Authors' Response to Reviews of

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

K. Baharan, H. Mirzabozorg
Advanced Computational Engineering,

RC: ***Reviewers' Comment***, AR: Authors' Response, ☐ Manuscript Text

1. Reviewer #1

1.1. Comment 1

RC: ***Boundary-condition transform: scope & proof.***
The hard constraint $x(1-x)$ ***enforces*** $w(0) = w(L) = 0$***, but the text also claims exact satisfaction for fourth-order systems. Please clarify what is enforced exactly: displacement only, or slope/moment/shear as well? Provide derivations/examples for mixed BCs (e.g., clamped-free, clamped-pinned). If the method targets only Dirichlet displacement BCs, state limits explicitly.***

AR: We appreciate the reviewer's insightful comment. The hard-constrained transformation

$$w_\theta(x) = x(1-x)\,\mathrm{NN}(x)$$

enforces only the *Dirichlet displacement boundary conditions*, i.e., $w(0) = w(L) = 0$, exactly. The higher-order boundary quantities (slope, bending moment, and shear) are not automatically satisfied by this transformation and are instead imposed through the physics-informed loss terms derived from the governing differential equation and corresponding natural boundary conditions.

To clarify, for a fourth-order beam equation, the boundary conditions can be of mixed types:

- **Simply supported beam:** $w(0) = w(L) = 0$ (enforced exactly by the transform) and $M(0) = M(L) = 0$ (enforced weakly via loss).
- **Clamped-free beam:** $w(0) = w'(0) = 0$ (partially enforced using a modified transform, e.g. $x^2(3-2x)\mathrm{NN}(x)$) and $M(L) = V(L) = 0$ (included in the residual loss).
- **Clamped-pinned beam:** $w(0) = w'(0) = 0$, $w(L) = 0$, $M(L) = 0$ — a mixed case where $x(1-x)$ cannot fully encode all constraints and thus requires combined hard-soft enforcement.

Accordingly, we have revised the text to state explicitly that the proposed $x(1-x)$ transformation ensures exact satisfaction of displacement boundary conditions only. Other boundary quantities are treated as soft constraints through loss terms, which maintain high accuracy while preserving training flexibility.

1.2. Comment 2

RC: ***Adaptive weighting ablation.***
Support the scheduler $W_{BC}(t) = 10e^{-0.0001t}$ ***with ablation: constant vs.exponential, different initial***

weights/decays, and report sensitivity. Normalize loss terms (e.g., by BC/PDE residual magnitudes) to ensure the schedule is not just rescaling.

AR: In response to the reviewer's request on adaptive weighting, we trained a direct unweighted baseline (no exponential decay; constant loss weights). Results show that without the adaptive boundary-weight scheduler the model fails to converge to the high-precision solution: final total loss increased from 1.84×10^{-4} to 1.81×10^{9} and the relative L2 error increased from 0.056% to 6.95×10^{3}. This demonstrates the necessity of the proposed scheduler under our training protocol. A full ablation sweep (varying initial weights, decay rates and performing loss-term normalization) can be provided upon request.

1.3. Comment 3

RC: ***Gaussian regularization for point load.***
You cite rules for σ (function of L and collocation density N_c) but then fix $\sigma = 0.01L$. Provide a systematic study of σ: error vs. σ, and how it affects recovering the expected jump in shear/continuity in $w^{(3)}$. Report moment/shear fields and verify jump conditions at $x = L/2$.

AR: nan

1.4. Comment 4

RC: ***Quantitative evaluation & fairness of baselines.***
Provide tables of relative L_2 error and BC max violation for all cases, not only a single point-load. Compare against: (a) classical FEM with multiple mesh sizes; (b) a soft-constraint PINN; (c) a PINN with Fourier features/RAS as cited. Include wall-clock time and hardware; "5x speedup" needs evidence and precise conditions.

AR: nan

1.5. Comment 5

RC: ***Convergence claims vs. solution accuracy.***
Training loss values (e.g., $\mathcal{O}(10^{-10})$) do not necessarily reflect solution error. Plot validation error vs. iterations and show early-stopping behavior.

