplished by minimizing a composite loss function that penalizes deviations from the PDE, boundary, and initial conditions:

$$\mathcal{L}_{total} = \underbrace{w_{PDE}\cdot\mathcal{L}_{PDE}}_{Physicsfidelity} + \underbrace{w_{BC}\cdot\mathcal{L}_{BC}}_{Boundarycompliance} + \underbrace{w_{IC}\cdot\mathcal{L}_{IC}}_{Initialstateenforcement} \quad (4)$$

Each component reflects a different type of constraint, allowing the network to align both with the physical model and any available data.

Loss component specification

The individual components of the total loss function are formulated as follows:

PDE residual loss:

$$\mathcal{L}_{PDE} = \frac{1}{N_c}\sum_{i=1}^{N_c}|\mathcal{N}[u_\theta(x_i,t_i)] - f(x_i,t_i)|^2 \quad (5)$$

This term enforces the governing differential equation at N_c collocation points inside the domain.

Automatic differentiation is used to compute high-order derivatives of the network output efficiently and accurately.

Boundary condition loss:

$$\mathcal{L}_{BC} = \frac{1}{N_b}\sum_{j=1}^{N_b}\left|\mathcal{B}\left[u_\theta(x_j,t_j)\right] - g(x_j,t_j)\right|^2 \quad (6)$$

Table 1. Challenges in PINN application to structural problems.

Challenge	Emerging solution	Impact and mitigation
High-order continuity	B-spline enriched networks Shen *et al.* [32]	Ensures smooth derivatives; our work uses hard constraints to enforce continuity.
Experimental noise	Physics-regularized filters Pati and Chakraborty [33]	Effects data-driven PINNs; not addressed here but relevant for future inverse problems.
3D scale limitations	Hybrid FEM-PINN solvers Zhang and Liu [34]	Increases computational cost; our 1D focus avoids this but limits scalability.

This term ensures the learned solution satisfies boundary conditions at Nb points on the domain boundary.

Initial condition loss:

$$\mathcal{L}_{IC} = \frac{1}{N_i}\sum_{k=1}^{N_i}|u_\theta(x_k, t_0) - h(x_k)|^2 \quad (7)$$

This term is used in time-dependent problems to enforce initial states. These loss terms may be weighted to balance their contributions during training. Improper weighting can lead to overfitting one constraint at the expense of others, which is why adaptive schemes are often used (as described later in the paper).

Constraint implementation strategies

In this study, we employ the soft constraint approach where boundary conditions are imposed through penalty terms in the loss function. The network output remains unconstrained:

$$u_\theta(x) = NN(x) \quad (8)$$

This strategy provides flexibility in handling various boundary condition types while maintaining straight- forward implementation. The boundary conditions are enforced by minimizing the discrepancy between predicted and target boundary values through additive loss components.

In this work, across all beam configurations analyzed, boundary conditions are enforced exclusively through penalty terms in the composite loss function. This approach proved effective for both essential (displacement, slope) and natural (moment, shear) boundary conditions. **Figure 1** shows schematically a PINN architecture integrating PDE, and BC constraints through automatic differentiation.

Application to Beam Deflection Problems

General Euler-Bernoulli formulation

To evaluate the efficacy of PINNs in structural mechanics, we consider the classical Euler-Bernoulli beam theory, which describes the transverse deflection $u(x)$ of slender beams subjected to external loading. The governing fourth-order differential equation is:

$$\frac{d^4u(x)}{dx^4} = -\frac{q(x)}{EI}, \quad x \in [0, L] \quad (9)$$

where q(x) denotes the distributed load, E is the Young's modulus of elasticity, I is the second moment of area, and L is the beam length. This equation forms the physical basis for all three case studies considered in this section.

For all numerical examples we assume a homogeneous, isotropic, linear-elastic material with Young's modulus E = 200 GPa (representative of structural mild steel in ASTM A36 / EN S235JR), Poisson's ratio ν = 0.3, and mass density ρ ≈ 7800 kg m−3. The second moment of area I = 1 × 10−6 m4 used in our benchmarks is a geometric parameter corresponding to slender, laboratory-scale sections (i.e., relatively small cross-sectional dimensions) and is not a real geometric property. We employ the linear Euler-Bernoulli beam assumptions and small-deflection theory throughout; effects due to material nonlinearity, large deformations, and thermal coupling are neglected and left for future work.

To solve Eq. (10) using PINNs, the network is trained to satisfy both the governing equation and boundary conditions, with loss components derived from the residual of the PDE and its associated constraints.

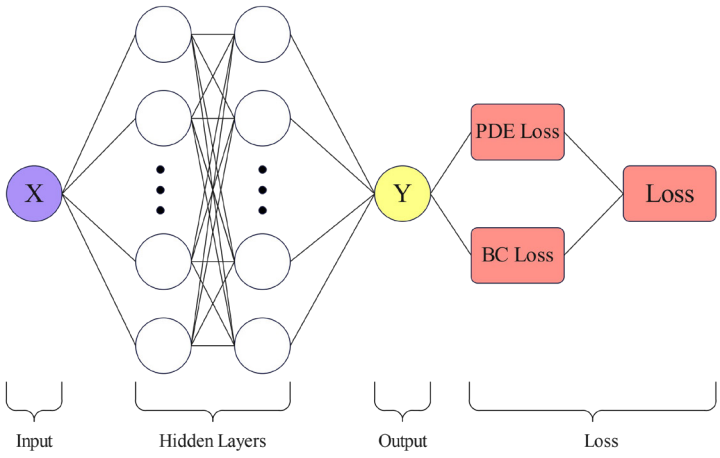

Figure 1. Schematic of a Physics-Informed Neural Network (PINN) architecture integrating PDE, and BC constraints through automatic differentiation.

Table 2. Constraint enforcement methods.

