Transfer learning for improved generalizability in causal physics-informed neural networks for beam simulations

Kapoor, Taniya; Wang, Hongrui; Núñez, Alfredo; Dollevoet, Rolf

DOI
10.1016/j.engappai.2024.108085

Publication date
2024

Document Version
Final published version

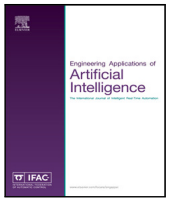
Published in
Engineering Applications of Artificial Intelligence

Engineering Applications of Artificial Intelligence 133 (2024) 108085

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Transfer learning for improved generalizability in causal physics-informed neural networks for beam simulations

Taniya Kapoor, Hongrui Wang *, Alfredo Núñez, Rolf Dollevoet

Department of Engineering Structures, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, 2628CN, The Netherlands

ARTICLE INFO

Dataset link: https://github.com/taniyakapoor/Causal-PINN-for-beam/tree/second

Keywords:
Transfer learning
Causality
Physics-informed neural networks (PINNs)
Biharmonic equations
Euler–Bernoulli beam
Timoshenko beam
Elastic foundation

ABSTRACT

This paper proposes a novel framework for simulating the dynamics of beams on elastic foundations. Specifically, partial differential equations modeling Euler–Bernoulli and Timoshenko beams on the Winkler foundation are simulated using a causal physics-informed neural network (PINN) coupled with transfer learning. Conventional PINNs encounter challenges in handling large space–time domains, even for problems with closed-form analytical solutions. A causality-respecting PINN loss function is employed to overcome this limitation, effectively capturing the underlying physics. However, it is observed that the causality-respecting PINN lacks generalizability. We propose using solutions to similar problems instead of training from scratch by employing transfer learning while adhering to causality to accelerate convergence and ensure accurate results across diverse scenarios. The primary contribution of this paper lies in introducing a causality-respecting PINN loss function in the context of structural engineering and coupling it with transfer learning to enhance the generalizability of PINNs in simulating the dynamics of beams on elastic foundations. Numerical experiments on the Euler–Bernoulli beam highlight the efficacy of the proposed approach for various initial conditions, including those with noise in the initial data. Furthermore, the potential of the proposed method is demonstrated for the Timoshenko beam in an extended spatial and temporal domain. Several comparisons suggest that the proposed method accurately captures the inherent dynamics, outperforming the state-of-the-art physics-informed methods under standard L^2-norm metric and accelerating convergence.

1. Introduction

Beams on elastic foundations (Fig. 1) are a fundamental and indispensable structural component in civil engineering, providing critical support and stability to different and diverse structures (Lamprea-Pineda et al., 2022; Deng et al., 2023; Tsudik, 2012; Kabir and Aghdam, 2019; Hetényi and Hetbenyi, 1946). Due to their characteristic to distribute loads, mitigate deformations, and enhance structural stability, these beams are extensively utilized in various structures, such as railway tracks (Lamprea-Pineda et al., 2022), pile foundations embedded in soils (Petrosian, 2022), and longitudinal fibers in a composite elastomer (Tsudik, 2012), among others. Understanding their dynamics is essential for ensuring the structural integrity of these systems, developing effective maintenance strategies, optimizing machine performance, refining design methodologies, and enabling precise control mechanisms. These issues highlight the need for advanced methodologies to simulate and predict the underlying dynamics of beams on elastic foundations, facilitating safer, more efficient, and reliable structures and systems.

However, accurately predicting the dynamics of beams on elastic foundations through experiments and measurements could be infeasible (Chang et al., 1999). Conducting many experiments with varying materials, conditions, and prototypes becomes impractical and prohibitively costly. In practice, finite element-based software provides a viable alternative for simulating such scenarios (Madenci and Guven, 2015). However, these software solutions are restricted in generalization. For instance, even a slight change in the problem domain requires performing the entire new simulation from scratch, including mesh creation and adjustments (Karniadakis et al., 2021). This non-generalization becomes particularly problematic when different aspects of the system need to be investigated separately or when multiple design iterations are required. The number of simulations necessary for tackling a design problem can quickly escalate into thousands, making the task laborious and time-consuming.

Recently, deep learning and neural networks, in particular, have been used extensively as surrogates to model the underlying physical

* Corresponding author.
E-mail addresses: t.kapoor@tudelft.nl (T. Kapoor), h.wang-8@tudelft.nl (H. Wang), A.A.NunezVicencio@tudelft.nl (A. Núñez), R.P.B.J.Dollevoet@tudelft.nl (R. Dollevoet).

https://doi.org/10.1016/j.engappai.2024.108085
Received 31 October 2023; Received in revised form 8 January 2024; Accepted 12 February 2024
Available online 18 February 2024

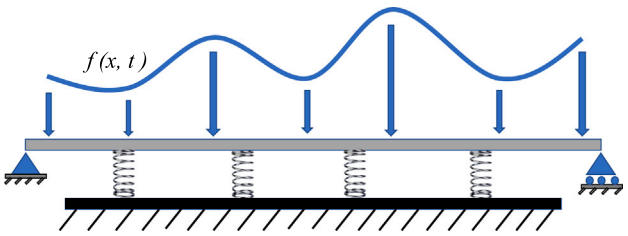


Fig. 1. Simply supported beam on an elastic foundation under varying transverse force.

phenomenon (Jumper et al., 2021; Carleo et al., 2019) , including applications to shape optimization, resulting in cost-efficient shapes (Erdaş et al., 2023; Aye et al., 2019; Yildiz et al., 2003). However, even state-of-the-art supervised machine-learning approaches encounter similar challenges as traditional experimental methods, requiring substantial input–output data at various fidelities to learn the underlying dynamics effectively. This large data requirement poses a significant hurdle, as obtaining such vast data can be arduous and resource-intensive (Karniadakis et al., 2021).

One potential approach to mitigate the need for an enormous amount of data is to incorporate the underlying physics into the learning procedure, thereby guiding the neural network based on physics principles as presented by Raissi et al. (2019), Karpatne et al. (2017b,a, 2022), Meng et al. (2023), among others. One popular class of methods that adopts this approach is physics-informed neural networks (PINNs) proposed by Raissi et al. (2019). PINNs are a form of semi-supervised learning where the boundary and initial conditions serve as input–output pairs while the solution is regularized by the governing partial differential equations (PDEs). However, several challenges for PINNs have emerged, including spectral bias (Wang et al., 2022), shock learning (Fuks and Tchelepi, 2020), generalization with even slight changes in physical parameters and computational domain (Kim et al., 2021; Kapoor et al., 2023b,a), and difficulties dealing with large coefficients (Krishnapriyan et al., 2021; Dekhovich et al., 2023; De Ryck et al., 2023).

Another such open problem for vanilla PINNs is handling extensive space–time domains, as discussed in Jagtap et al. (2020b), Lippe et al. (2023), Jagtap and Karniadakis (2021), Shukla et al. (2021), among others. This challenge can be attributed to the training process, as vanilla PINNs tend to prioritize training at a higher time level due to implicit gradient bias (Wang et al., 2022), leading to violation in temporal causality and inaccurate solutions, particularly for problems highly dependent on initial conditions.

Physical systems are known to possess an inherent causal structure. For instance, the deflection of the beam at any point in time is causally linked to the previous state of the system (deflection), the physical properties of the beam, and the external forces acting on it. This causality is a fundamental aspect of how the beam equations accurately model the behavior of beams in response to loads, making it a useful tool in engineering and physics. The PINN model could learn complicated solutions to PDEs when the causality is considered, enabling progressive sequential-time learning of the solution.

Our work proposes to train PINN while respecting causality (Wang et al., 2022) in the context of structural engineering, referred to as causal PINN hereafter. In particular, our principal aim is to resolve the training challenge and achieve precise predictions of beam dynamics in a large space–time domain. This challenge is overcome by proposing a modification in the training approach of PINNs, enforcing training at lower time levels before progressing to subsequent ones. Consequently, a weighted loss function is utilized, incorporating a causality parameter to preserve the physical causality inherent in beam dynamics. The causal PINN approach, validated through numerical experiments, demonstrates enhanced accuracy in prediction.

However, as we present in this work, even after employing causal PINN, the models lack generalizability to different initial conditions and computational domains, requiring each new problem to be solved from scratch. This limitation reverts the problem to the need for extensive simulations for each problem. To mitigate this issue, we propose employing transfer learning (TL) (Niu et al., 2020) in conjunction with causal PINN. The idea of transfer learning is to utilize the knowledge acquired from solving one problem in the form of trained model parameters to be utilized in a similar or related problem, accelerating the training process.

We examine the application of PINNs on well-known Euler–Bernoulli and Timoshenko beam models on elastic foundations, specifically the Winkler foundation (Lamprea-Pineda et al., 2022; Younesian et al., 2019). Through the numerical examples in our manuscript, we show that vanilla PINNs face challenges in approximating solutions for PDEs, for which even analytical solutions are available. In practice, only a handful of PDEs have analytical solutions that serve as prototypes for proof of concept to validate a proposed method. The inefficiency of vanilla PINNs in resolving solutions for such PDEs signifies its limited applicability in the real world, which we tackle in this paper by enforcing a causal training framework.

