

Multi-Level Physics-Informed Deep Learning for Solving Partial Differential Equations in Computational Structural Mechanics



Open Access This file is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. In the cases where the authors are anonymous, such as is the case for the reports of anonymous peer reviewers, author attribution should be to 'Anonymous Referee' followed by a clear attribution to the source work. The images or other third party material in this file are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

Reviewers' comments:

Reviewer #1 (Remarks to the Author):

Review: COMMS-23-0444-T

Recommendation: Reject (Resubmit)

The study presents an implementation of a Physics-Informed Neural Network (PINN) applied to academic example problems from structural mechanics. Multiple neural networks are trained to compute individual contributions to the investigated Partial Differential Equations (PDE).

The paper explores a trending topic in the field, namely how to solve PDEs with neural-network-based methods efficiently. The paper features some expressive high-quality visualizations, presenting sound results on the selected applications.

However, in its present form, the manuscript is unsuitable for publication. Several critical issues need to be carefully addressed in the current form, which should be amended in a resubmitted version of the manuscript. In the following, specific comments are provided.

Major

1. The manuscript, in particular the text, appears still unfinished and draft-like. Almost every other sentence contains minor grammatical errors. In several places, repetitions and bloated formulations hurt the chain of arguments. Some statements are too broad and unsupported by investigations or references, in particular, when competing approaches are concerned. Some examples are given in the following.

2. Main text/Introduction: (l. 44-45) Infinite domains with unknown boundary conditions / initial conditions would specify an undefined problem. Can the authors please reformulate the statement and give a relevant motivating application example that cannot be solved by FEM but by PINNs?

3. Main text/Introduction: (l. 50-52) What about physics-explaining neural networks and the entire research field of interpretable/explainable AI?

4. Main text/Introduction: (l. 79-83) The statement is inaccurate and potentially misleading. Matrix/tensor computations are also an integral part of neural networks through randomly initialized weight and bias parameters. The number of training/collocation points does affect the result and can be regarded as a PINN-specific hyperparameter. However, considerable overlap, parallelism, and compatibility between the FEM and NN exist, as shown by examples of hybrid solutions that combine FEM and neural networks (intelligent (finite/meta) elements).

5. Results/Discussion: (l. 107-109) The broad claim that a single NN for multiple outputs is more computationally expensive than individual NNs for each output is not supported by the arguments or investigations in the paper. Computational efficiency is not evaluated against result quality, nor are references provided. In particular, neurons sharing functionality in a monolithic PINN approach can both stabilize learning and improve efficiency, which is not possible with the proposed method.

6. Results: The investigated beam and plate examples are academic. In particular, only linear elastic behavior and small strain approximations are considered. These problems can be consistently and efficiently solved by analytical and/or FEM models. What advantages does the proposed PINN method offer, since the PDEs need to be known for all approaches? Can the PINN deal with unknown/uncertain material parameters?

7. Methods/Results: (l. 119-121) How are the different loss contributions, which may exhibit different variances/scales, balanced? In conventional NN, separate normalization steps/scaling/weighting balance different output feature importance. How do the contributions to the losses change during the training process? How is the generalization of the trained NN validated? Can the errors in the result figures be attributed to specific contributions?

8. Results (l. 239-256): The selection of collocation points appears arbitrary. Commonly in tensorflow (l. 401), a first-order optimization algorithm such as ADAM trains the neural network over multiple epochs until the solution converges. Depending on the information content of the dataset (number of collocation points), this may need a varying amount of epochs for different numbers of collocation points. Therefore, the reasoning in the sensitivity analysis, that $n_p = 25$ balances computational precision and speed is flawed, as a lower number of collocation points (less information) means more epochs until sufficient information has flown through the neural network. Correspondingly, table 3 in the manuscript states that more collocation points are better than fewer collocation points.

9. Methods: The training procedure, hyperparameter tuning, and neural network architecture/hyperparameters are not specified.

Minor

1. (l. 87-89) The sentence is bloated.

2. "efficacy" should be "efficiency" throughout the manuscript.

3. The name of the proposed method changes from "multi-physics-informed neural network (m-PINN)" (l. 23, 114, 160, ...) to "multi-level physics-informed neural network" (l. 94) / "ml-PINN" (l. 345). Since multi-physics is a fixed term describing couplings between different physical fields (e.g., electrical and mechanical), the first term is misleading and exaggerates the novelty of the proposed approach.

4. (l. 91 and 109) The term "hard convergence" is ambiguous. Allegedly, the author means that achieving convergence is difficult (near divergence, unstable), but hard convergence may also mean that something converges rapidly.

5. (l. 130-133 and l. 133-136) Successive sentences repeat the same information. Maybe a copy-paste error?

The reviewer suggests that the manuscript be rejected at the present point. However, a resubmission of the interesting approach would be encouraged after carefully rewriting the draft into a well-crafted, concise paper.

Reviewer #2 (Remarks to the Author):

In the paper entitled "Multi-Physics-informed deep learning approach for computational structural mechanics", multiple Physics Informed Neural Networks (PINNs) are used in combination for the prediction of displacements in simulated beam (1D) and shell (2D) structures. The main argued benefit of the method is that results comparable to FEM are able to be achieved with less computational expense.

In its current form, it is the opinion of the reviewer that this work would need some modification before being suitable for publication in Communications Engineering. Below is an outline of some comments and suggested improvements:

1. Emphasis and evidence of what benefit the proposed method brings

There are several statements throughout the paper relating to the computational expense of Finite Element methods and standard PINN approaches, for example:

Line 39

"The repeated establishment and computation of FEM model in terms of model updating process hardly satisfies the real-time computation in the DT system due to the cumbersome and time-consuming updating process."

Line 107

"Nevertheless, using a single NN to process multi-order differentiations and multiple parameters will increase the computation burden, leading to large computation time and hard convergence."

Line 289

"However, compared to the FEM requiring the repeated model establishment (e.g., mesh generation, parameters input and calibration), the m-PINN with the ability of generalization learning has the potential to quickly predict the results for different loading cases once the NNs are trained well."

It appears that the proposed advantage of the m-PINN is that comparable results to FEM/ standard PINN are able to be achieved at a lower computational cost?

The results of the FEM and m-PINN are compared with a selection of studies in the results section and are able to conclude that yes, for beam and shell deflection models, m-PINN and FEM models produce comparable results. However what is missing is some evidence/investigations in to the following:

-What benefits in computation does the m-PINN have over FEM and a standard PINN e.g. a

comparison of runtimes/computation cost

-Comparisons of results between a m-PINN and standard PINN. Is the m-PINN closer to FEM? If so, why do you think this is?

2. Structuring of the paper

The paper is currently structured with sections as follows:

Abstract

Main text

Results

Discussion

Methods

Data availability/Code availability/Competing interests

I believe it could really help the reader of the paper if the structure was more similar to:

Abstract

Introduction

Theory/Methods

Results

Discussion

Conclusions

Data availability/Code availability/Competing interests

I believe the two major benefits of this change would be:

a). The addition of a distinct conclusions section.

This should aim to quickly explain to the reader the benefits/outcome of the work. e.g. "the m-PINN achieved results within X% accuracy of FEM and standard PINN approaches but with Y% less computational expense"

b). The presentation of methods and relevant theory before results are shown. This would allow the reader to understand/learn how methods are working before trying to interpret results.

It is the opinion of the review that efforts spend on these changes would be very worthwhile and add to the quality and readability of the paper.

Detailed Responses to Review Comments

COMMS-23-0444-T: Multi-level physics-informed deep learning
approach for computational structural mechanics

Weiwei He, Jinzhao Li, Xuan Kong, Lu Deng

March 4, 2024

Dear Editors and Reviewers,

Thank you for your comments and suggestions concerning our original manuscript. We revised the manuscript in accordance with the reviewers' comments. The revised portions related to addressing reviewer comments are marked in red in the revised manuscript.

Additionally, we have gone through the full paper thoroughly and made improvements to the English throughout, to improve overall readability. We have not specifically highlighted these in red, as they are not in any way substantial.

We sincerely appreciate your earnest and warm work, and hope that our corrections and consideration will meet with approval, and that the revised manuscript will be found acceptable. The detailed responses to the comments from reviewers are listed as follows with this letter.

Yours sincerely,
Xuan Kong (Corresponding author)

Responses to Reviewer #1

The study presents an implementation of a Physics-Informed Neural Network (PINN) applied to academic example problems from structural mechanics. Multiple neural networks are trained to compute individual contributions to the investigated Partial Differential Equations (PDE).

The paper explores a trending topic in the field, namely how to solve PDEs with neural-network-based methods efficiently. The paper features some expressive high-quality visualizations, presenting sound results on the selected applications.

However, in its present form, the manuscript is unsuitable for publication. Several critical issues need to be carefully addressed in the current form, which should be amended in a resubmitted version of the manuscript. In the following, specific comments are provided.