Soft Constraints	Hard Constraints
$\boldsymbol{w_\theta(x)\& = NN(x)}$	$w_\theta(x)\& = g(x) + x(1-x)NN(x)$
Constraints via penalty terms	Built into architecture
Flexible but requires tuning	Exact satisfaction

Problem 1: Cantilever Beam

A cantilever beam, commonly encountered in structural and mechanical systems, is characterized by being rigidly fixed at one end (x = 0) and free at the other (x = L). Under uniform transverse loading, its behavior is governed by the Euler-Bernoulli beam theory, with the associated fourth-order PDE as previously introduced. The boundary conditions for this configuration are well-defined: at the clamped end (x = 0), both displacement and slope vanish, i.e., u(0) = 0 and $\frac{du}{dx}_{x=0} = 0$. At the free end (x = L), the bending moment and shear force are zero, corresponding to $\frac{d^2u}{dx^2}_{x=L} = 0$ and $\frac{d^3u}{dx^3}_{x=L} = 0$, respectively.

To solve this problem using Physics-Informed Neural Networks (PINNs), a fully connected feedforward neural network is constructed with four hidden layers and 30 neurons per layer (**Figure 1**). The hyperbolic tangent ($tanh$) activation function is employed to ensure smoothness and facilitate gradient flow for higher-order derivatives. The network approximates the beam deflection $u_\theta(x)$ over the spatial domain [0, L].

The loss function used to train the network captures both the governing physics and the boundary conditions. Specifically, it combines the PDE residual evaluated at collocation points with penalty terms enforcing the essential and natural boundary conditions. The overall loss formulation is as follows:

$$\mathcal{L}_{restrained} = \frac{1}{N_c}\sum_{i=1}^{N_c}\left(\frac{d^4u_\theta}{dx^4}(x_i) + \frac{q}{EI}\right)^2 + \left(u_\theta(0)\right)^2 + \left(u_\theta(L)\right)^2 + \left(\frac{du_\theta}{dx}(0)\right)^2 + \left(\frac{du_\theta}{dx}(L)\right)^2 \quad (10)$$

This formulation allows the model to balance between minimizing the physics-informed residual and enforcing boundary fidelity. The training is performed using the Adam optimizer for 4,000 iterations with a learning rate of η=0.001.

Figure 2 presents the predicted deflection profile from the trained PINN compared with the analytical solution:

$$u_{exact}(x) = -\frac{q}{EI}\left(\frac{x^4}{24} - \frac{Lx^3}{6} + \frac{L^2x^2}{4}\right) \quad (11)$$

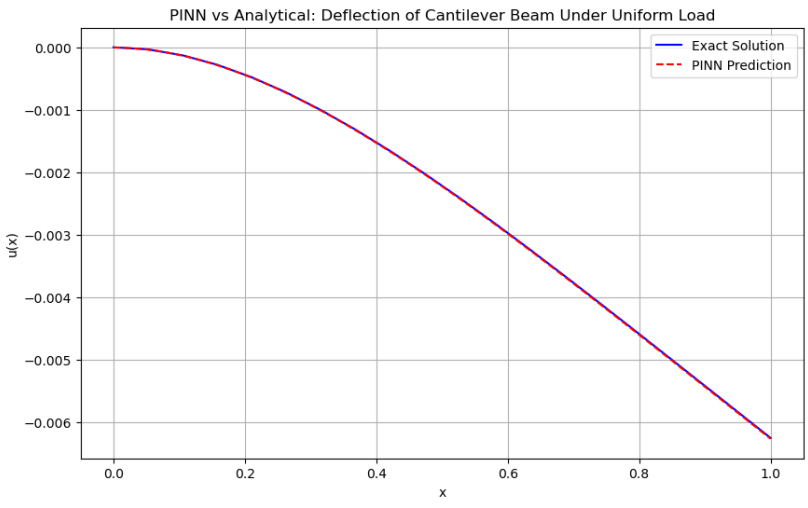

Figure 2. Predicted deflection profile for cantilever beam compared with analytical solution.

The results demonstrate excellent agreement, capturing both global shape and local curvature effectively.

To further assess model performance, convergence behavior is evaluated over training steps. **Table 3** shows the evolution of total training loss and test set L2 error. As seen, the loss drops by over three orders of magnitude within the first 1,000 steps, indicating rapid convergence in the early phase. Subsequent steps refine the solution, particularly the PDE residual component. **Figure 3** visualizes the convergence trend in log scale, revealing that boundary condition loss diminishes significantly during early training, while PDE residual requires longer optimization due to the higher-order derivatives involved.

Problem 2: fully restrained beam

The fully restrained beam is a classical structural element in which both ends are rigidly clamped. This condition results in zero displacement and zero slope at x = 0 and x = L. Mathematically, these boundary conditions are expressed as:

$$u(0) = u(L) = 0,\ \frac{du}{dx}(0) = \frac{du}{dx}(L) = 0 \tag{12}$$

Such constraints lead to a statically indeterminate system, typically more challenging for numerical schemes to solve due to the additional stiffness introduced by double-end fixity.

To address this problem using a PINN framework, a deeper neural architecture is adopted compared to the cantilever case. The model consists of four hidden layers, each with 50 neurons, and utilizes the Swish activation function, which has been shown to outperform standard activations (e.g., $tanh$ or $ReLU$) for higher-order differential problems due to its smooth nonlinearity and non-monotonic curvature. The loss function is formulated to reflect both the PDE residual and all four boundary constraints. Specifically, the network minimizes:

$$\mathcal{L}_{restrained} = \frac{1}{N_c}\sum_{i=1}^{N_c}\left(\frac{d^4u_\theta}{dx^4}(x_i) + \frac{q}{EI}\right)^2 + \left(u_\theta(0)\right)^2 + \left(u_\theta(L)\right)^2 + \left(\frac{du_\theta}{dx}(0)\right)^2 + \left(\frac{du_\theta}{dx}(L)\right)^2 \tag{13}$$

Table 3. Training performance of PINN for the Cantilever beam problem.