This paper proposes a novel approach to simulate beams on elastic foundations using the Euler–Bernoulli and Timoshenko theories, employing a transfer learning-based causal PINN framework to conduct comprehensive experiments. Specifically, transferring knowledge from one initial condition to another, handling noisy initial conditions, transferring knowledge for beams of different lengths, and systems with significant time dependencies are addressed. The primary contributions of this research paper are as follows:

A causality-respecting PINN loss function addresses the aforementioned limitations and effectively enforces the relevant physics. However, implementing this modified causal loss function requires a denser neural network with more parameters. Considering the importance of various factors in engineering structure design and the impracticality of simulating every instant, transfer learning is proposed within the causal PINN architecture. By incorporating transfer learning, the parameters of the previously trained model are leveraged to initialize and train new models. Consequently, this reduces the computational burden and enables faster convergence for subsequent tasks, improving the efficiency of simulating the dynamics of beams on elastic foundations.

The rest of the paper is structured as follows: Section 2 presents related works to this manuscript. Section 3 provides a detailed discussion of vanilla and causal PINN. Section 4 introduces the proposed framework of fusing transfer learning with causal PINN to train different models. Section 5 presents the numerical experiments results, showcasing the effectiveness of our methodology in addressing challenging beam problems where the vanilla and advanced PINN-based methods fail. Finally, the main findings are summarized, and conclusions drawn from this study are presented in Section 6.

2. Related works

This section outlines the pertinent studies conducted within the domain of transfer learning-driven PINNs, causal PINNs, and physics-informed methodologies for the simulation of beam models.

Applying transfer learning within PINNs has garnered significant attention (Goswami et al., 2020). Notably, Li et al. (2023) predicted laser deposition temperature fields accurately without labeled data, using physical losses and transfer learning. In another work, Liu et al. (2022) utilized transfer learning for accurate temperature field inversion with limited observations, employing a PINN and optimal position selection. Roy and Guha (2023) developed a multi-objective loss function and transfer learning for accurate elastoplastic solid mechanics solutions through PINN. In a different study, Haghighat et al. (2023) proposed a transfer learning-based PINN framework for efficient stress–strain constitutive modeling. While our research aligns with the

fundamental principle of leveraging transfer learning, a distinguishing aspect lies in our consideration of causality during the training of the models.

In the literature, research has been conducted to enforce causality in the PINN framework without incorporating transfer learning (Mu et al., 2023). In another work, Penwarden et al. (2023) proposed a causal framework incorporating transfer learning to simulate time-dependent PDEs. Although our work shares similar ideas of incorporating causality and utilizing transfer learning within the PINN framework, we employ transfer learning to train distinct models under diverse conditions. Conversely, Penwarden et al. (2023) employs transfer learning within a particular problem by segmenting the domain into multiple subdomains and leveraging insights from one subdomain to another, employing the concepts of domain decomposition and PINNs (Jagtap and Karniadakis, 2021).

Recently, beam simulations have concentrated on physics-informed methodologies, largely omitting the considerations of causality and transfer learning. Noteworthy works include Bazmara et al. (2023), which utilized PINNs for estimating nonlinear bending behavior within a confined domain. Similarly, Kapoor et al. (2023d,c) delved into applying PINNs for the system of beam models and moving load problems, albeit within the limited domain confines. Lee (2023) introduced a spatio-temporal PINN tailored for analyzing the dynamics of cantilever beams. In Xu et al. (2023), a self-adaptive PINN framework capable of accommodating varying load conditions is presented. Additionally, Fallah and Aghdam (2023) sought to enhance predictions by incorporating supplementary data, all still constrained within the confined domain bounded by the capabilities of PINNs. This work aims to enhance the potential of physics-informed methodologies for simulating beam dynamics.

3. Vanilla and causal PINN

This section is structured into two subsections. First, we provide an overview of the architecture of the vanilla PINN (Raissi et al., 2019). Second, a modification in the PINN loss function leading to the incorporation of causality in the PINN loss function, as proposed by Wang et al. (2022), is presented.

3.1. Vanilla PINN

Recently, PINNs have been widely used for solving PDEs across diverse domains, including but not limited to works by Chen et al. (2023), Zobeiry and Humfeld (2021), Shen et al. (2021). PINNs are based on deep neural network (DNN) architecture, and the idea of PINN is to incorporate physical knowledge in the loss function of DNN. The loss function consists of two terms - a data term and a physics term. The data term ensures that the neural network fits the provided data points, while the physics term enforces the PDE constraints. Here, the data term refers to the value of the quantity of interest at initial and boundary points. Minimizing the data term amounts to measuring the discrepancy between the predicted solution of the PINN and the measured data points. The physics term incorporates the PDE constraints into the loss function, evaluating the differential operator of the PDE using automatic differentiation (Paszke et al., 2017). The resulting equation is then included as a penalty term in the loss function. To elucidate these terms, we consider an abstract PDE as,

$$D(u(x,t,k)) = f(x,t), \quad (x,t) \in \mathcal{D} \times \mathcal{T} \tag{1}$$

where D is the differential operator, $\mathcal{D}$ is the spatial domain, and $\mathcal{T}$ is the temporal domain. The unknown solution is u depending on independent space (x) and time (t) variables. A constant parameter is k, and f is the source term. To ensure the uniqueness of the solution, appropriate initial and boundary conditions are necessary for the considered PDE.

$$\begin{aligned} u(x,0) &= g(x), \quad (x,0) \in \mathcal{D} \times \Gamma \\ u(x_b,t) &= \bar{g}(x_b,t), \quad (x_b,t) \in \Omega \times \mathcal{T} \end{aligned} \tag{2}$$

here, $g(x)$ and $\bar{g}(x_b,t)$ are the initial and boundary conditions, respectively. The initial temporal region and spatial boundary are Γ and Ω, respectively. The loss function of PINNs is defined as follows

$$L(\mu) = \lambda_1 L_{\text{PDE}}(\mu) + \lambda_2 L_{\text{IC}}(\mu) + \lambda_3 L_{\text{BC}}(\mu) \tag{3}$$

here, μ represents the trainable network parameters. The individual loss terms weighted by the hyperparameters λ_i, $i = 1,2,3$, are defined as,

$$L_{\text{PDE}}(\mu) = \frac{1}{N_{\text{int}}} \sum_{n=1}^{N_{\text{int}}} \| D(u^*(x^{(n)},t^{(n)},k)) - f(x^{(n)},t^{(n)}) \|^p \tag{4}$$

The loss terms for initial and boundary conditions in (3) are defined as follows,

$$\begin{aligned} L_{\text{IC}}(\mu) &= \frac{1}{N_i} \sum_{n=1}^{N_i} \| u^*(x^{(n)},0) - g(x^{(n)}) \|^p \\ L_{\text{BC}}(\mu) &= \frac{1}{N_b} \sum_{n=1}^{N_b} \| u^*(x_b^{(n)},t^{(n)}) - \bar{g}(x_b^{(n)},t^{(n)}) \|^p \end{aligned} \tag{5}$$

here, N is the total number of training points, which is the sum of interior training points (N_{int}), initial training points (N_i), and boundary training points N_b. The approximation of u by the neural network is denoted by u^*. Training with L^2-norm amounts to $p = 2$. The primary objective is minimizing (3) and obtaining optimal parameters (μ). These optimized parameters are then utilized for predicting the PDE solution $u(x,t), \forall (x,t) \in \mathcal{D} \times \mathcal{T}$.

3.2. Causal PINN

This subsection presents causal PINN, modifying the PINN loss function (Wang et al., 2022). The notion of causal PINNs is inspired by traditional numerical methods for solving differential equations that prioritize resolving the solution at lower times before approximating the solution at higher times. The modification in the loss function pertains to the PDE term $L_{\text{PDE}}(\mu)$, while the initial $L_{\text{IC}}(\mu)$ and boundary $L_{\text{BC}}(\mu)$ loss terms remain unchanged. The causal PDE loss term $L_{\text{PDE}}(\mu)$ is defined as

$$\begin{aligned} L_{\text{PDE}}(\mu) &= \sum_{i=1}^{N_t} w_i L_{\text{PDE}}(t_i,\mu) \\ w_1 = 1, \quad w_i &= e^{-\epsilon \sum_{k=1}^{i-1} L_{\text{PDE}}(t_k,\mu)}, \quad i = 2,3,\ldots,N_t \end{aligned} \tag{6}$$

Here, N_t is the number of timesteps in which the computational domain has been divided. The causality hyperparameter ϵ controls the steepness of the weights. The modification introduces a weighting factor, w_i, for loss at each time level t_i. The weight w_i depends on the accumulated PDE loss up to time t_i. The weights are adjusted to prioritize the fully resolved solution at lower time levels by exponentiating the negative of this accumulated loss. To summarize, the modified loss function ($L_{\text{PDE}}(\mu)$) for causal PINN could be written as

$$\frac{1}{N_t} \left[w_1 L_{\text{PDE}}(t_1,\mu) + \sum_{i=2}^{N_t} e^{-\epsilon \sum_{k=1}^{i-1} L_{\text{PDE}}(t_k,\mu)} L_{\text{PDE}}(t_i,\mu) \right] \tag{7}$$

From the above loss function, it is evident that for $L_{\text{PDE}}(\mu)$ to be minimized, the weights $w_1,\ldots,w_{N_t}$ should be large. However, the weights are defined in such a way that the minimization of $L_{\text{PDE}}(t_i,\mu)$ only starts if all residuals $L_{\text{PDE}}(t_j,\mu)$, for $1 \le j < i$ are minimized and vice versa. This modification of the loss function forces the neural network to train the model sequentially and first train the model at lower time levels. In other words, loss at time step t_i should only be minimized once losses at all previous time steps have been minimized. Hence, the causal PINN loss function prioritizes fully resolved solutions at lower time levels before approximating the solutions at higher time levels.

In the following section, the proposed transfer learning framework is presented along with the underlying motivation.