Response: The authors would like to thank the reviewer for your time and effort in reviewing our manuscript. We sincerely appreciate the reviewer's encouraging comments, which have greatly benefited us. All the suggestions and comments have been addressed point-by-point and the manuscript has been revised accordingly. Please find below our responses to each comment:

1. The manuscript, in particular the text, appears still unfinished and draft-like. Almost every other sentence contains minor grammatical errors. In several places, repetitions and bloated formulations hurt the chain of arguments. Some statements are too broad and unsupported by investigations or references, in particular, when competing approaches are concerned. Some examples are given in the following.

Response: We have gone through the full paper thoroughly and made improvements to the English throughout, to avoid grammatical errors and improve overall readability. The detailed revisions on some specific statements are listed as follows.

2. Main text/Introduction: (l. 44-45) Infinite domains with unknown boundary conditions/initial conditions would specify an undefined problem. Can the authors please reformulate the statement and give a relevant motivating application example that cannot be solved by FEM but by PINNs?

Response: PINNs can directly solve infinite domain problems without the need for imposing boundary conditions on model boundaries. In contrast, FEM is limited to solving finite domain problems and often requires modeling domains that are significantly larger than the target domains, with careful consideration of boundary conditions to mitigate boundary effects. PINN tackles infinite domain problems as a constrained optimization problem, where the boundary conditions are designed to enhance the predictive performance of neural networks. When the modeling domain is large enough to mask the local constraints of the target domain, boundary constraints can be ignored; otherwise, boundary constraints will be considered as one of the optimization objectives. This capability is also demonstrated in Reference [1]. In addition, PINN can solve the inverse problem but FEM cannot, e.g., the boundary condition parameters of the structure are inferred from the structural response. We have reformulated this statement and added the reference in the revised manuscript (Lines 46~48).

[1] Okazaki, T., Ito, T., Hirahara, K., & Ueda, N. (2022). Physics-informed deep learning approach for modeling crustal deformation. *Nature Communications*, 13(1), 7092.

3. *Main text/Introduction: (l. 50-52) What about physics-explaining neural networks and the entire research field of interpretable/explainable AI?*

Response: The conventional neural network is commonly regarded as a fitter of large amounts of data, which means that it mainly learns information from data as a purely data-driven method. However, the physics-explaining neural network such as PINN is to add the constraint of physical equations such as PDEs to the data. Therefore, physics-informed machine learning seamlessly integrates data and mathematical physics models, even in partially understood, uncertain and high-dimensional contexts[2].

[2] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422-440.

4. *Main text/Introduction: (l. 79-83) The statement is inaccurate and potentially misleading. Matrix/tensor computations are also an integral part of neural networks through randomly initialized weight and bias parameters. The number of training/collocation points does affect the result and can be regarded as a PINN-specific hyperparameter. However, considerable overlap, parallelism, and compatibility between the FEM and NN exist, as shown by examples of hybrid solutions that combine FEM and neural networks (intelligent (finite/meta) elements).*

Response: We agree with the review's opinion, and the statement has been revised as follows:

“Compared to the FEM approach that requires the generation of fine discrete meshes, PINNs directly model the continuous field, making them a meshless method. Moreover, PINNs demonstrate remarkable flexibility to solve problems by combining simulated data from FEM and relevant physical laws for training neural networks [3].” (Lines 81~84)

[3] Mitusch, S. K., Funke, S. W., & Kuchta, M. (2021). Hybrid FEM-NN models: Combining artificial neural networks with the finite element method. *Journal of Computational Physics*, 446, 110651.

5. *Results/Discussion: (l. 107-109) The broad claim that a single NN for multiple outputs is more computationally expensive than individual NNs for each output is not supported by the arguments or investigations in the paper. Computational efficiency is not evaluated against result quality, nor are references provided. In particular, neurons sharing functionality in a monolithic PINN approach can both stabilize learning and improve efficiency, which is not possible with the proposed method.*

Response: Thanks for the reviewer's suggestion, and we have added the comparison between the results obtained using classical PINN with a single NN and the ml-PINN with multiple NNs. The relevant content has been added in the revised manuscript (Lines 436~455). For your convenience, the corresponding text is presented below:

“The difference in prediction performance between the classical PINN and the proposed ml-PINN is compared as illustrated in Fig. A1. The figure here only shows the results on Case 1 of the 1D beam structure as can be seen in Table. 1 and Fig. 4b. It clearly shows that ml-PINN outperforms classical PINN in terms of prediction accuracy and computational efficiency. In addition, Table. A1 provides the detailed results of the four cases on the 1D beam structure using both the classical PINN and the ml-PINN (see Table. 1 for the detail of 4 cases). As shown in the Table. A1, after 10^4 iterations, the classical PINN still exhibits significantly higher errors in predicted displacements compared to ml-PINN. Meanwhile, the classical PINN has a quarter of the training parameters of ml-PINN, but four

times the computation time.”

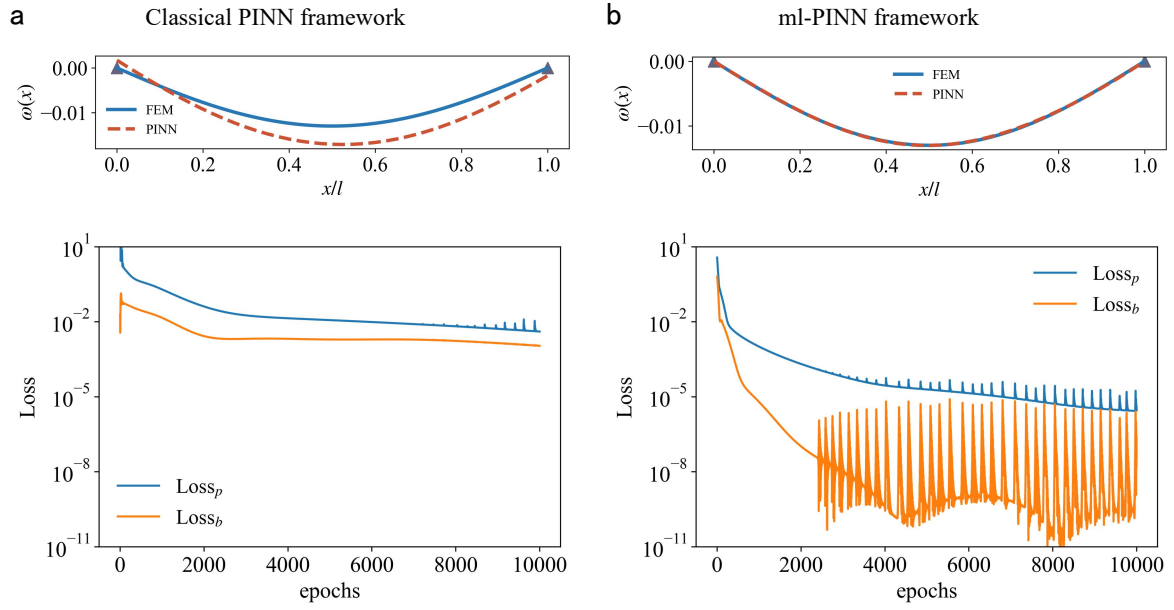


Fig. A1 | Comparison results of computation for 1D beam structure using classical PINN and ml-PINN. **a** Predicted outcomes and loss decline curve of classical PINN exhibit poor prediction results and convergence difficulties. $Loss_b$ and $Loss_p$ represent the loss functions from boundary constraints and loss functions from PDE, respectively. **b** Predicted outcomes and loss decline curve of ml-PINN demonstrate rapid convergence of the loss function.

Table. A1 Comparison results of computation for 1D beam structure obtained by classical PINN and ml-PINN

Cases	Methods	Error (%)	Loss _p	Loss _b	Time (s/epoch)
case 1	c-PINN	19.6	3.6×10^{-4}	3.7×10^{-5}	4.2×10^{-3}
	ml-PINN	0.225	2.8×10^{-6}	8.8×10^{-10}	1.1×10^{-3}
case 2	c-PINN	107.4	2.4×10^{-4}	2.4×10^{-5}	4.2×10^{-3}
	ml-PINN	2.47	3.4×10^{-5}	8.8×10^{-8}	1.1×10^{-3}
case 3	c-PINN	46.7	1.9×10^{-1}	8.0×10^{-4}	4.4×10^{-3}
	ml-PINN	0.087	3.8×10^{-5}	1.5×10^{-7}	1.1×10^{-3}
case 4	c-PINN	104.3	2.6×10^{-2}	6.8×10^{-4}	4.4×10^{-3}
	ml-PINN	0.084	4.6×10^{-6}	9.7×10^{-9}	1.1×10^{-3}

Notes: The c-PINN is an abbreviation for classical PINN. $Loss_p$ and $Loss_b$ denote the loss function of the

PDEs and the loss function of boundary constraint, respectively. $Error = \frac{\|\omega_{PINN} - \omega_{FEM}\|_2^2}{\|\omega_{FEM}\|_2^2}$ where ω_{PINN} and