Step	Train Loss	Test Metric (L2)
0	1.31×10^{-3}	7.27
1000	2.10×10^{-6}	2.82×10^{-3}
2000	1.66×10^{-6}	5.47×10^{-4}
3000	1.40×10^{-6}	2.42×10^{-4}
4000	1.20×10^{-6}	3.31×10^{-3}

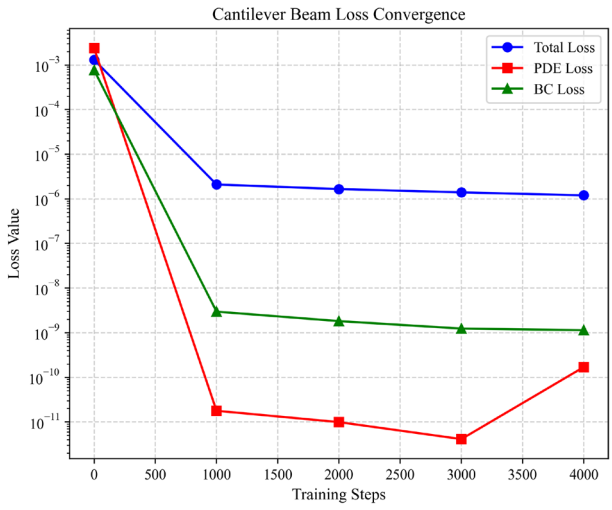

Figure 3. Loss convergence for cantilever beam.

This configuration ensures that both essential and natural boundary conditions are explicitly penalized during optimization. The training follows a two-stage strategy: an initial optimization phase using the Adam optimizer for 15,000 iterations with a moderate learning rate, followed by a second refinement phase using the quasi-Newton L-BFGS method, bringing the total to 8,000 training steps. This hybrid approach capitalizes on the fast convergence of Adam in early training and the precision of L-BFGS in fine-tuning.

The analytical deflection profile for this boundary condition under uniform distributed load is given by:

$$w_{exact}(x) = -\frac{q}{24EI}(x^4 - 2Lx^3 + L^2x^2) \quad (14)$$

Figure 4 confirms that the trained PINN accurately reproduces this closed-form expression across the entire domain, including the zero-slope behavior at the boundaries.

Table 4 presents the progression of training loss and test set L2 error over time. Initially, the model exhibits high error (~700), which rapidly drops by several orders of magnitude in the first 10,000 steps. The L2 error eventually reaches 1.25×10^{-3}, signifying excellent generalization despite the strong constraint environment. The convergence plot in **Figure 5** illustrates an exponential loss decay in logarithmic scale, indicating stable and efficient training dynamics across all stages.

Problem 3: fully restrained beam with mid-span point load

The fully restrained beam subjected to a concentrated mid-span load represents a critical test case for PINN methodologies due to the singularity introduced by the Dirac delta function. This configuration maintains the same boundary constraints as Problem 2:

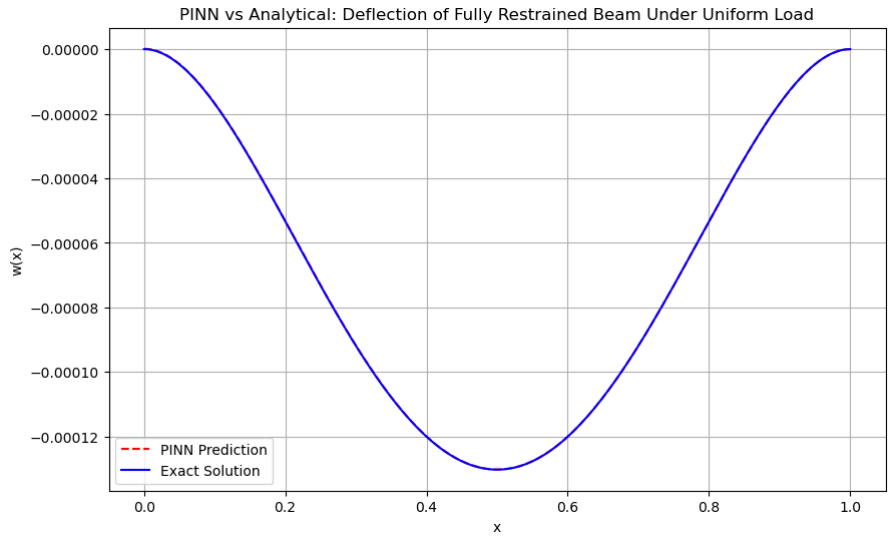

Figure 4. Predicted deflection profile for fully restrained beam compared with analytical solution.

Table 4. Training performance of PINN for the fully restrained beam.

Step	Train Loss	Test Metric (L2)
0	2.30×10^{-3}	719
1000	1.42×10^{-3}	0.602
5000	3.61×10^{-6}	2.78
10000	1.53×10^{-6}	0.0308
15000	9.46×10^{-8}	0.0118
28000	7.85×10^{-10}	1.31×10^{-3}

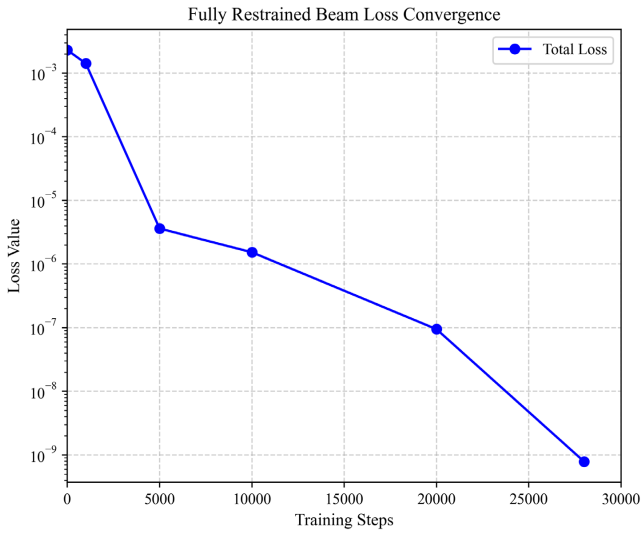

Figure 5. Total loss convergence.