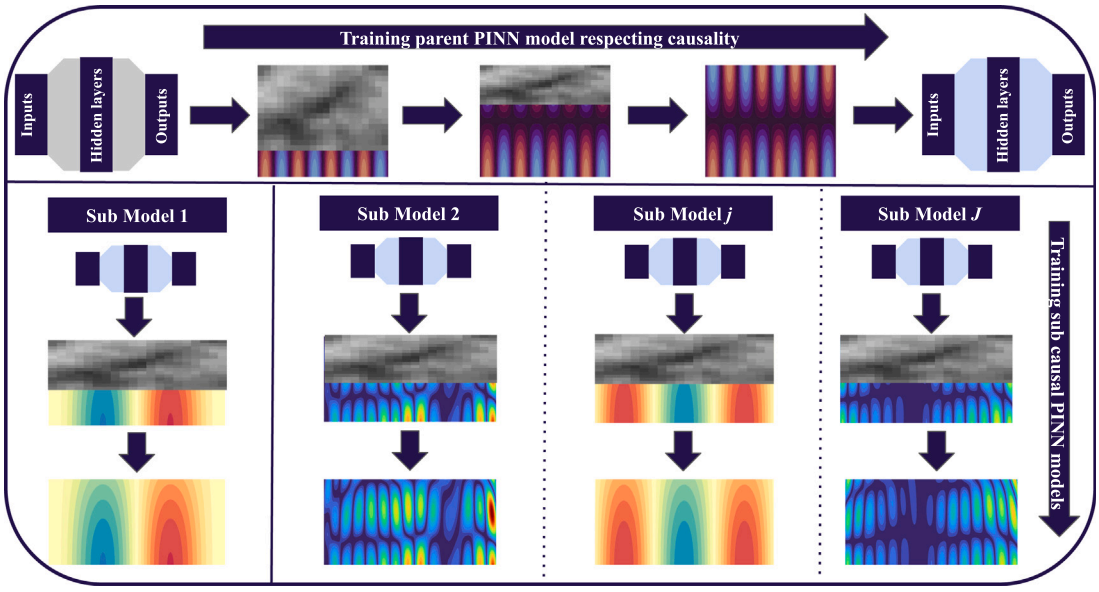


Fig. 2. Proposed transfer learning framework in causal PINN: The top horizontal block outlines the training process of the causal PINN for the parent model, which is the primary beam model under consideration. The model parameters, initialized using Xavier initialization within the first neural network, undergo training while adhering to causality. The resulting parameters of the trained model serve as the initialization for J subsequent tasks shown by the bottom vertical blocks ($1 \leq j \leq J$). These tasks pertain to different initial conditions and extensions of both spatial and temporal domains. The training of these subtasks is also performed to adhere to causality.

4. Transfer learning for causal PINN

Several factors are crucial for designing an engineering structure, and solving the problem for each case is important. However, training the neural network for every case is time-consuming and laborious. Here, we propose to utilize transfer learning for beam problems on the Winkler foundation. The idea is to train the parent beam model for one case, for instance, to train an Euler–Bernoulli beam for a specific initial condition and then utilize the parameters for different initial conditions. The aim is to reduce the training time for the transfer learning case compared to the case without transfer learning. This reduction in computational time in terms of epochs is done by utilizing the previously trained model parameters and using them as initialization for subsequent cases.

The proposed approach incorporates transfer learning for different scenarios for the same physical beam equation. Fig. 2 visually demonstrates the steps: initially, the parent model is trained using causal PINN for a significant number of epochs (n_1). Subsequently, the trained parameters are utilized as an initialization for the training of other problems of the physical equation with different initial conditions or for an extended domain, which is trained for a reduced number of epochs (n_2), where $n_2 \ll n_1$, reducing the computational cost of training the model again from the start. The step-by-step illustration is provided in Fig. 2.

In Fig. 2, the top horizontal block illustrates the training of causal PINN for the parent model, specifically the primary beam model, either the Euler–Bernoulli or the Timoshenko beam model. Xavier initialization (Glorot and Bengio, 2010) is utilized to train the parent beam model to address the vanishing or exploding gradient problem in neural networks. Proper weight initialization is crucial for stability and convergence. The parent model captures common features applicable to different subcases. The model parameters, generated using Xavier initialization for the initial neural network, undergo a training process adhering to the causal loss function. This training involves the resolution of solutions at lower times prior to approximating at higher times, as shown by the snapshots of the resolved solution. As the number of epochs increases, the model prediction at higher time levels improves only when the solution at lower time levels has been resolved up to a certain accuracy.

The resulting parameters from this training serve as the initialization for subsequent j tasks presented by the bottom vertical blocks in Fig. 2. Training subcases involve initializing parameters from the parent model, speeding up convergence and avoiding training from scratch. Reusing parameters reduces computational costs, making the process more efficient. Knowledge transfer from the parent model improves generalization, enabling submodels to adapt effectively to variations in conditions or domains. These subcases involve diverse initial conditions and extensions of both spatial and temporal domains. Notably, the training of these subtasks is also performed by minimizing the loss terms (7) and (5) in the loss function (3), ensuring a coherent and principled transfer learning framework.

Transferring good knowledge from the parent beam model would accelerate the convergence of subcases. However, dealing with highly complex subcases presents a challenge where improving or optimizing the network may not significantly enhance model accuracy. This situation is akin to the "Kolmogorov complexity" concept, which measures the length of the shortest computer program required to produce a specific output. While not considered in the current work, it is important to consider the Kolmogorov complexity of the parent task and subcases as discussed in Bolon-Canedo and Remeseiro (2020), Kabir and Garg (2023). In transfer learning, Kolmogorov complexity becomes pivotal as it captures the intricacy within a dataset or the solution of the PDE in our case. A highly complex solution containing intricate patterns and possible noise can pose challenges for even well-optimized neural networks in extracting meaningful features. This complexity is particularly pertinent in transfer learning, where pre-trained parent models may face challenges in transferring knowledge effectively to a target domain characterized by high intricacy. The diminished transferability of knowledge hampers anticipated improvements in model accuracy. To address this, reducing dataset complexity might be essential. The trained model can better focus on crucial patterns by processing the solution through feature reduction and noise elimination, fostering improved generalization and accuracy in the target task (Olivares et al., 2021; Whittaker et al., 2023).

The proposed framework addresses key structural engineering issues, contributing to design, optimization, and control methodologies. Causal PINNs prioritize lower time levels during training, enhancing the understanding of temporal structural behaviors, especially critical

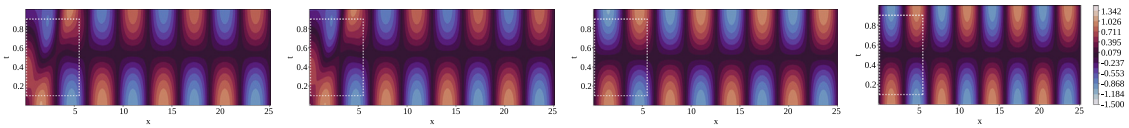


Fig. 3. Euler–Bernoulli beam displacement on the Winkler foundation **(a.)** Predicted solution using PINN **(b.)** Predicted solution using SA-PINN **(c.)** Predicted solution using causal PINN **(d.)** Reference solution.

for dynamic load responses and environmental changes. Incorporating transfer learning reduces the computational cost, aiding the application of the approach in real-world scenarios. Fusing temporal causality and transfer learning contributes to a larger design space exploration essential for a structural design problem. The proposed framework can be utilized to adapt the control strategies of structures based on knowledge gained from lower time levels (Faria et al., 2024; Arnold and King, 2021). This adaptability is valuable in developing control systems that can respond dynamically to changes in the structural environment, ensuring optimal performance and safety. The proposed method could also help structural health monitoring by updating the model as new data becomes available, enabling real-time monitoring and early detection of potential issues (Mai et al., 2023; Liu and Meidani, 2023).

In the next section, we perform a series of experiments to demonstrate the efficacy of the proposed framework.

5. Numerical experiments

This section presents the numerical experiments for simulating the dynamics of the Euler–Bernoulli and Timoshenko beam models using the proposed framework. The proposed framework is compared with five PINN-based methodologies, namely vanilla PINNs (Raissi et al., 2019), Self-adaptive PINNs (SA-PINN) (McClenny and Braga-Neto, 2023), gradient-enhanced PINN (gPINN) (Yu et al., 2022), PINNs with adaptive activation function (Adap. PINN) (Jagtap et al., 2020a), and Wavelet PINN (Wav. PINN) (Uddin et al., 2023). In addition, leveraging transfer learning, several other experiments are performed for noisy data, different initial conditions, and extrapolation in both spatial and temporal domains for the beam models.

The experimental setup involves first simulating the parent case and utilizing the trained parameters for various subcases. Specifically, transfer learning is employed for these subcases. The main model utilizes a neural network architecture comprising four hidden layers with 200 neurons each. These hyperparameters are the same for baseline PINNs and all other advanced PINN methods compared in this study, to have a fair comparison. The activation function employed is the hyperbolic tangent (tanh), and the limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) optimizer is utilized with a learning rate of 0.1. The parent model is trained for a total of 10,000 epochs. Within the causal-respecting PINN function, the causality hyperparameter (ϵ) is set to 5 and the number of timesteps N_t is taken to be 100. During the training process, $N_\mathrm{i} = 500$ initial points, $N_\mathrm{b} = 1000$ boundary points, and $N_\mathrm{int} = 10{,}000$ interior points are considered. The weight hyperparameters λ_1, λ_2 and λ_3 are taken to be 1 each. The selected evaluation metric is the L^2 relative error percentage ($\mathcal{R}$) defined as

$$\mathcal{R} = \frac{\|u^* - u\|_2}{\|u\|_2} \times 100 \tag{8}$$

where u^* is the approximated PDE solution by the neural network, and u refers to the ground truth. We utilize the trained parameters (μ) of the main model as initialization for training the subcase neural networks for only 1500 epochs, achieving the same level of accuracy as the main model.