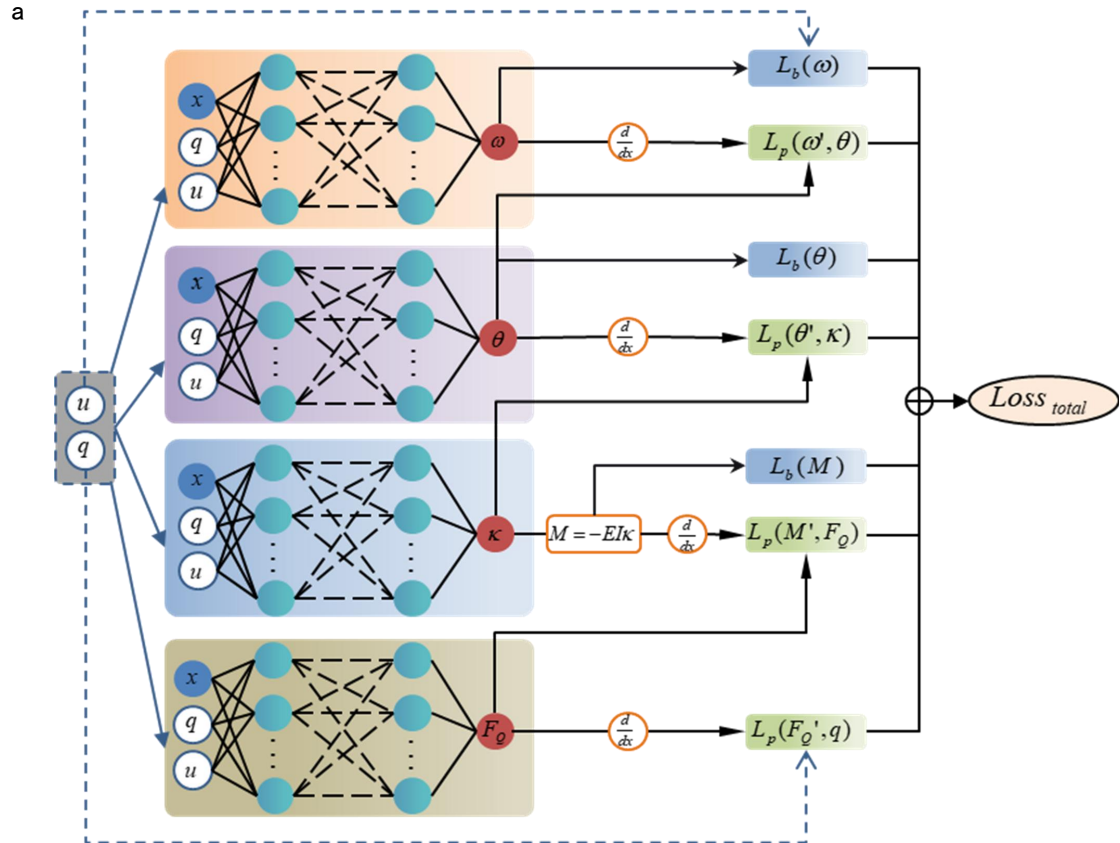
ω_{FEM} represent the value of the vertical displacement computed by ml-PINN and FEM respectively.

6. *Results: The investigated beam and plate examples are academic. In particular, only linear elastic behavior and small strain approximations are considered. These problems can be consistently and efficiently solved by analytical and/or FEM models. What advantages does the proposed PINN method*

offer, since the PDEs need to be known for all approaches? Can the PINN deal with unknown/uncertain material parameters?

Response: The disadvantage of FEM mainly lies in its repetitive computation for similar working conditions. However, PINN possess enhanced flexibility and capability to perform generalization learning, which means that PINN can directly predict the structural response based on generalized parameters, thereby achieving efficient end-to-end output. Currently, our research is primarily focused on advancement of this field; however, it is crucial to emphasize that the generalized computation is built on successful individual computations. The relevant content has been added in the revised manuscript (Lines 298~316), and for your convenience, the corresponding text is presented below :

“Currently, the primary disadvantage of ml-PINN lies in its time-consuming training process, which is a general problem yielded in PINN when solving the forward problem. However, once the ml-PINN is trained well, it can serve as a surrogate model to enable efficient end-to-end prediction. Overall, the aforementioned results and discussion indicate that the ml-PINN achieves comparable accuracy to FEM for solving bending structures. Moreover, unlike the FEM requiring repeated model establishment (e.g., mesh generation, parameters adjustment and calibration) and computation for different conditions, the ml-PINN possesses generalization learning capabilities and has the potential to rapidly predict the results for different loading cases once the NNs are well-trained. Fig. 7a shows the ml-PINN framework for generalization learning, where the physical variables related to loading conditions (u and q) are considered as additional input parameters of the network. Fig.7b exhibits the direct prediction performance under different loading conditions, indicating that incorporating generalization into ml-PINN significantly improves computational efficiency while maintaining comparable accuracy with FEM computation.”



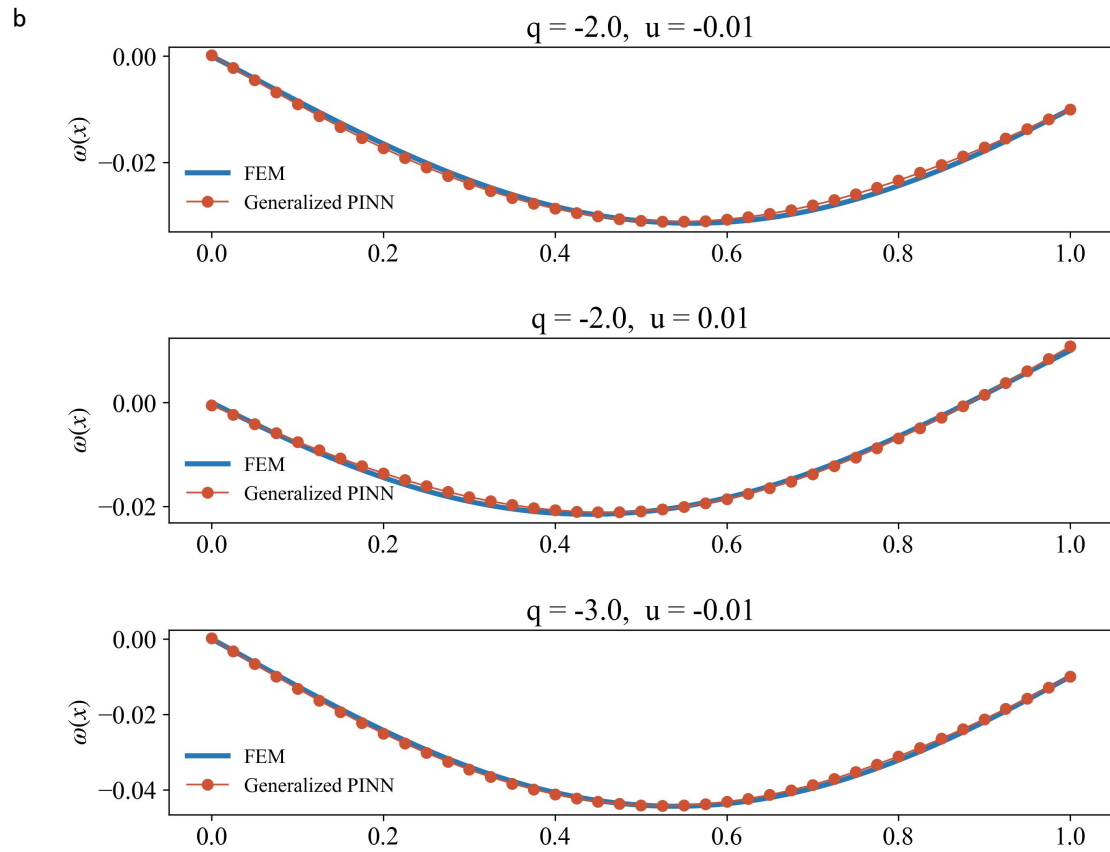


Fig. 7 | The ml-PINN for generalization computation. **a** Framework of ml-PINN for generalization learning by incorporating the physical variables related to loading conditions (u and q) as input parameters of the networks. **b** Direct prediction results of ml-PINN for bending beam cases under different loading conditions.

7. *Methods/Results: (l. 119-121) How are the different loss contributions, which may exhibit different variances/scales, balanced? In conventional NN, separate normalization steps/scaling/weighting balance different output feature importance. How do the contributions to the losses change during the training process? How is the generalization of the trained NN validated? Can the errors in the result figures be attributed to specific contributions?*

Response: In conventional PINN, using a single NN to handle multi-order differentiations and multiple parameters can significantly increase computation burden, leading to prolonged computation time and unstable convergence (can be seen from the aforementioned response for Question 5). The reason is that when a neural network undergoes automatic differentiation many times, the accumulation of errors become exponentially. Moreover, the varying orders of magnitude among different differentiation terms pose difficulties for network convergence (e.g., 10^{-1} vs. 10^{-10}). Herein we alleviate this issue by reducing the higher-order differentiation terms to multiple first-order or second-order terms and using multiple NNs to train these lower-order terms, which makes the weight of each NN is at a similar level. This strategy facilitates the neural networks to converge.