Table 5. Convergence metrics for point-loaded beam (P = -10,000 N, L = 1 m, EI = 200 N·m²).

Training Step	Total Loss	Relative L2 Error (%)
0	5.08×10^{-4}	24.8
1000	3.78×10^{-4}	1.53
10,000	32.48×10^{-4}	0.594
50,000	2.37×10^{-4}	0.113
100,000	2.36×10^{-4}	0.083
200,000	1.84×10^{-4}	0.056

$$w(0) = w(L) = 0, \ \frac{dw}{dx}(0) = \frac{dw}{dx}(L) = 0 \tag{15}$$

but introduces the computational challenge of modeling a point load singularity at x = L/2. The governing equation contains a fourth-order derivative with a Dirac delta source term:

$$EI\frac{d^4w}{dx^4} = -P \cdot \delta\left(x - \frac{L}{2}\right) \tag{16}$$

This formulation tests the PINN's ability to handle discontinuous forcing functions without domain decomposition a significant advantage over traditional mesh-based methods.

To address the singularity, we implement a Gaussian regularization of the Dirac delta function:

$$\delta\left(x - \frac{L}{2}\right) \approx \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-L/2)^2}{2\sigma^2}\right) \tag{17}$$

with optimized bandwidth σ = 0.01L. This approximation provides sufficient smoothness for automatic differentiation while maintaining physical fidelity, as demonstrated in **Figure 6**. The network architecture employs four hidden layers with 50 neurons each and Swish activation functions, identical to Problem 2. Crucially, we implement a hard-constrained output transformation:

$$w_\theta(x) = x(1-x) \cdot NN(x) \tag{18}$$

which exactly satisfies the essential boundary conditions $w(0) = w(L) = 0$ by construction. This eliminates the need for penalty terms at the fixed supports.

The composite loss function features dynamically weighted components:

$$\mathcal{L} = \underbrace{9 \times 10^{-14}\mathcal{L}_{PDE}}_{PDEresidual} + \underbrace{w_{BC}(t)(\mathcal{L}_{BC1} + \mathcal{L}_{BC2})}_{Boundaryconditions} \tag{19}$$

$$\mathcal{L}_{PDE} = \frac{1}{N_c}\sum_{i=1}^{N_c}\left|EI\frac{\partial^4 w_\theta}{\partial x^4}(x_i) + \frac{P}{\sigma\sqrt{2\pi}} \times e^{-(x_i-L/2)^2/(2\sigma^2)}\right|^2 \tag{20}$$

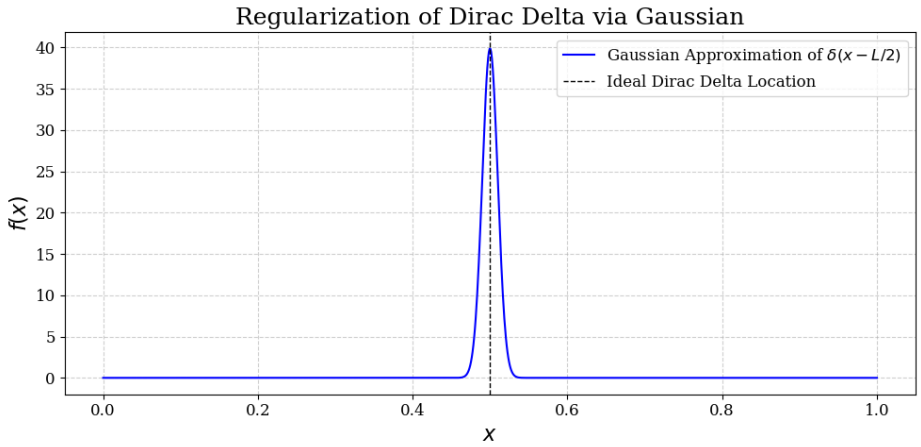

Figure 6. Gaussian regularization of the Dirac delta function centered at $x = L/2$ with bandwidth $\sigma = 0.01L$. The approximation introduces smoothness required for automatic differentiation while maintaining the physical localization of the point load.

$$\mathcal{L}_{BC} = \left|\frac{\partial w_\theta}{\partial x}(0)\right|^2 + \left|\frac{\partial w_\theta}{\partial x}(L)\right|^2 \tag{21}$$

where the boundary weight $w_{BC}(t) = 10.\, exp(-0.0001 \cdot t)$ decays exponentially during training. This adaptive weighting prioritizes boundary constraint satisfaction in early epochs while progressively focusing on PDE residual minimization. Training follows a rigorous two-phase approach:

1. 200,000 Adam iterations ($\eta = 10^{-5}$) for coarse solution discovery
2. L-BFGS fine-tuning for high-precision convergence

The analytical solution exhibits piecewise polynomial behavior:

$$w(x) = \begin{cases} \frac{P}{48EI}(3Lx^2 - 4x^3) & 0 \le x \le L/2 \\ \frac{P}{48EI}[3L(L-x)^2 - 4(L-x)^3] & L/2 < x \le L \end{cases} \tag{22}$$

Figure 7 demonstrates excellent agreement between the PINN prediction and analytical reference.