Table 1
Euler–Bernoulli beam: $\mathcal{R}$ at $t = 1$ for $k = 1$.

	PINN	SA-PINN	gPINN	Adap. PINN	Wav. PINN	Causal PINN
$\mathcal{R}$	5.33	5.15	3.54	5.32	4.38	0.03

5.1. Euler–Bernoulli beam

The Euler–Bernoulli beam model is a mathematical framework used to analyze the behavior of beams when subjected to loads. It is derived from the three-dimensional elasticity theory or through principles such as Newton's second law or the generalized Hamiltonian Principle (Öchsner, 2021). The model assumes certain simplifications: it neglects the effects of rotary inertia and transverse shear deformations. The Euler–Bernoulli beam equation describes the behavior of a beam subjected to bending (Fig. 1). When the beam is supported on a Winkler foundation, representing an elastic foundation, the Euler–Bernoulli beam equation is modified to account for the interaction between the beam and the foundation. This modified equation considers the stiffness of the foundation and its influence on the behavior of the beam. The mathematical model of a simply supported Euler–Bernoulli beam on a Winkler foundation is described by Younesian et al. (2019)

$$u_{tt} + u_{xxxx} + p(x,t) = f(x,t), \quad x \in [0, 8\pi], t \in [0, 1] \tag{9}$$

where u represents the vertical displacement of the beam. u_{tt}, and u_{xxxx} represent the two times partial derivative of u with respect to t, and four times partial derivative with respect to x, respectively. The loading on the beam is defined by $f(x,t) = (2 - \pi^2)\sin(x)\cos(\pi t)$. The initial and boundary conditions are given as

$$\begin{aligned} u(x,0) = \sin(x), \quad u_t(x,0) = 0 \\ u(0,t) = u(8\pi,t) = u_{xx}(0,t) = u_{xx}(8\pi,t) = 0 \end{aligned} \tag{10}$$

The foundation reaction, $p(x,t)$, assumes that the reaction at every location is proportional to the displacement at a particular location, and the springs are linear and independent, as described in (9). The reaction force of the foundation is given by $p(x,t) = ku(x,t)$, where $u(x,t)$ is vertical displacement and k is the stiffness of linear springs. The exact solution for this problem is given by $u(x,t) = \sin(x)\cos(\pi t)$.

Solving (9), one can determine the vertical displacement of the beam at any point along its length and other important quantities of interest, such as bending moments and beam acceleration. These quantities help engineers understand how the beam will perform structurally and ensure it meets the desired design criteria. By calculating the displacement, engineers can check whether the beam deflects within acceptable limits under the applied loads.

We simulate (9) with five different methods to establish that incorporating causality provides more accuracy in the predicted solution than vanilla PINN, SA-PINN, adaptive activation PINN, wavelet PINN and gPINN for beam dynamics. The results presented in Table 1 indicate that vanilla PINN, SA-PINN, adaptive activation PINN, wavelet PINN and gPINN provide less accurate displacement predictions at $t = 1$ for the Euler–Bernoulli equation for stiffness $k = 1$. In contrast, causal PINN yields more accurate displacement predictions as the relative percent error is 0.03. This observation is further supported by the findings depicted in Fig. 3(a), (b), which demonstrates that PINN and

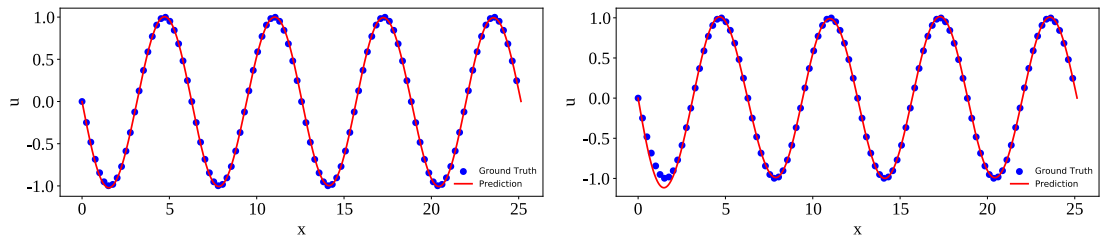


Fig. 4. Euler–Bernoulli beam equation on the Winkler foundation with noise in the initial condition of displacement of the beam **Left:** Predicted solution at final time ($t = 1$) with 10% Gaussian noise; **Right:** Predicted solution at final time ($t = 1$) with 20% Gaussian noise.

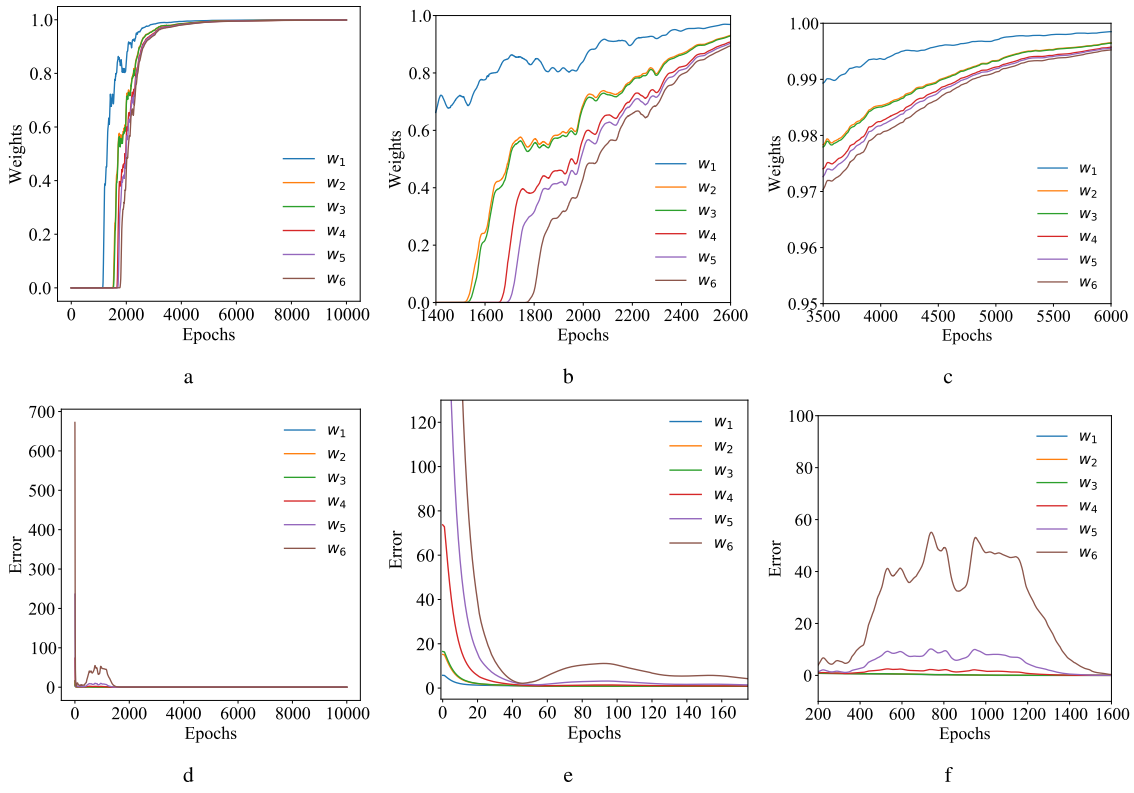


Fig. 5. Variation of weights (w_i) and the corresponding loss at that time step ($L_{\text{PDE}}(t_i, \mu)$) over training epochs. (a:) Magnitude of six random weights ($w_1, \ldots, w_6$) varying with epochs, where $w_1 < w_2 \ldots < w_6$. (b–c:) Zoomed-in segments of (a). (d) Relative percent error at six distinct time levels ($L_{\text{PDE}}(t_i, \mu)$) corresponding to weights ($w_1, \ldots, w_6$) varying with epochs. (e–f:) Zoomed-in segments of (d).

SA-PINN models are not accurate, particularly during the initial time, highlighted by the white rectangular box in Fig. 3. However, this challenge is effectively overcome by incorporating a causality-respecting loss function (Fig. 3(c)), which facilitates training the solution at lower time levels before training at higher times. Additional experiments concerning comparison of the proposed framework with a combination of PINNs and recurrent neural architectures for the Euler–Bernoulli beam equation on the Winkler foundation are presented in Appendix A. Furthermore, Additional plots for adaptive activation PINN, gPINN and wavelet PINN are presented in Appendix B.

Moreover, we empirically analyze the correlation between weights (w_i) and the corresponding loss at that time step ($L_{\text{PDE}}(t_i, \mu)$) for causal PINNs. We visualize the evolution of the magnitude of weights and errors over epochs for six random weights to understand how their magnitudes impact model training at that time level. The goal is to observe how the error decreases as the magnitudes of the weights increase over epochs, as shown in Fig. 5. Six random weights ($w_1, \ldots, w_6$) are considered at time steps $0.06, 0.29, 0.30, 0.41, 0.45$, and 0.53. From Fig. 5(a), it is evident that, as training progresses, the magnitude of weights increases and approaches a value of 1. Figs. 5(b) and 5(c) provide zoomed-in segments of Fig. 5(a), revealing a sequential convergence pattern at each time level: initially, w_1 converges, followed by w_2, and so forth. After 6000 epochs, all weights nearly converge to 1.