8. *Results (l. 239-256): The selection of collocation points appears arbitrary. Commonly in tensorflow (l. 401), a first order optimization algorithm such as ADAM trains the neural network over multiple epochs until the solution converges. Depending on the information content of the dataset (number of*

collocation points), this may need a varying amount of epochs for different numbers of collocation points. Therefore, the reasoning in the sensitivity analysis, that $n_p = 25$ balances computational precision and speed is flawed, as a lower number of collocation points (less information) means more epochs until sufficient information has flown through the neural network. Correspondingly, table 3 in the manuscript states that more collocation points are better than fewer collocation points.

Response: There is no doubt that more collocation points yield better results compared to fewer collocation points. Inadequate collocation points fail to provide sufficient information for accurate predictions. However, once the number of collocation points reaches a certain threshold (e.g., $n_p = 25$ in beam element), the prediction accuracy tends to be independent of the number of input data points. It is worth mentioning that the collection of location points here is not a concept of training datasets, but rather as a mesh grid similar to FEM. The more points there are, the higher the of the results, but it also requires more computational resources. This aspect has been elucidated in detail within the revised manuscript (Lines 247~257). For your convenience, the corresponding text is presented below.

“For the 1D beam (Fig. 6a), the computational error is relatively high (close to 3%) when rather sparse collocation points are used ($n_p = 10$), while the error is reduced less than 1% as the number of collocation points increases ($n_p > 25$). Thus, to balance the computational precision and speed, the number of points $n_p = 25$ has been adopted. It is concluded that more collocation points are better than fewer collocation points. When there are fewer allocation points, there is not enough information in the data points to support a better prediction effect. As the number of allocation points reaches a certain value ($n_p = 25$ in beam element for the studied case), the prediction effect will be almost independent of the number of input data points. It is noted that the number of allocation points can be determined by sensitivity analysis in different applications”

9. *Methods: The training procedure, hyperparameter tuning, and neural network architecture/hyperparameters are not specified.*

Response: The essential information on the hyperparameters of the neural network has been added to the section of Method/Model setting of the revised manuscript (Lines 415~424). For your convenience, the corresponding text is presented below:

“All codes are optimized using ADAM algorithm with a learning rate of 10^{-3} . A total of four neural networks (NNs) were established to predict the output of ω , θ , M and F_Q in the 1D beam structure, namely ω_{NN} , θ_{NN} , M_{NN} and F_{QNN} . All four fully connected feedforward NNs consist of an input layer with 1 node, 8 hidden layers with 40 nodes, and an output layer with 1 node. Similarly, for the 2D shell structure, three NNs were set up as ω_{NN} , $\kappa_{NN}(\kappa_x, \kappa_y, \text{ and } \kappa_{xy})$, $M_{NN}(M_x, M_y, \text{ and } M_{xy})$. The NN of ω_{NN} consists of an input layer with 2 nodes, 3 hidden layers with 30 nodes, and an output layer with 1 node. Additionally, the NNs used in κ_{NN} and M_{NN} were constructed with an input layer with 2 nodes, 3 hidden layers with 20 nodes, and an output layer with 3 nodes.”

10. *Minor*

(1) (l. 87-89) *The sentence is bloated.*

(2) *“efficacy” should be “efficiency” throughout the manuscript.*

(3) *The name of the proposed method changes from “multi-physics-informed neural network (m-PINN)” (l.23, 114, 160, ...) to “multi-level physics-informed neural network” (l. 94) / “ml-PINN” (l. 345). Since multi-physics is a fixed term describing couplings between different physical fields (e.g., electrical*

and mechanical), the first term is misleading and exaggerates the novelty of the proposed approach.

(4) (l. 91 and 109) The term “hard convergence ” is ambiguous. Allegedly, the author means that achieving convergence is difficult (near divergence, unstable), but hard convergence may also mean that something converges rapidly.

(5) (l. 130-133 and l. 133-136) Successive sentences repeat the same information. Maybe a copy-paste error?

Response: (1) The sentence has been revised as: “The governing equations for bending structures are characterized by fourth-order nonlinear PDEs featuring multiple orders of differential terms, which significantly decreases the accuracy when it comes to higher-order differentiation.” (Lines 87~90)

(2) The “efficacy” has been revised as “efficiency”.

(3) The “multi-physics-informed neural network (m-PINN)” has been revised as “multi-level physics-informed neural network (ml-PINN)”.

(4) The “hard convergence” has been revised as “unstable convergence”.

(5) It is not a copy-paste error. To avoid misunderstandings, the sentence has been revised as:

“The governing equation for the beam structure is a fourth-order differential equation relating q and ω , which can be transformed into five first-order equations. The five equations actually correspond to the three fundamental equations in structural mechanics: the geometric equation, stress-strain equation, and equilibrium equation with physical variables (i.e., θ , κ , M , and F_Q) as transition parameters (see ‘Methods’ for details).” (Lines 133~138)

Responses to Reviewer #2

In the paper entitled “Multi-Physics-informed deep learning approach for computational structural mechanics”, multiple Physics Informed Neural Networks (PINNs) are used in combination for the prediction of displacements in simulated beam (1D) and shell (2D) structures. The main argued benefit of the method is that results comparable to FEM are able to be achieved with less computational expense. In its current form, it is the opinion of the reviewer that this work would need some modification before being suitable for publication in Communications Engineering. Below is an outline of some comments and suggested improvements:

Response: The authors would like to thank the reviewer for your time and effort in reviewing our manuscript. We sincerely appreciate the reviewer’s encouraging comments, which have greatly benefited us. All the suggestions and comments have been addressed point-by-point and the manuscript has been revised accordingly. Please find below our responses to each comment.

1. Emphasis and evidence of what benefit the proposed method brings. There are several statements throughout the paper relating to the computational expense of Finite Element methods and standard PINN approaches, for example:

Line 39 “The repeated establishment and computation of FEM model in terms of model updating process hardly satisfies the real-time computation in the DT system due to the cumbersome and time-consuming updating process.”

Line 107 “Nevertheless, using a single NN to process multi-order differentiations and multiple parameters will increase the computation burden, leading to large computation time and hard convergence.”

Line 289 “However, compared to the FEM requiring the repeated model establishment (e.g., mesh generation, parameters input and calibration), the m-PINN with the ability of generalization learning has the potential to quickly predict the results for different loading cases once the NNs are trained well.”

It appears that the proposed advantage of the m-PINN is that comparable results to FEM/ standard PINN are able to be achieved at a lower computational cost?

The results of the FEM and m-PINN are compared with a selection of studies in the results section and are able to conclude that yes, for beam and shell deflection models, m-PINN and FEM models produce comparable results.

However what is missing is some evidence/investigations in to the following:

-What benefits in computation does the m-PINN have over FEM and a standard PINN e.g. a comparison of runtimes/computation cost

-Comparisons of results between a m-PINN and standard PINN. Is the m-PINN closer to FEM? If so, why do you think this is?

Response: Thanks for the reviewer’s kind concern. Regarding the advantage of m-PINN (renamed as ml-PINN according to the previous reviewer’s suggestion) over both standard PINN and FEM, The comparison of results in terms of computational accuracy and efficiency has been added in the revised manuscript (Lines 436~455 and 298~316). For ease of reference, the corresponding text is presented as follows:

Comparison between classical PINN and ml-PINN:

“The difference in prediction performance between the classical PINN and the proposed ml-PINN is compared as illustrated in Fig. A1. The figure here only shows the results on Case 1 of the 1D beam structure as can be seen in Table. 1 and Fig. 4b. It clearly shows that ml-PINN outperforms classical PINN in terms of prediction accuracy and computational efficiency. In addition, Table. A1 provides the detailed results of the four cases on the 1D beam structure using both the classical PINN and the ml-PINN (see Table. 1 for the detail of 4 cases). As shown in the Table. A1, after 10^4 iterations, the classical PINN still exhibits significantly higher errors in predicted displacements compared to ml-PINN. Meanwhile, the classical PINN has a quarter of the training parameters of ml-PINN, but four times the computation time.”