AR: nan

1.6. Comment 6

RC: ***Reproducibility package.***
Report multiple random seeds with mean $\pm$ std to demonstrate robustness. List all hyperparameters (depth/width, activation, optimizer, LR schedule, batch/collocation sizes, BC point sampling, PDE residual weighting, L-BFGS settings, gradient clipping, random seeds). Provide exact material/geometry parameters used (E, I, P, q, L, units) and the non-dimensionalization if any. If code is shared, ensure it's anonymized for double-blind review (the current GitHub mention would break anonymity).

AR: We have added a reproducibility table to the manuscript and provided an anonymized code archive in the supplementary materials. All reported performance numbers are computed as mean $\pm$ std across multiple random seeds.

1.7. Comment 7

RC: ***Higher-order derivatives & smoothness.***
Since Euler-Bernoulli uses $w^{(4)}$, discuss numerical stability of AD on deep networks and whether Sobolev training/gradient regularization or swish vs. tanh materially changes higher-order derivative quality.

AR: nan

Summary

The manuscript proposes three PINN enhancements for Euler–Bernoulli beam deflection problems:

1. a hard-constrained output transform $w_\theta(x) = x(1-x)\,\mathrm{NN}(x)$ to enforce fixed-end Dirichlet BCs;
2. an exponentially-decaying boundary-loss weight $W_{BC}(t) = 10e^{-0.0001t}$; and
3. a Gaussian regularization of Dirac delta point loads with $\sigma = 0.01L$.

Results are shown for cantilever, fully restrained, and mid-span point-loaded beams, with training curves and brief accuracy claims.

2. Reviewer #2

2.1. Comment 1

RC: ***Novelty and Comparative Assessment: The claim of a 37% convergence improvement via adaptive weighting—how was this quantified? Was it based on a direct comparison with a static weighting scheme under identical conditions (network architecture, optimizer, initializations)? Please provide more details or a side-by-side comparison figure/table.***

AR: nan

2.2. Comment 2

RC: ***Implementation Details and Reproducibility:***

- ***The GitHub link provided in the Limitations section appears to be broken. Please ensure the code is publicly available and the link is correct for the sake of reproducibility.***
- ***What was the exact number and spatial distribution of collocation points (N_c) for each case? Was uniform random sampling used, or a different strategy?***

AR: nan

2.3. Comment 3

RC: ***Error Analysis and Validation:***

- ***Why is the relative L_2 error for the point load case (0.56%) higher than the 0.30% reported by Zhang et al. (2020)? Does this indicate a limitation of the Gaussian regularization approach compared to other singularity-handling methods?***
- ***Was the Maximum Absolute Error computed for all cases? It is only reported in the comparative Table 6. Presenting it for all case studies would provide a more complete picture of solution accuracy.***

AR: nan

2.4. Comment 4

RC: ***Parameter Selection and Sensitivity:***

- ***Was a systematic sensitivity analysis performed on key hyperparameters, such as the Gaussian bandwidth (σ), the decay rate in $W_{BC}(t)$, or the network depth/width? The choice of $\sigma = 0.01L$ is stated as optimal, but the process for determining this should be explained (e.g., via a brief parametric study).***
- ***Material Property: The material properties are defined only via the composite parameter EI. Please clarify: What specific material is being modeled? (e.g., structural steel with $E = 200$ GPa, or a generic material?). This is crucial for readers to contextualize the physical scale of the deflections and the value of $EI = 200\ \mathrm{N} \cdot \mathrm{m}^2$ used in the point load example, which seems unusually low for real-world beams (suggesting a lab-scale or normalized example).***

AR: nan

2.5. Comment 5

RC: ***Scalability and Generalizability:***

- ***Has the proposed framework been tested on more complex boundary conditions (e.g., multi-span beams, springs) or time-dependent loads?***
- ***What are the anticipated primary challenges in extending this 1D methodology to 2D plate or 3D solid mechanics problems, beyond those mentioned in the limitations?***

AR: nan

2.6. Comment 6

RC: ***Convergence Analysis:***

- ***Was the identified three-phase convergence pattern (boundary fitting, physics compliance, fine-tuning) consistent across all three benchmark problems? Can this behavior be justified analytically or linked to the properties of the optimizer?***

AR: nan

2.7. Comment 6

RC: ***Comparison with Classical Methods:***

- ***Beyond setup time, was a direct comparison of computational runtime and memory usage performed between the proposed PINN and a standard FEM solver (e.g., Abaqus or FEniCS) for an equivalent level of accuracy?***

AR: nan