Fig. 5(d) illustrates the relationship between error and epochs, demonstrating a consistent decrease in error with increasing epochs. Figs. 5(e) and 5(f) are zoomed-in segments of Fig. 5(d), showing a sequential reduction in error corresponding to an increase in the magnitude of weight. It is evident that as the weights sequentially increase at each time level, the error also decreases sequentially. The observed pattern suggests that weights first converge at lower time levels before progressing to subsequent levels, gradually improving accuracy.

The parameters from compared PINN-based methods are not used subsequently to avoid incomplete or bad knowledge transfer. Only the trained parameters from the causal PINN formulation are transferred to the subsequent experiments presented in the next two subsections, fostering convergence by effectively reducing the training epochs.

5.1.1. *Noisy initial conditions*

This subsection presents the performance of the proposed method with noisy initial conditions. Initial conditions may not be perfectly

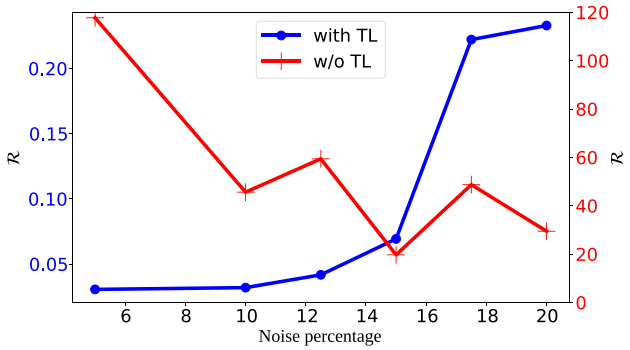


Fig. 6. $\mathcal{R}$ vs noise percentage in the initial condition for the Euler–Bernoulli beam for both approaches — with and without transfer learning.

Table 2
Euler–Bernoulli Beam: $\mathcal{R}$ at $t = 1$ for different percentages of noise in the initial condition. "TL" refers to transfer learning, and "w/o" refers to without. Abbreviations are used consistently for the following tables.

	5%	10%	12.5%	15%	17.5%	20%
with TL	0.03063	0.03198	0.04180	0.06937	0.222182	0.23296
w/o TL	117.7389	45.65849	59.42882	19.7473	48.75515	29.50691

known in real-world scenarios or contain uncertainties or noise. By learning displacements for noisy initial conditions, we can develop models that accurately represent the system's behavior under such realistic conditions, allowing us to account for uncertainties and better understand the actual response of the system. To observe the dynamics of beam models under these conditions, we introduce Gaussian noise in the initial condition ranging from 5% to 20%. The hyperparameter selection is the same as the main model, except for the number of epochs. With transfer learning, we perform 1500 epochs instead of 10000.

Table 2 presents the results from 5% to 20% Gaussian noise levels in the initial conditions for the displacement of the beam with and without (w/o) using transfer learning. The proposed method predicts u with less relative error percent. This prediction is significantly more accurate compared to the case without transfer learning. Also, Fig. 4 shows the results for 10% and 20% noise levels in the initial conditions for the displacement of the beam using transfer learning, demonstrating the computational efficiency of the proposed method.

Fig. 6 illustrates the comparison of relative error percentages concerning the noise percentage for both methods, one with transfer learning and the other without it. In the transfer learning scenario, it becomes apparent that an increase in the noise percentage results in a corresponding increase in the relative error percentage. When the subcases use the trained parameters for initialization, noise and error percentages exhibit a direct proportional relationship. However, in cases where trained parameters are not utilized, no discernible pattern emerges due to the non-convergence in minimizing the loss function.

5.1.2. Different initial displacements and velocities

In this section, we present the results of the Euler–Bernoulli beam for different initial conditions characterized by the change in initial displacements and velocities of the beam. Learning deflections for different initial conditions and force functions allows for generalization. Beams or structures can have varying initial conditions, such as different magnitudes, positions, or load distributions. By learning the deflections for a diverse set of initial conditions, we can develop models that capture the underlying patterns and behavior of the system, enabling accurate predictions for unseen or novel initial conditions.

Here, we consider different initial conditions compared to the parent model. The initial conditions for this case are $u(x,0) = a\sin(x)$ and $u_t(x, t = 0) = a\sin(x)$. The analytical solution for the corresponding

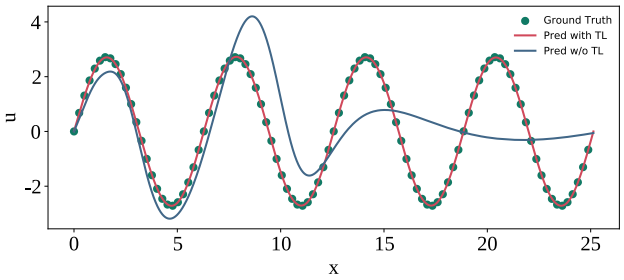


Fig. 7. Euler–Bernoulli beam on the Winkler foundation for initial velocity for case 1: Causal PINN prediction at final time $t = 1$ with and without transfer learning.

Table 3
Euler–Bernoulli beam: $\mathcal{R}$ at $t = 1$ for different velocities.

u^*	$\mathcal{R}$ (case 1)	$\mathcal{R}$ (case 2)
with TL	0.00105	0.02188
w/o TL	70.72229	193.85024

problem is $u(x,t) = a\sin(x)e^t$. We utilize the trained parameters of the Euler–Bernoulli beam model as an initialization for training this problem with different initial conditions considering $a = 1, 2$ (representing case 1 and case 2 in Table 3). The hyperparameters remain unchanged; the only change is the number of epochs, which is only 3000. Relative error percentages of displacement are presented in Table 3, which shows a large difference in relative percent errors. From Fig. 7, it is evident for the first case that the transfer learning approach achieves accurate predictions in fewer epochs.

5.2. Timoshenko beam

The Timoshenko beam theory considers the shear deformation and rotational effects neglected in the Euler–Bernoulli beam equation (Öchsner, 2021). Hence, in addition to the quantity vertical displacement (u), Timoshenko's theory considers the cross-sectional rotation (θ) as another unknown variable. The mathematical model for a beam resting on a Winkler foundation and subjected to an external load based on the Timoshenko beam theory is given as follows (Younesian et al., 2019)

$$\begin{aligned} &\theta_{tt} - \theta_{xx} + (\theta - u_x) = 0; \\ &u_{tt} + (\theta - u_x)_x + ku = h(x,t) \end{aligned} \tag{11}$$

where the symbols have their usual meaning, as in the case of the Euler–Bernoulli beam model. We consider $h(x,t) = \cos(t)$ and the computational domain to be $x \in [0, 3\pi]$ and $t \in [0, 1]$. The supporting initial and boundary conditions are given as

$$\begin{aligned} &\theta(x,0) = \frac{3\pi}{2}\cos(x) + \left(x - \frac{3\pi}{2}\right), \quad \theta_t(x,0) = 0 \\ &u(x,0) = \frac{3\pi}{2}\sin(x), \quad u_t(x,0) = 0 \\ &\theta(0,t) = \theta(3\pi,t) = u(0,t) = u(3\pi,t) = 0 \end{aligned} \tag{12}$$

The analytic solution for the rotation and vertical displacement is given as follows

$$\begin{aligned} &\theta(x,t) = \left(\frac{3\pi}{2}\cos(x) + \left(x - \frac{3\pi}{2}\right)\right)\cos(t) \\ &u(x,t) = \frac{3\pi}{2}\sin(x)\cos(t) \end{aligned} \tag{13}$$

Solving the Timoshenko beam model (11)–(12) would help engineers obtain more accurate predictions of beam deflections and rotations, especially for beams with high aspect ratios or subjected to high shear forces. This accuracy is crucial for assessing structural integrity, ensuring compliance with design criteria, and preventing potential failures.

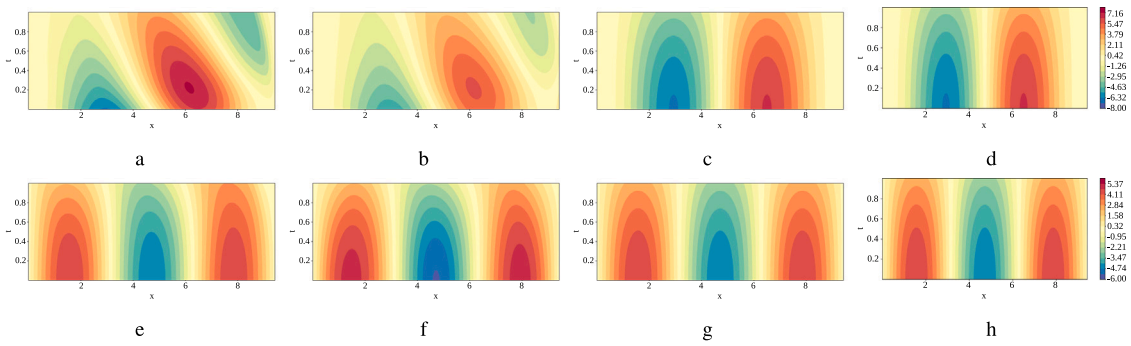


Fig. 8. Timoshenko beam on the Winkler foundation **Top:** Predicted Displacement (u^*) **(a.)** Using PINN **(b.)** Using SA-PINN **(c.)** Using causal PINN **(d.)** Reference solution **Bottom:** Predicted Rotation (θ^*) **(e.)** Using PINN **(f.)** Using SA-PINN **(g.)** Using causal PINN **(h.)** Reference solution.