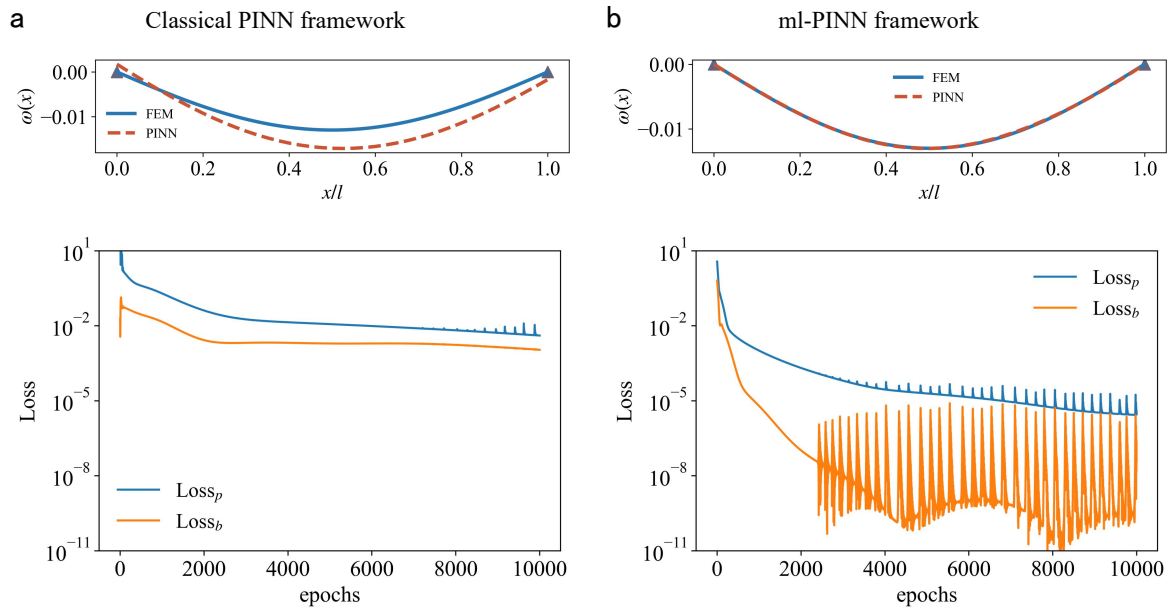


Fig. A1 | Comparison results of computation for 1D beam structure using classical PINN and ml-PINN.

a Predicted outcomes and loss decline curve of classical PINN exhibit poor prediction results and convergence difficulties. $Loss_b$ and $Loss_p$ represent the loss functions from boundary constraints and loss functions from PDE, respectively. **b** Predicted outcomes and loss decline curve of ml-PINN demonstrate rapid convergence of the loss function.

Table. A1 Comparison results of computation for 1D beam structure obtained by classical PINN and ml-PINN

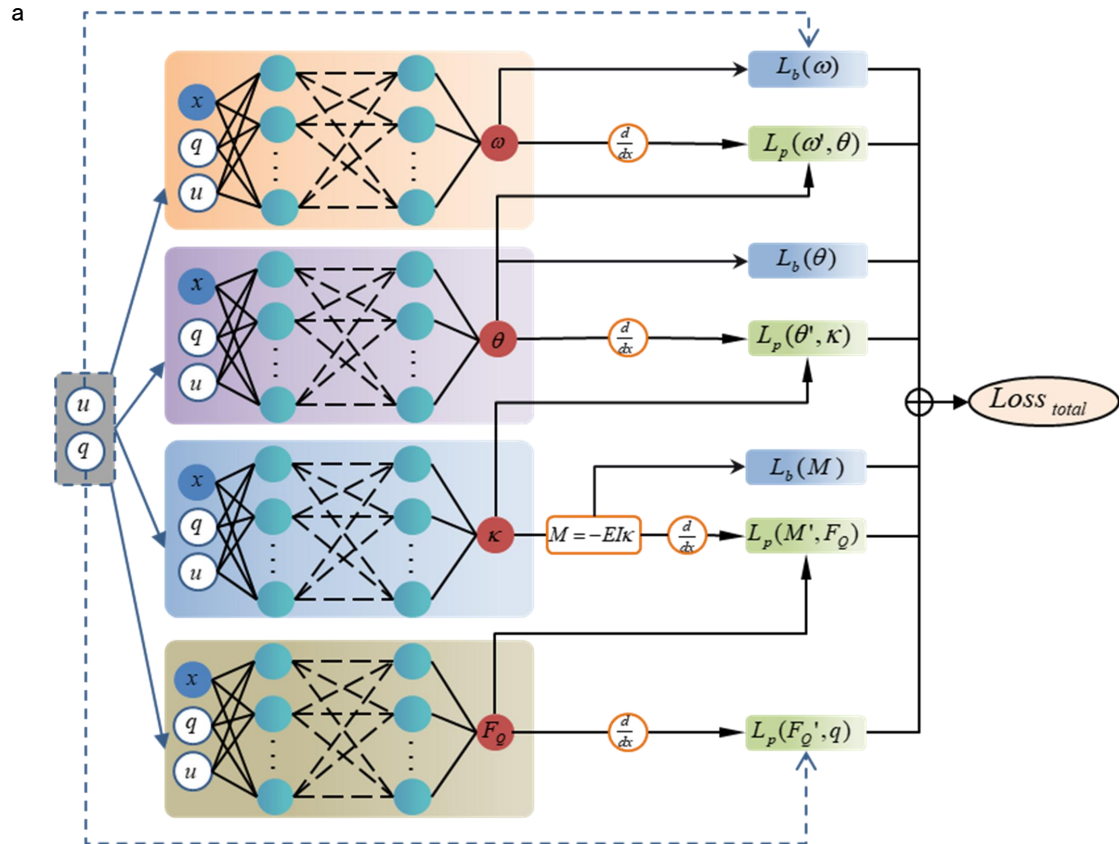
Cases	Methods	Error (%)	Loss _p	Loss _b	Time (s/epoch)
Case 1	c-PINN	19.6	3.6×10^{-4}	3.7×10^{-5}	4.2×10^{-3}
	ml-PINN	0.225	2.8×10^{-6}	8.8×10^{-10}	1.1×10^{-3}
Case 2	c-PINN	107.4	2.4×10^{-4}	2.4×10^{-5}	4.2×10^{-3}
	ml-PINN	2.47	3.4×10^{-5}	8.8×10^{-8}	1.1×10^{-3}
Case 3	c-PINN	46.7	1.9×10^{-1}	8.0×10^{-4}	4.4×10^{-3}
	ml-PINN	0.087	3.8×10^{-5}	1.5×10^{-7}	1.1×10^{-3}
Case 4	c-PINN	104.3	2.6×10^{-2}	6.8×10^{-4}	4.4×10^{-3}
	ml-PINN	0.084	4.6×10^{-6}	9.7×10^{-9}	1.1×10^{-3}

Notes: The c-PINN is an abbreviation for classical PINN. $Loss_p$ and $Loss_b$ denote the loss function of the

PDEs and the loss function of boundary constraint, respectively.
$$\text{Error} = \frac{\|\omega_{PINN} - \omega_{FEM}\|_2^2}{\|\omega_{FEM}\|_2^2}$$
 where ω_{PINN} and ω_{FEM} represent the value of the vertical displacement computed by ml-PINN and FEM respectively.

Comparison between FEM and ml-PINN:

“Currently, the primary disadvantage of ml-PINN lies in its time-consuming training process, which is a general problem yielded in PINN when solving the forward problem. However, once the ml-PINN is trained well, it can serve as a surrogate model to enable efficient end-to-end prediction. Overall, the aforementioned results and discussion indicate that the ml-PINN achieves comparable accuracy to FEM for solving bending structures. Moreover, unlike the FEM requiring repeated model establishment (e.g., mesh generation, parameters adjustment and calibration) and computation for different conditions, the ml-PINN possesses generalization learning capabilities and has the potential to rapidly predict the results for different loading cases once the NNs are well-trained. Fig. 7a shows the ml-PINN framework for generalization learning, where the physical variables related to loading conditions (u and q) are considered as additional input parameters of the network. Fig. 7b exhibits the direct prediction performance under different loading conditions, indicating that incorporating generalization into ml-PINN significantly improves computational efficiency while maintaining comparable accuracy with FEM computation.”