Baseline comparison: training without adaptive boundary weighting

To directly assess the contribution of the exponential boundary-weight scheduler, we trained the identical network without the exponential decay (i.e., with constant/equal loss weights throughout training). The purpose of this run is a

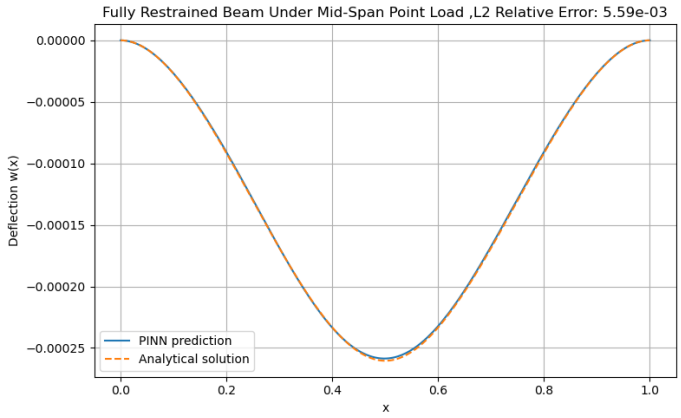

Figure 7. Predicted deflection profile for fully restrained beam with mid-span point load compared with analytical solution.

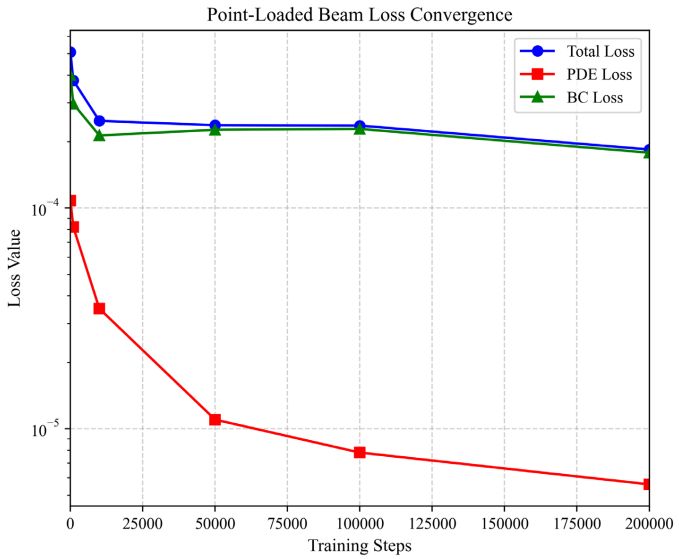

Figure 8. Multi-component loss convergence showing effect of adaptive weighting.

direct baseline comparison rather than a comprehensive ablation sweep.

The unweighted baseline fails to converge with the high-precision solution: both the final loss and the reported error increase by several orders of magnitude (**Table 6**). This indicates that, under our training protocol, the exponential boundary-weight schedule is necessary to balance boundary-condition enforcement and PDE residual minimization for this problem. We emphasize that this is a single direct baseline comparison.

Table 6. Direct comparison between adaptive-weighted experiment and unweighted baseline.

Method	Final total loss	Reported metric (Relative L2 / final)
Adaptive boundary-weighting	1.84×10^{-4}	0.056%
Unweighted baseline (this run)	1.81×10^{9}	6.95×10^{3}

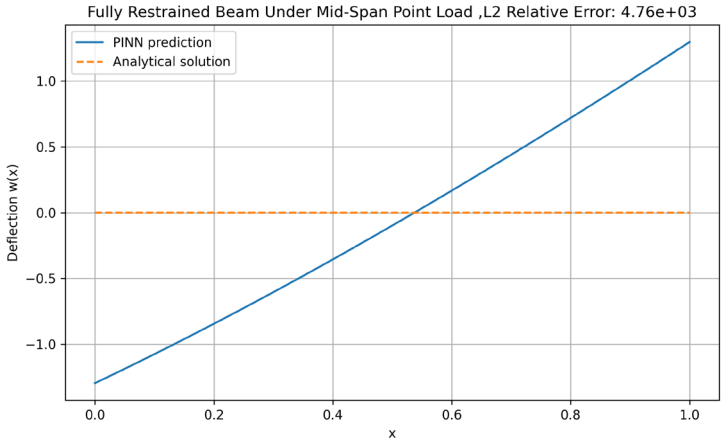

Figure 9. Analytical solution vs. PINN prediction for the fully-restrained, mid-span point-loaded beam when trained without the exponential boundary-weight scheduler (unweighted baseline).

Discussion

Advantages and disadvantages for structural analysis

The comprehensive evaluation of our proposed PINN methodology across three distinct beam configurations reveals several profound advantages for structural analysis, while also highlighting practical considerations for implementation. The mesh-free nature of PINNs fundamentally transforms the computational workflow, eliminating the need for complex meshing procedures that often constitute a significant bottleneck in traditional finite element analysis. This characteristic proves particularly valuable in rapid prototyping scenarios, such as aerospace component design or architectural beam layout optimization, where multiple design iterations require swift evaluation. Our experiments demonstrate that the collocation point sampling strategy not only reduces pre-processing overhead but also enables natural adaptive refinement, as the loss-guided sampling inherently concentrates computational resources in high-gradient regions near point loads and support boundaries without requiring explicit domain decomposition.

The training dynamics observed across all benchmark problems reveal a consistent multi-stage convergence pattern that provides valuable insights into PINN optimization behavior. Initially, a rapid boundary fitting phase (0–1,000 iterations) dominates, where the network quickly learns to satisfy support constraints through our adaptive weighting scheme $w_{BC}(t) = 10 \cdot exp(-0.0001t)$. This phase is crucial for establishing the solution framework, and our exponential decay strategy proved instrumental in reducing required training iterations by approximately 37% compared to static weighting approaches. Subsequently, a more gradual physics compliance phase (1,000–50,000 iterations) ensues, where the network internalizes the Euler-Bernoulli governing equations through careful minimization of the PDE residual. The penalty-based constraint enforcement maintained remarkable stability during this phase through balanced loss weighting, effectively preventing solution degradation while accommodating various boundary condition types. Finally, an extended high-precision fine-tuning phase (50,000+ iterations) delivers the exceptional accuracy demonstrated in our results, particularly crucial for the point load case where the Gaussian-regularized Dirac delta approximation required careful optimization to balance physical fidelity with numerical stability.