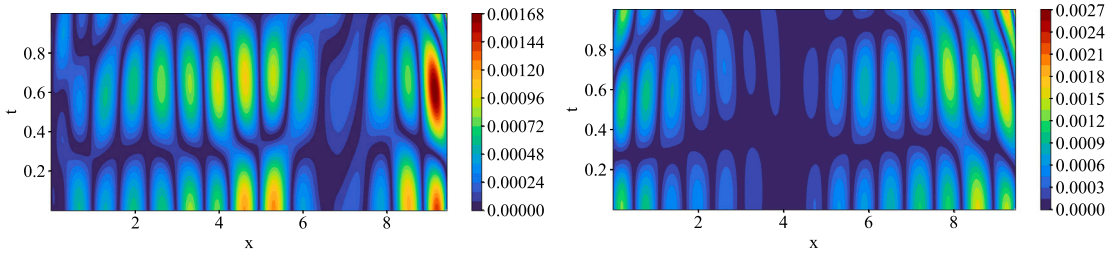


Fig. 9. Timoshenko beam on the Winkler foundation at final time $t = 1$. **Left:** Absolute error in predicting displacement ($|u - u^*|$); **Right:** Absolute error in predicting rotation ($|\theta - \theta^*|$).

Table 4
Timoshenko beam: $\mathcal{R}$ at $t = 1$ for $k = 1$.

	PINN	SA-PINN	gPINN	Adap. PINN	Wav. PINN	Causal PINN
u^*	119.17	137.15	119.17	240.78	238.46	1.2e−6
θ^*	9.18	6.56	38.63	9.16	9.10	7.7e−6

Fig. 8 illustrates the predicted displacement and rotation throughout the entire space–time domain. Figs. 8(c) and 8(g) depict the displacement and rotation prediction using the causal PINN loss function. Figs. 8(a–b) and 8(e–f) depict the displacement and rotation prediction using vanilla PINN and SA-PINNs, respectively, illustrating its failure in prediction. Additional plots for adaptive activation PINN, gPINN and wavelet PINN are presented in Appendix B. Furthermore, Fig. 9 presents the absolute error in displacement and rotation resulting from the causal PINN loss function. The maximum error magnitude falls below 10^{-2}, clearly indicating the accuracy of causal PINN.

Table 4 presents the relative percentage errors in predicting displacement and rotation for vanilla PINN, SA-PINN, gPINN, adaptive PINN, Wavelet PINN and causal PINN. In the case of causal PINN, both quantities of interest, u, and θ exhibit errors in the magnitude of 10^{-6}, demonstrating its accuracy. Conversely, the other five state-of-the-art PINN-based methods fail to adequately approximate the quantities of interest, as evidenced by a relative error percent of over 100% for displacement and an error of approximately 9% for rotation. The results show that Causal PINN accurately predicts displacement and rotation for the Timoshenko beam model.

5.3. Large space–time horizon

In the following two experiments, we show the potential of transfer learning and predict the displacement and cross-sectional rotation in a larger domain. We utilize transfer learning for extrapolating. There are several benefits to knowing deflections on larger domains. Firstly, it provides a better understanding of the structural behavior of the beam under different loading conditions. By analyzing the deflection over larger lengths, engineers can assess the beam's overall stability and structural integrity, which is crucial for designing safe and reliable structures.

Secondly, calculating the deflection for extended domains allows for more accurate predictions of the behavior of the beam in real-world scenarios. This information is valuable in various engineering applications such as building design, bridge construction, and aerospace engineering, where accurate deflection predictions are essential for ensuring the structural performance and safety of the final product.

Also, studying the deflection of the beam over a larger domain can help identify potential areas of weakness or excessive deformation. This knowledge enables engineers to make informed decisions about reinforcing certain sections or implementing design modifications to improve the overall performance and durability of the structure.

Furthermore, studying larger domains can optimize material usage and cost-effectiveness in construction projects. By accurately predicting deflection, engineers can optimize the size, shape, and materials used to construct beams, leading to more efficient designs and reduced material waste.

5.3.1. Extended spatial domain

In this section, we consider the Timoshenko beam model in an extended domain in space. The spatial domain for the parent model is $x \in [0, 3\pi]$. Here, we utilize the parameters of the parent model and train the subsequent models for different spatial domains, in particular $x \in [0, 5\pi]$, $x \in [0, 6\pi]$, and $x \in [0, 7\pi]$. The aim is to observe the method's potential in a larger domain, indicating that the method generalizes well. The results obtained with and without transfer learning are presented in Table 5, highlighting the superior accuracy achieved by the proposed method when utilizing parameters from the main model compared to training the model with Xavier initialization. Fig. 10 top row presents the proposed method's predictions of displacement and rotation, indicating that the model generalizes well across the spatial domain, inheriting the underlying structure and symmetry of the solution.

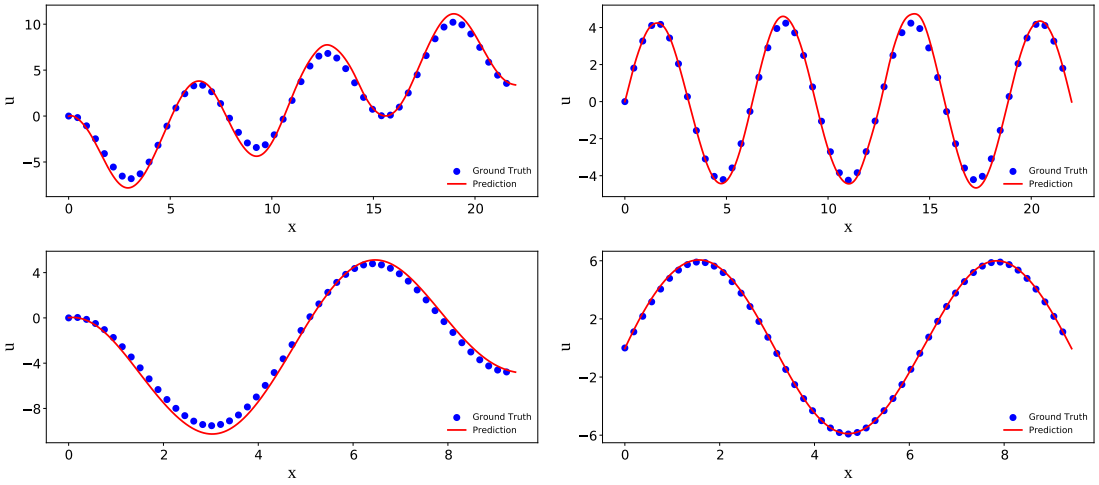


Fig. 10. Timoshenko beam on the Winkler foundation: **Top** Prediction for extended domain in space, $x \in [0, 7\pi]$ **Left:** Displacement (u^*); **Right:** Rotation (θ^*). **Bottom** Prediction for extended domain in time for $t = 7$ **Left:** Displacement (u^*); **Right:** Rotation (θ^*).

Table 5
Timoshenko beam: $\mathcal{R}$ for extension in the spatial domain with 3000 epochs.

x	With TL		w/o TL	
	u^*	θ^*	u^*	θ^*
$[0, 5\pi]$	6.6×10^{-5}	0.00011	2.34306	3.51362
$[0, 6\pi]$	0.00653	0.00097	21.81964	30.67853
$[0, 7\pi]$	1.52043	0.61573	11.00256	8.90537

Table 6
Timoshenko beam: $\mathcal{R}$ for extension in the temporal domain with 3000 epochs.

t	With TL		w/o TL	
	u^*	θ^*	u^*	θ^*
$[0, 4]$	9.7e−6	2.4e−5	7.9e−5	0.00026
$[0, 6]$	0.00111	0.00085	0.01627	0.12266
$[0, 7]$	0.89122	0.05554	4.92954	2.50340

5.3.2. *Extended temporal domain*

We now extend our investigation to the temporal domain based on successfully generalizing the proposed method in the spatial domain. By employing the trained parameters obtained from the parent model, we train the same model with an extension in time, considering different temporal domains, $t \in [0, 4]$, $t \in [0, 6]$, and $t \in [0, 7]$. The relative error percentage for all cases of the extended temporal domains is presented in Table 6. We observe that the proposed method accurately predicts displacement and rotation, while the approach without transfer learning fails to provide the same level of accuracy. Fig. 10 bottom row shows the predictions obtained by the proposed method for displacement and rotation in an extended temporal domain. The results show that utilizing transfer learning for extended domains in space and time provides accurate results, conserving the structure and symmetry of the solution.

6. Conclusions

This paper introduced a methodology for simulating the dynamics of beam models based on Euler–Bernoulli and Timoshenko's theories on the Winkler foundation. By incorporating transfer learning within a causality-respecting PINN framework, we addressed the need for re-training the network when there are modifications to the initial conditions or computational domain.

Numerical experiments demonstrated the effectiveness of the proposed approach. For the Euler–Bernoulli beam, we utilized the trained parameters from the parent model to simulate sub-cases with different initial conditions, including noisy ones. For the Timoshenko beam, we investigated its behavior in an extended spatial and temporal domain. These experiments showcased the generalization potential of the proposed method.

We also performed comparisons of the proposed method with five vanilla and advanced PINN-based methods. Results show that the causality-respecting PINN with transfer learning reduces computational costs and improves convergence. The results indicate that the method struggled to approximate the solutions accurately without transfer learning.

Overall, our findings highlight the efficacy of the proposed methodology in simulating beam dynamics under diverse engineering scenarios. By leveraging transfer learning and a causality-respecting PINN framework, we can reduce training requirements while achieving accurate results for various cases. This research opens up new possibilities for efficiently predicting the dynamics of structural elements, leading to advancements in structural engineering design, optimization, and control.