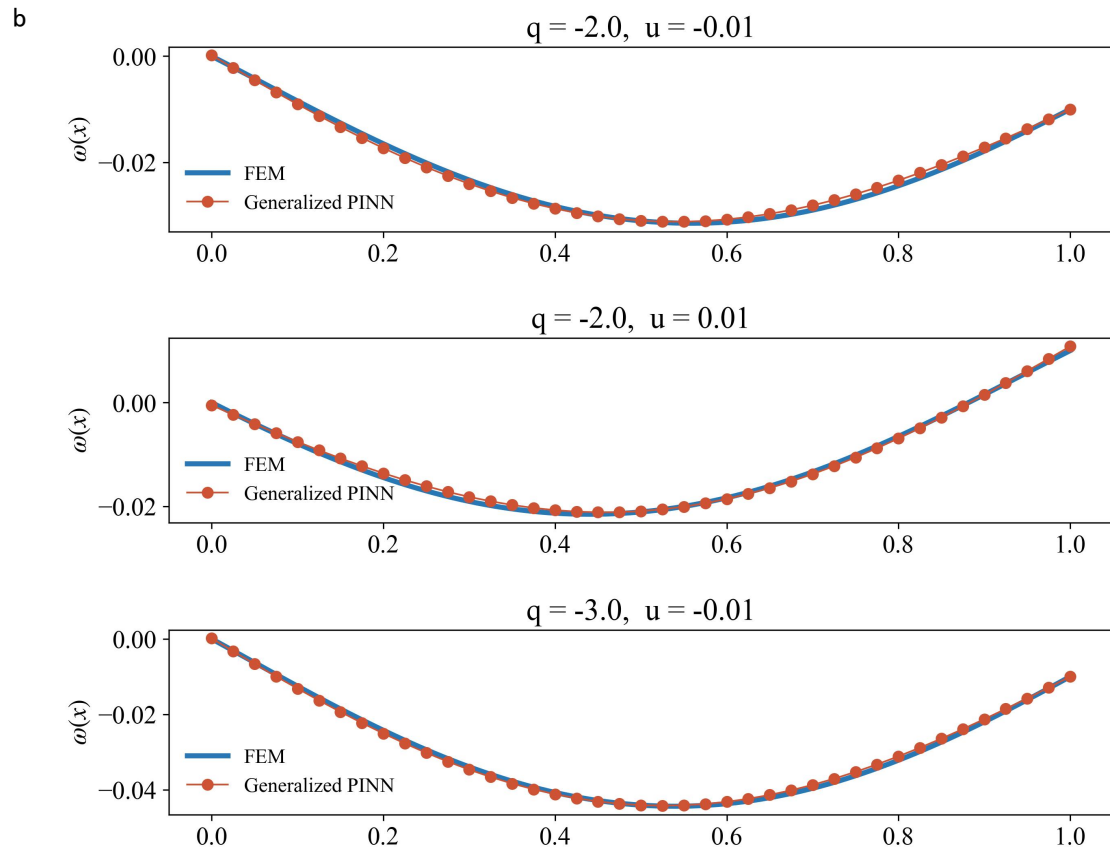


Fig. 7 | The ml-PINN for generalization computation. **a** Framework of ml-PINN for generalization learning by incorporating the physical variables related to loading conditions (u and q) as input parameters of the networks. **b** Direct prediction results of ml-PINN for bending beam cases under different loading conditions.

2. Structuring of the paper:

The paper is currently structured with sections as follows: Abstract/ Main text / Results / Discussion / Methods / Data availability/Code availability/Competing interests. I believe it could really help the reader of the paper if the structure was more similar to: Abstract / Introduction / Theory/Methods / Results / Discussion / Conclusions / Data availability/Code availability/Competing interests.

I believe the two major benefits of this change would be: a). The addition of a distinct conclusions section. This should aim to quickly explain to the reader the benefits/outcome of the work. e.g. “the m-PINN achieved results within X% accuracy of FEM and standard PINN approaches but with Y% less computational expense” b). The presentation of methods and relevant theory before results are shown. This would allow the reader to understand/learn how methods are working before trying to interpret results. It is the opinion of the review that efforts spend on these changes would be very worthwhile and add to the quality and readability of the paper.

Response: We appreciate the reviewer's suggestions on the structure of the article. “Abstract / Main text / Results / Discussion / Methods / Data availability / Code availability / Competing interests” is the universal typesetting mode of Nature series journals. The purpose of putting Results in the front is to make it easier for the reader to get to the point firstly and directly. However, in this paper, the essential methodological information involving the framework description has been provided prior to interpreting

results such that the readers can understand the general method. Further details on the method and the common knowledge are presented at the end of the manuscript. Additionally, according to the reviewer's suggestion, the clear conclusion highlighting the advantage of ml-PINN has been included in the Discussion section, as "It is evident that the proposed ml-PINN framework significantly enhances computational accuracy by reducing the error of beam and shell structures computation to 0.5% and 2%, respectively, surpassing that achieved by classical PINN, while simultaneously decreasing computational time." (Lines 291~294)

Reviewers' comments:

Reviewer #1 (Remarks to the Author):

Review: COMMS-23-0444-A
Resubmission of COMMS-23-0444-T
Recommendation: Major Revisions

The authors have responded to and addressed most of the reviewer's concerns. However, the following concerns from the reviewer's comments have not been answered or follow the additional information given by the authors in response to the reviewer's comments.

Follow-up comment #1.3 Main text/Introduction (l. 50-52)

The authors appear to have mixed up the terms physics-explaining and physics-informed in their response. Interpretable and physics-explaining learning are methods of explainable AI applied to conventional neural networks trained on physical problems. Those methods can illuminate the blackbox mentioned in the text and are relevant to mention, if that specific short-coming of purely data-driven vs physics-informed methods is discussed.

<https://doi.org/10.1016/j.engstruct.2023.115723>

<https://doi.org/10.3389/fmats.2021.824958>

<https://doi.org/10.1016/j.compind.2020.103359>

Follow-up comment #1.5 Results/Discussion (l. 107-109)

Since all neural networks are identical in their architecture and hyperparameters (author's response to Comment 9), the ml-PINN (3-4 NNs) consists of 3-4 times (beam vs. plate) the capacity of the conventional PINN. As such, the improved performance of the ml-PINN is not surprising. Since the capacity for fully connected neural networks scales quadratically with the number of units per layer, the reviewer suggests comparing a c-PINN with 80 units per layer with an ml-PINN consisting of 4 neural networks with 40 units per layer. Please also ensure optimal hyperparameters (base learning rates for ADAM, ...) are used for both models.

Follow-up comment #1.9 Methods

Since no hyperparameter tuning is reported that usually optimizes the architectures of the neural networks for the two considerably different problems of the c-PINN and the ml-PINN (different difficulty due to the number of outputs per network), it is not clear if the results of the c-PINN are the optimal solution for the given physical problem. Please ensure optimal hyperparameters through a hyperparameter tuning step.

The reviewer suggests that the manuscript be revised.

Reviewer #2 (Remarks to the Author):

"Response: The authors would like to thank the reviewer for your time and effort in reviewing our manuscript. We sincerely appreciate the reviewer's encouraging comments, which have greatly benefited us. All the suggestions and comments have been addressed point-by-point and the manuscript has been revised accordingly. Please find below our responses to each comment."

Thank you to the authors for their effort in revising the manuscript. I am much happier with the ability of the paper to present the advantages of proposed method. I would now recommend it be allowed to be published.

The ml-PINN is able to achieve comparable results to FEM on simulated beam and shell models. It also outperforms a standard PINN (lower error, lower computation cost) for this task.

The beam and shell examples used are well understood in engineering, with analytical solutions available. Although this makes them familiar to the reader, the results achieved are therefore of limited use on their own. This paper therefore mainly serves as a display of the method and its potential advantages, rather than the focus being on application.

"Response: Thanks for the reviewer's kind concern. Regarding the advantage of m-PINN (renamed as ml-PINN according to the previous reviewer's suggestion) over both standard PINN and FEM, The comparison of results in terms of computational accuracy and efficiency has been added in the revised manuscript (Lines 436~455 and 298~316). For ease of reference, the corresponding text is presented as follows:"

Evidence and discussion of the benefits the proposed ml-PINN over FEM and standard PINNs was the major element missing from the original paper and the added sections address this specifically. I believe that the manuscript is stronger with this inclusion.

"Response: We appreciate the reviewer's suggestions on the structure of the article. "Abstract / Main text / Results / Discussion / Methods / Data availability / Code availability / Competing interests" is the universal typesetting mode of Nature series journals. The purpose of putting Results in the front is to make it easier for the reader to get to the point firstly and directly. However, in this paper, the essential methodological information involving the framework description has been provided prior to interpreting results such that the readers can understand the general method. Further details on the method and the common knowledge are presented at the end of the manuscript."

If that is the universal typeset of the journal, then yes, I'm happy to remain with the current paper structure. With some of the rewording present in the revised manuscript, the proposed method is presented to the reader more clearly than before. This helps the reader interpret results shown.

"Additionally, according to the reviewer's suggestion, the clear conclusion highlighting the advantage of ml-PINN has been included in the Discussion section, as "It is evident that the proposed ml-PINN framework significantly enhances computational accuracy by reducing the error of beam and shell structures computation to 0.5% and 2%, respectively, surpassing that achieved by classical PINN, while simultaneously decreasing computational time. " (Lines 291~294)"

Concisely stating the advantages of the method was a very welcome addition. This helps to motivate the work of the paper. I believe a similar statement could be used at the end of the abstract. It currently reads:

"The verification results demonstrate that ml-PINN achieves comparable accuracy to FEM in computing beam and shell structures. The proposed method holds the potential to become a new paradigm for structural mechanics computation and facilitate the intelligent computation of digital twin systems."

This statement, especially "intelligent computation of digital twin systems" could mean any number of things and I believe a more specific statements such as "4 times less computation time than a standard PINN" would be better in place of this. In general, I would only make claims that are able to be backed up with evidence (results or citations).