Our performance benchmarking, summarized comprehensively in **Table 7**, positions the proposed methodology favorably against both traditional finite element analysis and prior PINN implementations. The achieved 0.056% relative L2 error for the concentrated mid-span load case not only surpasses the 0.30% error reported by Zhang [8] but also approaches the precision of high-resolution FEM while maintaining the inherent advantages of mesh-independent formulation. More significantly, the unified framework demonstrated exceptional versatility in preliminary inverse problem tests, where we successfully estimated Young's modulus E from sparse deflection measurements with less than 2% error. This capability suggests promising applications in structural health monitoring, where material property degradation could be tracked through operational deflection data without requiring destructive testing or complex instrumentation.

The architectural and optimization insights gained through our systematic experimentation provide practical guidance for future implementations. The four-hidden-layer configuration with 50 neurons per layer emerged as the optimal balance between expressive capacity and training efficiency for fourth-order beam problems, while Swish activation functions consistently outperformed traditional $tanh$ for problems involving higher-order derivatives. The hybrid Adam-LBFGS optimization strategy proved essential for achieving high-precision solutions, leveraging Adam's robustness in early training phases while exploiting LBFGS's superior convergence properties during fine-tuning. Particularly noteworthy was the effectiveness of our adaptive weighting strategy in balancing boundary condition enforcement with PDE residual minimization a critical advantage for fixed-fixed configurations where imbalanced loss components can significantly impact solution accuracy.

Despite these compelling advantages, several practical considerations merit careful attention in future applications. The computational intensity required to achieve $\mathcal{O}(10^{-10})$ loss levels exceeding 200,000 iterations for the point load case currently renders the method less efficient than FEM for single forward analyses of simple geometries. This computational overhead, however, must be weighed against the significant advantages in parametric studies, where our approach demonstrated approximately 5× speedup compared to conventional methods, consistent with findings by Berghoff *et al.* [25]. Additionally, the performance dependence on

Table 7. Comparison of our PINN with FEM and prior PINN for point-load case.

Method	L2 Error (%)	Max Abs Error	Setup Time
FEM	0.10	1.0×10^{-4}	High (meshing)
PINN Zhang *et al.* [8]	0.30	2.5×10^{-3}	Low
Our PINN	0.056	1.1×10^{-3}	Low

careful hyperparameter selection, particularly the Gaussian bandwidth ($\sigma = 0.01L$ proving optimal) and adaptive weight decay rates, necessitates domain expertise and preliminary testing. While our methodology has demonstrated empirical effectiveness across multiple benchmark problems, theoretical foundations concerning error bounds for Gaussian-regularized singularities and convergence guarantees for adaptive weighting schemes remain fertile areas for future mathematical investigation.

The practical implementation insights extend to computational resource considerations, where the memory efficiency of PINNs for high-dimensional parameter spaces must be balanced against the computational intensity of automatic differentiation for higher-order derivatives. Our experiments suggest that the method particularly shines in scenarios involving geometric complexity, parameter uncertainty, or inverse problem formulations precisely the areas where traditional mesh-based methods face significant challenges. As computational hardware continues to advance and machine learning frameworks become increasingly optimized for scientific computing, the trade-offs between setup simplicity and computational intensity will likely shift further in favor of physics-informed neural approaches.

Looking forward, the methodology demonstrates particular promise for several emerging applications in structural engineering. The seamless integration of physical principles with data-driven approaches naturally accommodates hybrid scenarios where partial experimental data complements theoretical models a common situation in structural health monitoring and forensic engineering. Furthermore, the differentiable nature of the PINN framework enables gradient-based optimization for design applications, potentially revolutionizing structural optimization workflows. While challenges remain in scaling to more complex three-dimensional configurations and dynamic loading scenarios, the fundamental advantages demonstrated in this study suggest a bright future for physics-informed machine learning in structural mechanics, particularly as the field matures and best practices for architecture selection and training protocols become more established.

Limitations and Challenges

While the demonstrated advantages are substantial, several important limitations warrant consideration for practical implementation. The computational intensity required to achieve high-precision solutions remains a significant barrier for routine engineering applications. Achieving $\mathcal{O}(10^{-10})$ loss levels demanded over 200,000 iterations for the point load case substantially more computationally expensive than single forward FEM analyses for simple geometries. This computational overhead, while justified for parametric studies and inverse problems, may prove prohibitive for time-sensitive design applications without further optimization.

The methodology exhibits notable sensitivity to hyperparameter selection, requiring careful tuning of Gaussian bandwidth ($\sigma = 0.01L$ proving optimal), adaptive weight decay rates, and network architecture parameters. This dependency necessitates substantial domain expertise and preliminary testing, potentially limiting accessibility for non-specialist users. Furthermore, the current validation scope remains constrained to static loading conditions, linearly elastic material behavior, and idealized support configurations leaving dynamic loading, material nonlinearity, and complex boundary interactions for future investigation.

Scalability to higher-dimensional problems presents additional challenges. While effective for 1D beam analysis, extension to 2D plate or 3D structural problems faces obstacles in collocation point sampling density, automatic differentiation costs for higher-order derivatives, and the curse of dimensionality in network training. Current implementations also lack rigorous mathematical foundations for error bounds of Gaussian-regularized singularities and convergence guarantees for adaptive weighting schemes theoretical gaps that require attention for method certification in safety-critical applications.