Future researchers should consider specific nuances to successfully apply the proposed framework in engineering domains. The performance of the proposed framework can be sensitive to hyperparameters, including the choice of causal parameter. Finding the optimal values may require empirical hyperparameter optimization, which is generally required for deep learning methods. Transferring knowledge from one engineering system to another requires understanding the domain characteristics and aligning them appropriately. Applicability of the proposed methodology in real-world engineering problems necessitates validation in complex environments, emphasizing interdisciplinary knowledge. Additionally, the choice of transfer learning method would depend on the real-world engineering challenge being solved.

Future research directions involve extending the methodology to other structural elements like systems of beams, strings and plates. An alternative research trajectory may involve training a family of PDE models and applying meta-learning techniques to derive a universal set of parameters applicable across diverse models. This unified parameter set could potentially be employed to test novel models, contributing to a generalized and efficient approach in the field. The codes will be made available upon publication.

Table A.1
Euler–Bernoulli beam: $\mathcal{R}$ at $t = 1$ for $k = 1$.

	PINN-RNN	PINN-LSTM	PINN-GRU
$\mathcal{R}$	5.77769	5.7789	5.8257

CRediT authorship contribution statement

Taniya Kapoor: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Hongrui Wang:** Writing – review & editing, Supervision, Project administration. **Alfredo Núñez:** Writing – review & editing, Supervision, Project administration. **Rolf Dollevoet:** Resources, Funding acquisition, Supervision, Project administration.

Declaration of competing interest

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

Data availability

The complete source code and data used to generate the results presented in this paper can be found at https://github.com/taniyakapoor/Causal-PINN-for-beam/tree/second.

Acknowledgment

This research was partly supported by ProRail and Europe's Rail Flagship Project IAM4RAIL - Holistic and Integrated Asset Management for Europe's RAIL System (grant agreement 101101966). The authors would also like to acknowledge the support and computational resources provided by the DelftBlue Cluster contributing to this research.

Appendix A. Training PINNs followed by recurrent architectures

This appendix presents a comparison of the proposed framework with a combination of PINNs and recurrent architectures for the Euler–Bernoulli beam equation on the Winkler foundation. The application of PINNs to solve PDEs has conventionally utilized deep neural network (DNN) architectures, as opposed to recurrent neural networks (RNN) or their gated variants such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (GRU) (Cho et al., 2014). In a PINN framework employing deep neural networks, the input comprises spatial and temporal variables, with the output representing the quantity of interest. For example, in the Euler–Bernoulli beam problem, the input variables are x and t, and the output variable is u.

Translating this problem into a recurrent network architecture would necessitate reformulating it as a sequential problem. The coordinate at each time step t^n (x_j, t^n) would become the input to a recurrent architecture (RNN, LSTM, or GRU) cell, with the beam deflection at that time step (u_j^n) serving as the output. However, transitioning from a DNN to recurrent structures raises three significant issues.

First, in our causal PINN-based proposed framework, the solution is learned sequentially, ensuring that the solution at a later time (e.g., u_j^2) is only learned after the solution at an earlier time (e.g., u_j^1) has been learned to a desirable accuracy. In recurrent architectures, the solutions at all time levels $(u_j^1, \ldots, u_j^n)$ are trained simultaneously, contrary to numerical methods where solutions at lower time levels are resolved before higher time levels.

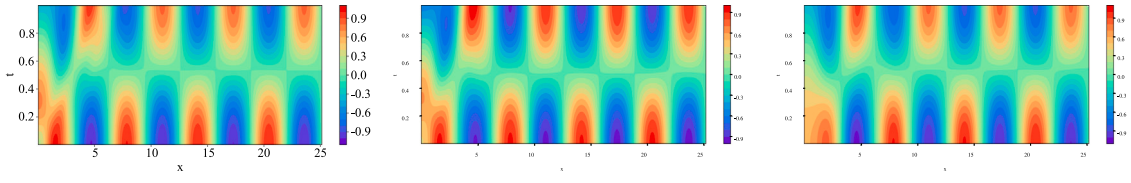
Second, in an RNN structure, due to the transfer of hidden state information from one cell to the next, u_j^2 depends on x_j and t^1, which does not align with the physical problem. The deflection of a beam acting under a force at a space–time location should depend only on the current space–time location, not any previous locations.

Third, RNN structures process data time-step-wise, making it non-trivial to incorporate boundary conditions intuitively. This difficulty is exacerbated when additional conditions, such as Neumann boundary conditions, need to be integrated, as in our case. Consequently, most recurrent physics-informed architectures primarily focus on solving ordinary differential equations (ODEs) rather than PDEs (Viana et al., 2021; Zhai et al., 2023; Lai et al., 2021), where enforcing boundary conditions within the loss function is relatively straightforward. Some works also deal with minimizing the data loss along with physical loss, taking the data prior from numerical methods (Liu et al., 2023; Bolandi et al., 2023).

However, as proposed in Kapoor et al. (2023b) and Michałowska et al. (2023), a hybrid approach combining neural PDE solvers with a recurrent architecture could be utilized to use RNN structures along with PINNs. A PINN is initially trained to incorporate boundary conditions in such an approach (Kapoor et al., 2023b). Subsequently, the trained model is processed through a recurrent architecture, leveraging the synergy of physics and data for efficient model training.

Following, we conduct three comparisons of the proposed causal PINN framework, with the combination of vanilla-PINN with recurrent neural architectures (RNN, LSTM, and GRU) for the Euler–Bernoulli beam on the Winkler foundation. We refer to the combination of PINN with RNN, LSTM or GRU as PINN-RNN, PINN-LSTM, and PINN-GRU, respectively. The PINN model is trained as discussed in Section 3 and is tested on a grid of 256 × 200 in space–time. The tested data is processed through the recurrent neural architecture, having an input and output size of 256, a single hidden layer of size 32, and a sequence length of 200, representing the time steps. The ADAM optimizer with a learning rate 0.01 is used across all models, executing 20,000 epochs. The trained recurrent models are finally tested on the same spatial locations for a time step size of 0.001, implying testing on 1000 time steps.

As presented in Table A.1, the results indicate that our proposed causality-enforced PINN method outperforms PINN-RNN, PINN-LSTM, and PINN-GRU. Consequently, the proposed framework empirically provides a better framework to preserve causality and mitigate the challenge of large space–time domains in vanilla PINN, at least for our problem.



a b c

Fig. B.1. Euler–Bernoulli beam displacement on the Winkler foundation **(a.)** Predicted solution using adaptive activation function (Adap. activation) **(b.)** Predicted solution using gradient enhanced PINN (gPINN) **(c.)** Predicted solution using wavelet PINN (Wav. PINN).

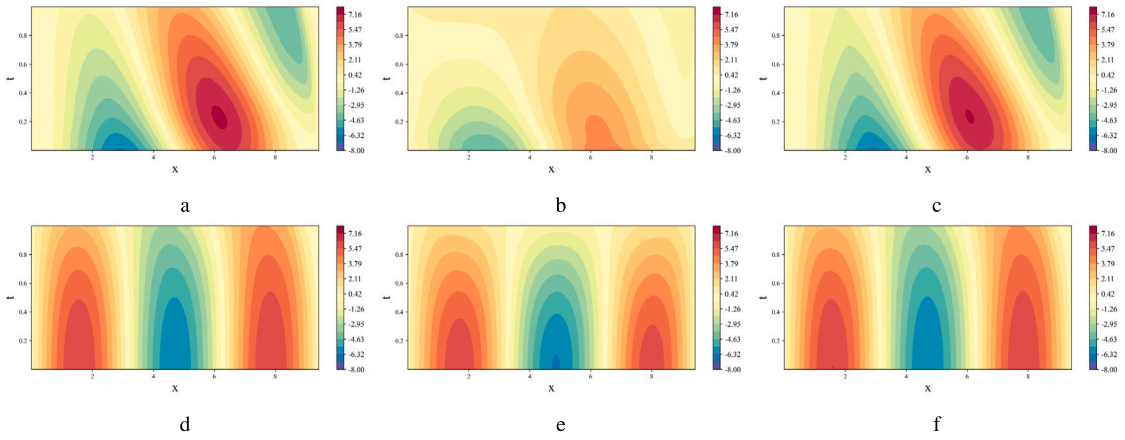


Fig. B.2. Timoshenko beam on the Winkler foundation **Top:** Predicted Displacement (u^*) **(a.)** Using adaptive activation PINN (Adap. PINN) **(b.)** Using gradient enhanced PINN (gPINN) **(c.)** Using Wavelet PINN (Wav. PINN) **Bottom:** Predicted Rotation (θ^*) **(d.)** Using adaptive activation PINN (Adap. PINN) **(e.)** Using gradient enhanced PINN (gPINN) **(f.)** Using Wavelet PINN (Wav. PINN).

Appendix B. Additional plots for the PINN-based compared methods

This appendix presents additional plots for the compared methods (adaptive activation PINN, gPINN, and wavelet PINN) with our proposed method. Fig. B.1 presents the results for these three methods for the Euler–Bernoulli beam equation. Fig. B.2 presents the results for these three methods for the Timoshenko beam equation.

References

Arnold, F., King, R., 2021. State–space modeling for control based on physics-informed neural networks. Eng. Appl. Artif. Intell. 101, 104195.

Aye, C.M., Pholdee, N., Yildiz, A.R., Bureerat, S., Sait, S.M., 2019. Multi-surrogate-assisted metaheuristics for crashworthiness optimisation. Int. J. Veh. Des. 80, 223–240.

Bazmara, M., Silani, M., Mianroodi, M., et al., 2023. Physics-informed neural networks for nonlinear bending of 3D functionally graded beam. In: Structures. Elsevier, pp. 152–162.

Bolandi, H., Sreekumar, G., Li, X., Lajnef, N., Boddeti, V.N., 2023. Physics informed neural network for dynamic stress prediction. Appl. Intell. 53, 26313–26328.