Detailed Responses to Review Comments

COMMS-23-0444-A: Multi-level physics-informed deep learning
approach for computational structural mechanics

Weiwei He, Jinzhao Li, Xuan Kong, Lu Deng

May 18, 2024

Your revised manuscript entitled "Multi-level physics-informed deep learning approach for computational structural mechanics" has now been seen by two referees. As you will see from their comments below that they are quite delighted to see the progress of the manuscript. However, important issues do remain. We are interested in the possibility of publishing your study in Communications Engineering, but would like to consider your response to these concerns in the form of a revised manuscript before we make a final decision on publication.

Dear Editors and Reviewers,

Thank you for your comments and suggestions regarding our revised manuscript. In this round of review, we have diligently incorporated the reviewers' feedback into our manuscript. The revised portions addressing the reviewers' comments are highlighted in red within the revised version.

Additionally, we have gone through the entire paper thoroughly and made improvements to the English throughout, to improve overall readability. We have not specifically highlighted these in red, as they are not in any way substantial.

We sincerely appreciate your earnest and warm work, and hope that our corrections and consideration will meet with approval, and that the revised manuscript will be found acceptable. The detailed responses to the comments from reviewers are provided below along with this letter.

Yours sincerely,
Xuan Kong (Corresponding author)

Responses to Reviewer #1

The authors have responded to and addressed most of the reviewer's concerns. However, the following concerns from the reviewer's comments have not been answered or follow the additional information given by the authors in response to the reviewer's comments.

1. Follow-up comment #1.3 Main text/Introduction (l. 50-52)

The authors appear to have mixed up the terms physics-explaining and physics-informed in their response. Interpretable and physics-explaining learning are methods of explainable AI applied to conventional neural networks trained on physical problems. Those methods can illuminate the blackbox mentioned in the text and are relevant to mention, if that specific short-coming of purely data-driven vs physics-informed methods is discussed.

<https://doi.org/10.1016/j.engstruct.2023.115723>

<https://doi.org/10.3389/fmats.2021.824958>

<https://doi.org/10.1016/j.compind.2020.103359>

Response: Thanks for the reviewer's reminder. The physics-explaining machine learning could explain the relationships between variables in a conventional model trained on physical problems. For instance, we could utilize SHAP (SHapley Additive exPlanations) to explain the output prediction by calculating the contribution of each input variable (feature) to the final prediction, and then explain the "black-box model" from both global and local perspectives. The physics-informed deep learning is directly imposed to make the model satisfy the physical prior knowledge, especially the physical differential equations. The previous content was not sufficiently rigorous in presentation and has been appropriately revised. Additionally, we have included the three references suggested by the reviewer in this revised version.

Previous content: "However, the neural networks still remain as black boxes that lack explicit physical descriptions, relying heavily on large amounts of high-fidelity data. To address this issue, the physics-informed neural network (PINN) was proposed, which incorporates underlying physical laws by integrating them into the network."

The previous content has been revised as: "Data-driven network models rely on large amounts of high-quality data and mostly lack physical interpretability. Some studies currently explain the black box model by employing SHAP (i.e., SHapley Additive exPlanations) to quantify the contribution of each input feature to the output prediction [21-23]. However, the physics-explaining machine learning still finds it difficult to directly learn physics prior knowledge. To address this issue, the physics-informed neural network (PINN) was proposed, which has the capability to learn the priori knowledge of partial differential equations with or without labeled data." (Lines 51~58 in revised manuscript)

[21] Lai, D., Demartino, C., Xiao, Y. (2023). Interpretable machine-learning models for maximum displacements of RC beams under impact loading predictions. *Engineering Structures*, 281, 115723.

[22] Onchis, D.M., Gillich, G.R. (2021). Stable and explainable deep learning damage prediction for prismatic cantilever steel beam. *Computers in Industry*, 125, 103359.

[23] Koepe, A., Bamer, F., Selzer, M., Nestler, B., Markert, B. (2022). Explainable artificial intelligence for mechanics: physics-explaining neural networks for constitutive models. *Frontiers in Materials*, 8, 824958.

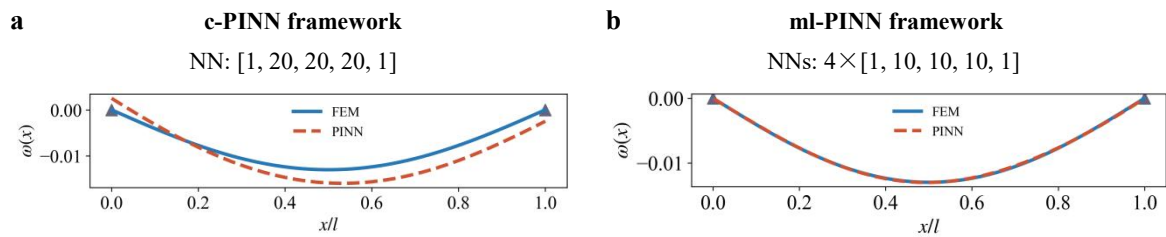
2. Follow-up comment #1.5 Results/Discussion (l. 107-109)

Since all neural networks are identical in their architecture and hyperparameters (author's response to Comment 9), the ml-PINN (3-4 NNs) consists of 3-4 times (beam v.s. plate) the capacity of the conventional PINN. As such, the improved performance of the ml-PINN is not surprising. Since the capacity for fully connected neural networks scales quadratically with the number of units per layer, the reviewer suggests comparing a c-PINN with 80 units per layer with an ml-PINN consisting of 4 neural networks with 40 units per layer. Please also ensure optimal hyperparameters (base learning rates for ADAM, ...) are used for both models.

Response: Thanks for the reviewer's suggestion. To ensure the same capacity of fully connected neural networks for both ml-PINN and c-PINN, the c-PINN is constructed by a single neural network (NN) with a structure of [1, 20, 20, 20, 1], while the ml-PINN consists of four NNs structured as [1, 10, 10, 10, 1] for each NN. The notation [1, 20, 20, 20, 1] means that the NN comprises an input layer with 1 node, 3 hidden layers with 20 nodes each, and an output layer with 1 node. As shown in Fig. A1, the comparison result of beam computation illustrates that ml-PINN outperforms c-PINN significantly under the same number of parameters.

The comparison of results in terms of c-PINN and ml-PINN has been revised and added in the revised manuscript, and the relevant content including Figures and Tables has been revised. For ease of reference, the corresponding text is presented below (Lines 446~464 in revised manuscript):

“The difference in prediction performance between the classical PINN (c-PINN) and the proposed ml-PINN is compared, as illustrated in Fig. A1. This figure shows the results for Case 1 of the 1D beam structure, as shown in Table 1 and Fig. 4b. It should be note that both c-PINN and ml-PINN employ fully connected neural networks with identical capacities, as illustrated in Fig. A1. It clearly shows that ml-PINN outperforms c-PINN in terms of prediction accuracy and computational efficiency. In addition, Table A1 provides detailed results for all four cases on the 1D beam structure using both c-PINN and ml-PINN (refer to Table 1 for cases details). As shown in Table A1, even after performing 10^4 iterations, c-PINN still exhibits significantly higher errors compared to ml-PINN in predicted displacements. Additionally, it is worth mentioning that ml-PINN achieves the superior result while requiring only a quarter of computation time compared to c-PINN under the same training parameters.”(Lines 445~454 in revised manuscript)



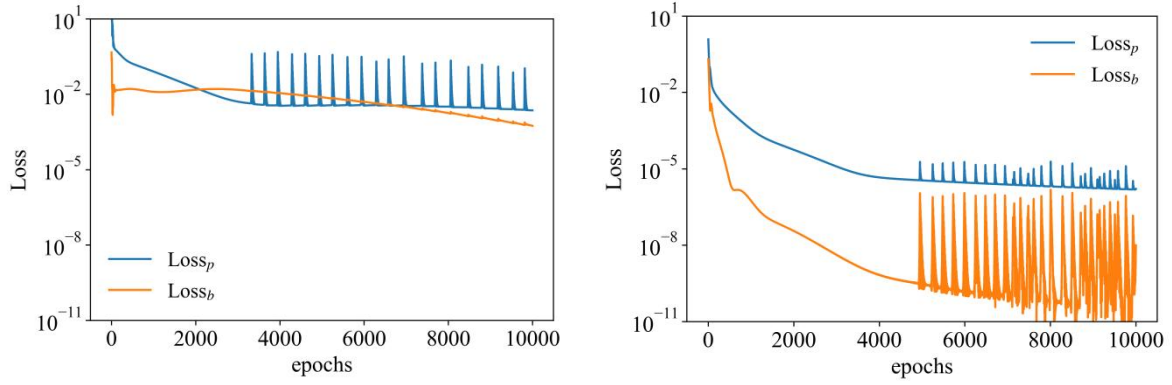


Fig. A1 | Comparison results of computation for 1D beam structure using classical PINN (c-PINN) and ml-PINN. **a** Predicted outcomes and loss decline curve of c-PINN exhibit poor prediction accuracy and convergence difficulties. $Loss_b$ and $Loss_p$ represent the loss functions from boundary constraints and PDEs, respectively. The notation [1, 20, 20, 20, 1] means that the neural network consists of an input layer with 1 node, 3 hidden layers with 20 nodes each, and an output layer with 1 node. **b** Predicted outcomes and loss decline curve of ml-PINN demonstrate rapid convergence of the loss function along with improved prediction accuracy. It should be note that the single network of [1, 20, 20, 20, 1] for c-PINN and four networks of [1, 10, 10, 10, 1] for ml-PINN possess the same capacity of fully connected neural networks.