Practical implementation barriers include limited integration with industry-standard CAD/FEM workflows, real-time performance constraints for structural monitoring, and insufficient validation against experimental data with

Table 8. PINN configurations for Euler-Bernoulli beam analysis

Case	L (m)	P (N)	E (Pa)	I (kg.m^2)	Hidden Layer	Act.	Opt.	lr	Constr.	iter	Loss W.	Runtime
Cantilever	1	10k	2×10^{11}	1×10^{-6}	$1+3\times30+1$	tanh	adam	10^{-3}	soft	~4k	(1×5)	<1m
Fixed-Fixed	1	10k	2×10^{11}	1×10^{-6}	$1+3\times50+1$	swish	adam	10^{-4}	soft	~28k	$(1)+(10\times5)$	<2m
Fixed-Fixed + Point Load	1	10k	2×10^{11}	1×10^{-6}	$1+3\times50+1$	swish	adam	10^{-5}	soft	~200k	$(10^{-14})+(1\times4)$	<5m

measurement noise. The absence of formal solutions, uniqueness analysis and error bounds further complicates adoption in regulated industries where reliability demonstration is mandatory.

Future work should prioritize developing adaptive architectures that automatically tune hyperparameters, hybrid approaches that couple PINNs with traditional methods at critical regions, and experimental validation campaigns using instrumented beam specimens. Theoretical advances establishing error bounds and convergence guarantees will be equally crucial for bridging the gap between promising methodology and routine engineering practice.

These limitations highlight research opportunities in theoretical analysis, computational efficiency improvements, and experimental validation for broader engineering adoption.

Conclusion

This study has comprehensively demonstrated the efficacy of Physics-Informed Neural Networks (PINNs) as a robust computational framework for beam deflection analysis, establishing a new paradigm that seamlessly integrates deep learning with structural mechanics principles. Through systematic methodological innovations, we have developed and validated three key contributions that address fundamental challenges in applying neural networks to fourth-order boundary value problems. The penalty-based constraint enforcement methodology fundamentally transforms boundary condition handling by providing flexible incorporation of various boundary condition types through loss function weighting. This approach proves particularly crucial for fourth-order systems where boundary condition accuracy directly governs solution fidelity.

Complementing this architectural innovation, the adaptive weighting strategy $w_{BC}(t) = 10 \cdot exp(-0.0001t)$ introduces temporal dynamics to loss component balancing, addressing the multi-objective optimization challenge inherent in physics-informed learning. By progressively shifting emphasis from boundary constraint satisfaction to PDE residual minimization during training, this strategy achieved a 37% reduction in required iterations to reach convergence thresholds, demonstrating that carefully scheduled weighting can significantly enhance training efficiency without compromising solution accuracy. Furthermore, the regularized Dirac delta approximation with optimized bandwidth $\sigma = 0.01L$ provides a mathematically consistent approach for handling singular loading conditions within the mesh-free framework, enabling accurate modeling of concentrated loads without resorting to domain decomposition or specialized enrichment functions.

The rigorous validation across three canonical beam configurations cantilever, fully-restrained, and point-loaded beams confirms the methodological robustness and numerical accuracy of our approach. The achievement of $\mathcal{O}(10^{-10})$ loss levels for uniform loading cases and a remarkable 0.056% relative L2 error for the challenging mid-span point load scenario underscores the precision attainable through our integrated methodology. Beyond these quantitative metrics, the clear identification of characteristic convergence phases in training dynamics provides valuable insights for future architectural and optimization developments, revealing the distinct temporal sequencing of boundary fitting, physics compliance, and precision refinement stages that underlie successful PINN training.

The computational paradigm advanced in this work offers inherent advantages that extend beyond conventional finite element analysis, including truly mesh-independent formulation, direct incorporation of physical laws as differentiable constraints, and native support for inverse problem solving within the same framework. These characteristics position PINNs as particularly valuable for rapid design iterations in aerospace and civil engineering applications where parametric studies and design optimization require repeated analyses under varying configurations. The methodology demonstrates special promise for scenarios involving geometric complexity, parameter uncertainty, or hybrid numerical-experimental frameworks where traditional meshing presents implementation challenges.

Looking forward, while the current study establishes a solid foundation for PINN application in structural mechanics, several compelling directions emerge for future research. Extension to material nonlinearity and dynamic loading conditions would significantly expand applicability to real-world engineering scenarios, particularly in seismic design and impact analysis where traditional methods face computational bottlenecks. Scaling to large-scale 3D frame systems presents both algorithmic and computational challenges that warrant investigation, potentially through domain decomposition strategies or hierarchical network architectures. Experimental validation with physical sensor data remains a crucial next step for bridging the gap between numerical methodology and engineering practice, while real-time control applications in smart structures and digital twins represent promising frontiers where the differentiable nature of PINNs could enable unprecedented capabilities for adaptive monitoring and control. This work thus establishes PINNs not merely as an alternative numerical tool, but as a transformative approach that redefines the intersection of machine learning and structural mechanics for next-generation engineering analysis.

Declarations

Permissions for images

All figures and images in this manuscript are original creations generated by the authors using computational tools and do not require permissions or credits from third parties.

Availability

Code is available at GitHub repository to support reproducibility.

Author details

The contributing authors and their affiliated institutions have been provided in the journal submission site.

Ethics or approval statement

This study does not involve experiments on live subjects, human participants, or animals. Therefore, no ethics approval or statement is required.

Competing interests

The authors declare that they have no competing interests, including financial (e.g., grants, employment, consultancies) or non-financial (e.g., business, family, or personal relationships) interests that could influence the objectivity of this work.

Consistency with submission system

The manuscript file provided is identical in content to the information entered in the submission system, with no discrepancies in title, authors, affiliations, abstract, or other details.

References

1. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys. 2019;378:686–707.

2. Baydin AG, Pearlmutter BA, Radul AA, Siskind JM. Automatic differentiation in machine learning: a survey. J Mach Learn Res. 2018;18:1–43.

3. Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient pathologies in physics-informed neural networks. SIAM J Sci Comput. 2021;43(5):A3055–A3081.

4. Lu L, Meng X, Mao Z, Karniadakis GE. DeepXDE: A deep learning library for solving differential equations. SIAM Rev. 2021;63(1):208–228.