Table. A1 Comparison results of computation for 1D beam structure obtained by classic PINN (c-PINN) and ml-PINN

Cases	Methods	Error (%)	Time (s/epoch)
Case 1	c-PINN	1.2~56.9	4.3×10^{-3}
	ml-PINN	0.6 ± 0.5	1.1×10^{-3}
Case 2	c-PINN	2.6~107.4	4.3×10^{-3}
	ml-PINN	2.4 ± 0.8	1.1×10^{-3}
Case 3	c-PINN	11.4~52.0	4.5×10^{-3}
	ml-PINN	1.1 ± 1.0	1.1×10^{-3}
Case 4	c-PINN	15.2~104.3	4.5×10^{-3}
	ml-PINN	0.3 ± 0.1	1.1×10^{-3}

Notes: The c-PINN is an abbreviation for classic PINN. $Loss_p$ and $Loss_b$ denote the loss function of the PDEs and the loss function of boundary constraints, respectively. $Error = \frac{\|\omega_{PINN} - \omega_{FEM}\|_2^2}{\|\omega_{FEM}\|_2^2}$ where ω_{PINN} and ω_{FEM} represent the value of the vertical displacement computed by ml-PINN and FEM respectively.

3. Follow-up comment #1.9 Methods

Since no hyperparameter tuning is reported that usually optimizes the architectures of the neural networks for the two considerably different problems of the c-PINN and the ml-PINN (different difficulty due to the number of outputs per network), it is not clear if the results of the c-PINN are the optimal solution for the given physical problem. Please ensure optimal hyperparameters through a hyperparameter tuning step.

Response: Thanks for the reviewer's reminder. We have added the hyperparameters of c-PINN in the revised manuscript and made revisions to the Model Setting section. For your convenience, the

corresponding text is presented below (Lines 416~424 in revised manuscript):

“All codes are optimized using the ADAM algorithm with a learning rate of 10^{-3} . A total of four neural networks (NNs) were established to predict the output of ω , θ , M and F_Q in the 1D beam structure, namely ω_{NN} , θ_{NN} , M_{NN} and F_{QNN} . All four fully connected feedforward NNs with ml-PINN consist of an input layer with 1 node, 3 hidden layers with 10 nodes each, and an output layer with 1 node. For c-PINN, a single fully connected feedforward NN was constructed with an input layer containing 1 node, 3 hidden layers comprising 20 nodes each, and an output layer with 1 node. The aforementioned hyperparameters setting is optimized to make c-PINN and ml-PINN posses the same capacity in terms of fully connected neural networks. Additionally and similarly, for the 2D shell structure with ml-PINN, three NNs were set up as ω_{NN} , $\kappa_{NN}(\kappa_x, \kappa_y, \text{ and } \kappa_{xy})$, $M_{NN}(M_x, M_y, \text{ and } M_{xy})$. The NN of ω_{NN} consists of an input layer with 2 nodes, 3 hidden layers with 30 nodes, and an output layer with 1 node. Additionally, the NNs used in κ_{NN} and M_{NN} were constructed with an input layer with 2 nodes, 3 hidden layers with 20 nodes, and an output layer with 3 nodes.”

Responses to Reviewer #2

"Response: The authors would like to thank the reviewer for your time and effort in reviewing our manuscript. We sincerely appreciate the reviewer's encouraging comments, which have greatly benefited us. All the suggestions and comments have been addressed point-by-point and the manuscript has been revised accordingly. Please find below our responses to each comment."

Thank you to the authors for their effort in revising the manuscript. I am much happier with the ability of the paper to present the advantages of proposed method. I would now recommend it be allowed to be published. The ml-PINN is able to achieve comparable results to FEM on simulated beam and shell models. It also outperforms a standard PINN (lower error, lower computation cost) for this task. The beam and shell examples used are well understood in engineering, with analytical solutions available. Although this makes them familiar to the reader, the results achieved are therefore of limited use on their own. This paper therefore mainly serves as a display of the method and its potential advantages, rather than the focus being on application.

"Response: Thanks for the reviewer's kind concern. Regarding the advantage of m-PINN (renamed as ml-PINN according to the previous reviewer's suggestion) over both standard PINN and FEM, The comparison of results in terms of computational accuracy and efficiency has been added in the revised manuscript (Lines 436~455 and 298~316). For ease of reference, the corresponding text is presented as follows:"

Evidence and discussion of the benefits the proposed ml-PINN over FEM and standard PINNs was the major element missing from the original paper and the added sections address this specifically. I believe that the manuscript is stronger with this inclusion.

"Response: We appreciate the reviewer's suggestions on the structure of the article. "Abstract / Main text / Results / Discussion / Methods / Data availability / Code availability / Competing interests" is the universal typesetting mode of Nature series journals. The purpose of putting Results in the front is to make it easier for the reader to get to the point firstly and directly. However, in this paper, the essential methodological information involving the framework description has been provided prior to interpreting results such that the readers can understand the general method. Further details on the method and the common knowledge are presented at the end of the manuscript."

If that is the universal typeset of the journal, then yes, I'm happy to remain with the current paper structure. With some of the rewording present in the revised manuscript, the proposed method is presented to the reader more clearly than before. This helps the reader interpret results shown.

"Additionally, according to the reviewer's suggestion, the clear conclusion highlighting the advantage of ml-PINN has been included in the Discussion section, as "It is evident that the proposed ml-PINN framework significantly enhances computational accuracy by reducing the error of beam and shell structures computation to 0.5% and 2%, respectively, surpassing that achieved by classical PINN, while simultaneously decreasing computational time." (Lines 291~294)"

Concisely stating the advantages of the method was a very welcome addition. This helps to motivate the work of the paper. I believe a similar statement could be used at the end of the abstract.

It currently reads:

"The verification results demonstrate that ml-PINN achieves comparable accuracy to FEM in computing beam and shell structures. The proposed method holds the potential to become a new paradigm for structural mechanics computation and facilitate the intelligent computation of digital twin systems."

This statement, especially "intelligent computation of digital twin systems" could mean any number of things and I believe a more specific statements such as "4 times less computation time than a standard PINN" would be better in place of this. In general, I would only make claims that are able to be backed up with evidence (results or citations).

Response: We sincerely appreciate the reviewer's encouraging comments. In response, we have made revisions to the Abstract that highlight ml-PINN's superior computational accuracy and efficiency over classic PINN, specifically as (Lines 27~29 in revised manuscript): "The proposed ml-PINN demonstrates a remarkable advancement over the classic PINN in terms of the accuracy and computation time. The proposed method holds the potential to become a new paradigm for structural mechanics computation."

REVIEWERS' COMMENTS:

Reviewer #1 (Remarks to the Author):

The authors have done a tremendous job addressing the reviewer's concerns. The reviewer believes that the revised manuscript can be accepted and is ready for production.

Reviewer #2 (Remarks to the Author):

Thank you to the authors for their effort in revising the manuscript. I would now recommend it be allowed to be published.

Though my decision was previously accept, the suggestions of the other reviewer appear to be very valuable and I agree with their inclusion.

Detailed Responses to Review Comments

COMMS-23-0444-B: Multi-level physics-informed deep learning
approach for computational structural mechanics

Weiwei He, Jinzhao Li, Xuan Kong, Lu Deng

July 19, 2024

REVIEWERS' COMMENTS:

Reviewer #1 (Remarks to the Author):

The authors have done a tremendous job addressing the reviewer's concerns. The reviewer believes that the revised manuscript can be accepted and is ready for production.

Reviewer #2 (Remarks to the Author):

Thank you to the authors for their effort in revising the manuscript. I would now recommend it be allowed to be published.

Though my decision was previously accept, the suggestions of the other reviewer appear to be very valuable and I agree with their inclusion.

Dear Editors and Reviewers,

Thank you for your positive comments and suggestions regarding our revised manuscript.

We have reviewed the editorial comments and requests in the Editorial Requests Table, and we checked and corrected the manuscript one by one. Additionally, according to the editor's suggestion, the title has been changed as "Multi-Level Physics-Informed Deep Learning for Solving Partial Differential Equations in Computational Structural Mechanics", and we provided a Supplementary Information (the original Appendix) in the new submitted version.

More details on the revision of the manuscript can be found in the Editorial Requests Table.

Yours sincerely,

Xuan Kong (Corresponding author)