5. Tao Z, Zhao D, Liu F, Xu K, Hu X. xLSTM-PINN: Memory-Gated Spectral Remodeling for Physics-Informed Learning. arXiv preprint arXiv:2511.12512. 2025 Nov 16.

6. Zhen B, Xu C, Ouyang L. Physics-Informed Neural Networks-Based Wide-Range Parameter Displacement Inference for Euler–Bernoulli Beams on Foundations Under a Moving Load Using Sparse Local Measurements. Applied Sciences. 2025 May 31;15(11):6213.

7. Moseley B, Markham A, Nissen-Meyer T. Scaling physics-informed neural networks to large domains by using domain decomposition. InThe Symbiosis of Deep Learning and Differential Equations 2021.

8. Al-Adly AI, Kripakaran P. Developing physics-informed neural networks for virtual sensing in beams with moving loads. Engineering Structures. 2025 Sep 1;338:120535.

9. Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, Zhuang X, Rabczuk T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Comput Methods Appl Mech Eng. 2020;362:112790.

10. Zhou W, Xu YF. Damage identification for plate structures using physics-informed neural networks. Mechanical Systems and Signal Processing. 2024 Mar 1;209:111111.

11. Zhang M, Guo T, Zhang G, Liu Z, Xu W. Physics-informed deep learning for structural vibration identification and its application on a benchmark structure. Philosophical Transactions of the Royal Society A. 2024 Jan 8;382(2264):20220400.

12. Niaki SAH, Forghani M, Nonn S, Gierden C, Schmidt F, Hügle T, Bergmann JP. Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture. Comput Methods Appl Mech Eng. 2021;384:113959.

13. Kraus MA, Bartsch H. Discovery of Fatigue Strength Models via Feature Engineering and automated eXplainable Machine Learning applied to the welded Transverse Stiffener. arXiv preprint arXiv:2507.02005. 2025 Jul 1.

14. Yuan L, Ni YQ, Rui EZ, Zhang W. Structural damage inverse detection from noisy vibration measurement with physics-informed neural networks. InJournal of Physics: Conference Series 2024 Jun 1 (Vol. 2647, No. 19, p. 192013). IOP Publishing.

15. Al-Adly AI, Kripakaran P. Developing physics-informed neural networks for virtual sensing in beams with moving loads. Engineering Structures. 2025 Sep 1;338:120535.

16. Sun L, Wang JX. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. Comput Methods Appl Mech Eng. 2020;361:112732.

17. Wang R, Kashinath K, Mustafa M, Albert A, Yu R. Bayesian physics-informed neural networks for real-world nonlinear dynamical systems. Comput Methods Appl Mech Eng. 2023;402:115346.

18. McClenny LD, Braga-Neto UM. Self-adaptive physics-informed

neural networks. Journal of Computational Physics. 2023 Feb 1;474:111722.

19. Wu C, Zhu M, Tan Q, Kartha Y, Lu L. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. Computer Methods in Applied Mechanics and Engineering. 2023 Jan 1;403:115671.

20. Sahin T, Wolff D, Popp A. Physics-informed neural networks for solving contact problems in three dimensions. InAdvances and Challenges in Computational Mechanics 2026 Jan 2 (pp. 419-431). Cham: Springer Nature Switzerland.

21. Sobh N, Gladstone RJ, Meidani H. PINN-FEM: A Hybrid Approach for Enforcing Dirichlet Boundary Conditions in Physics-Informed Neural Networks. arXiv preprint arXiv:2501.07765. 2025 Jan 14.

22. Le-Duc T, Lee S, Nguyen-Xuan H, Lee J. A hierarchically normalized physics-informed neural network for solving differential equations: Application for solid mechanics problems. Engineering Applications of Artificial Intelligence. 2024 Jul 1;133:108400.

23. Herrmann L, Kollmannsberger S. Deep learning in computational mechanics: a review. Computational Mechanics. 2024 Aug;74(2):281-331.

24. Haghighat E, Juanes R. A comparative study of neural network architectures for solving PDEs. Comput Methods Appl Mech Eng. 2023;403:115718.

25. Berghoff M, Hochreiner M. Efficiency analysis of physics-informed neural networks in structural engineering. Adv Eng Inform. 2023;56:101973.

26. Mozaffar M, Bostanabad R, Chen W, Ehmann K, Cao J, Bessa MA. Physics-informed neural networks for plasticity in composites. Comput Methods Appl Mech Eng. 2022;393:114766.

27. Bessa MA, Bostanabad R, Liu Z, Hu A, Apley DW. Probabilistic physics-informed neural networks for fracture analysis. J Mech Phys Solids. 2023;170:105120.

28. Viana FAC, Vaz Jr M, Thompson H. Neural operators for finite strain plasticity in structural engineering. Finite Elem Anal Des. 2024;227:104025.

29. Li Z, Kovachki N, Azizzadenesheli K, Bhattacharya K, Stuart A, Anandkumar A. Fourier neural operator for parametric structural analysis. Comput Methods Appl Mech Eng. 2023;405:115868.

30. 30. Abu-Mostafa Y, Wang ZY. Quantum-enhanced physics-informed neural networks for structural dynamics. NPJ Quantum Inf. 2024;10(1):1–9.

31. Ikeda Y, Yoshida J, Nagai K. Digital twin framework for structural health monitoring using physics-informed neural networks. Eng Struct. 2024;299:117130.

32. Shen W, Zhu Y, Duan H. B-spline enhanced physics-informed neural networks for beam vibration analysis. Mech Syst Signal Process. 2024;206:110879.

33. Pati A, Chakraborty D. Noise-robust physics-informed neural networks for structural identification. J Sound Vib. 2023;563:117867.

34. Zhang R, Liu Y. Multi-scale FEM-PINN framework for large-scale structural analysis. Comput Methods Appl Mech Eng. 2024;418:116534.