Bolon-Canedo, V., Remeseiro, B., 2020. Feature selection in image analysis: a survey. Artif. Intell. Rev. 53, 2905–2931.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., Zdeborová, L., 2019. Machine learning and the physical sciences. Rev. Modern Phys. 91, 045002.

Chang, P.B., Williams, B.J., Santner, T.J., Notz, W.I., Bartel, D.L., 1999. Robust Optimization of Total Joint Replacements Incorporating Environmental Variables. J. Biomech. Eng. 121, 304–310.

Chen, Q., Ye, Q., Zhang, W., Li, H., Zheng, X., 2023. Tgm-nets: A deep learning framework for enhanced forecasting of tumor growth by integrating imaging and modeling. Eng. Appl. Artif. Intell. 126, 106867.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

De Ryck, T., Bonnet, F., Mishra, S., de Bézenac, E., 2023. An operator preconditioning perspective on training in physics-informed machine learning. arXiv preprint arXiv:2310.05801.

Dekhovich, A., Sluiter, M.H., Tax, D.M., Bessa, M.A., 2023. iPINNs: Incremental learning for physics-informed neural networks. arXiv preprint arXiv:2304.04854.

Deng, J., Shahroudi, M., Liu, K., 2023. Dynamic stability and responses of beams on elastic foundations under a parametric load. Int. J. Struct. Stab. Dyn. 23, 2350018.

Erdaş, M.U., Kopar, M., Yildiz, B.S., Yildiz, A.R., 2023. Optimum design of a seat bracket using artificial neural networks and dandelion optimization algorithm. Mater. Test. 65, 1767–1775.

Fallah, A., Aghdam, M.M., 2023. Physics-informed neural network for bending and free vibration analysis of three-dimensional functionally graded porous beam resting on elastic foundation. Eng. Comput. 1–18.

Faria, R., Capron, B., Secchi, A., De Souza, Jr., M., 2024. A data-driven tracking control framework using physics-informed neural networks and deep reinforcement learning for dynamical systems. Eng. Appl. Artif. Intell. 127, 107256.

Fuks, O., Tchelepi, H.A., 2020. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. J. Mach. Learn. Model. Comput. 1.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings. pp. 249–256.

Goswami, S., Anitescu, C., Chakraborty, S., Rabczuk, T., 2020. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theor. Appl. Fract. Mech. 106, 102447.

Haghighat, E., Abouali, S., Vaziri, R., 2023. Constitutive model characterization and discovery using physics-informed deep learning. Eng. Appl. Artif. Intell. 120, 105828.

Hetényi, M., Hetbenyi, M.I., 1946. Beams on elastic foundation: theory with applications in the fields of civil and mechanical engineering., vol. 16, University of Michigan Press, Ann Arbor, MI.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

Jagtap, A.D., Karniadakis, G.E., 2021. Extended physics-informed neural networks (XPINNs): A generalized space–time domain decomposition based deep learning framework for nonlinear partial differential equations. In: AAAI Spring Symposium: MLPS.

Jagtap, A.D., Kawaguchi, K., Karniadakis, G.E., 2020a. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. J. Comput. Phys. 404, 109136.

Jagtap, A.D., Kharazmi, E., Karniadakis, G.E. and, 2020b. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Comput. Methods Appl. Mech. Engrg. 365, 113028.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., et al., 2021. Highly accurate protein structure prediction with alphafold. Nature 596, 583–589.

Kabir, H., Aghdam, M., 2019. A robust bézier based solution for nonlinear vibration and post-buckling of random checkerboard graphene nano-platelets reinforced composite beams. Compos. Struct. 212, 184–198.

Kabir, H., Garg, N., 2023. Machine learning enabled orthogonal camera goniometry for accurate and robust contact angle measurements. Sci. Rep. 13 (1497).

Kapoor, T., Chandra, A., Tartakovsky, D., Wang, H., Núñez, R., 2023a. Neural oscillators for generalizing parametric PDEs. In: The Symbiosis of Deep Learning and Differential Equations III. URL https://openreview.net/forum?id=1J9xoPFaKU.

Kapoor, T., Chandra, A., Tartakovsky, D.M., Wang, H., Nunez, A., Dollevoet, R., 2023b. Neural oscillators for generalization of physics-informed machine learning. arXiv preprint arXiv:2308.08989.

Kapoor, T., Wang, H., Núñez, A., Dollevoet, R., 2023c. Physics-informed machine learning for moving load problems. arXiv preprint arXiv:2304.00369.

Kapoor, T., Wang, H., Núñez, A., Dollevoet, R., 2023d. Physics-informed neural networks for solving forward and inverse problems in complex beam systems. IEEE Trans. Neural Netw. Learn. Syst. 1–15. http://dx.doi.org/10.1109/TNNLS.2023.3310585.

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. Nat. Rev. Phys. 3, 422–440.

Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., Kumar, V., 2017a. Theory-guided data science: A new paradigm for scientific discovery from data. IEEE Trans. Knowl. Data Eng. 29, 2318–2331.

Karpatne, A., Kannan, R., Kumar, V., 2022. Knowledge Guided Machine Learning: Accelerating Discovery using Scientific Knowledge and Data. CRC Press.

Karpatne, A., Watkins, W., Read, J., Kumar, V., 2017b. Physics-guided neural networks (PGNN): An application in lake temperature modeling. arXiv preprint arXiv:1710.11431 2.

Kim, J., Lee, K., Lee, D., Jhin, S.Y., Park, N., 2021. Dpm: a novel training method for physics-informed neural networks in extrapolation. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 8146–8154.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., Mahoney, M.W., 2021. Characterizing possible failure modes in physics-informed neural networks. Adv. Neural Inf. Process. Syst. 34, 26548–26560.

Lai, Z., Mylonas, C., Nagarajaiah, S., Chatzi, E., 2021. Structural identification with physics-informed neural ordinary differential equations. J. Sound Vib. 508, 116196.

Lamprea-Pineda, A.C., Connolly, D.P., Hussein, M.F., 2022. Beams on elastic foundations–a review of railway applications and solutions. Transp. Geotech. 33, 100696.

Lee, J., 2023. Physics informed neural networks for extreme mechanics problems. Available at SSRN 4362563.

Li, S., Wang, G., Di, Y., Wang, L., Wang, H., Zhou, Q., 2023. A physics-informed neural network framework to predict 3D temperature field without labeled data in process of laser metal deposition. Eng. Appl. Artif. Intell. 120, 105908.

Lippe, P., Veeling, B.S., Perdikaris, P., Turner, R.E., Brandstetter, J., 2023. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. arXiv preprint arXiv:2308.05732.

Liu, F., Li, J., Wang, L., 2023. Pi-lstm: Physics-informed long short-term memory network for structural response modeling. Eng. Struct. 292, 116500.

Liu, T., Meidani, H., 2023. Physics-informed neural networks for system identification of structural systems with a multiphysics damping model. J. Eng. Mech. 149, 04023079.

Liu, X., Peng, W., Gong, Z., Zhou, W., Yao, W., 2022. Temperature field inversion of heat-source systems via physics-informed neural networks. Eng. Appl. Artif. Intell. 113, 104902.

Madenci, E., Guven, I., 2015. The Finite Element Method and Applications in Engineering using ANSYS®. Springer.

Mai, H.T., Truong, T.T., Kang, J., Mai, D.D., Lee, J., 2023. A robust physics-informed neural network approach for predicting structural instability. Finite Elem. Anal. Des. 216, 103893.

McClenny, L.D., Braga-Neto, U.M., 2023. Self-adaptive physics-informed neural networks. J. Comput. Phys. 474, 111722.

Meng, Z., Qian, Q., Xu, M., Yu, B., Yıldız, A.R., Mirjalili, S., 2023. Pinn-form: A new physics-informed neural network for reliability analysis with partial differential equation. Comput. Methods Appl. Mech. Engrg. 414, 116172.

Michałowska, K., Goswami, S., Karniadakis, G.E., Riemer-Sørensen, S. and, 2023. Neural operator learning for long-time integration in dynamical systems with recurrent neural networks. arXiv preprint arXiv:2303.02243.

Mu, B., Qin, B., Yuan, S., Wang, X., Chen, Y., 2023. PIRT: A physics-informed red tide deep learning forecast model considering causal-inferred predictors selection. IEEE Geosci. Remote Sens. Lett. 20, 1–5.

Niu, S., Liu, Y., Wang, J., Song, H., 2020. A decade survey of transfer learning (2010–2020). IEEE Trans. Artif. Intell. 1, 151–166.

Öchsner, A., 2021. Classical Beam Theories of Structural Mechanics, vol. 42, Springer.

Olivares, E., Ye, H., Herrero, A., Nia, B.A., Ren, Y., Donovan, R.P., et al., 2021. Applications of information channels to physics-informed neural networks for wifi signal propagation simulation at the edge of the industrial internet of things. Neurocomputing 454, 405–416.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch.

Penwarden, M., Jagtap, A.D., Zhe, S., Karniadakis, G.E., Kirby, R.M., 2023. A unified scalable framework for causal sweeping strategies for physics-informed neural networks (pinns) and their temporal decompositions. arXiv preprint arXiv:2302.14227.

Petrosian, L.G., 2022. Analysis of Structures on Elastic Foundation: Incorporating the Spectral Method of Boundary Elements. CRC Press.

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

Roy, A.M., Guha, S., 2023. A data-driven physics-constrained deep learning computational framework for solving von mises plasticity. Eng. Appl. Artif. Intell. 122, 106049.

Shen, S., Lu, H., Sadoughi, M., Hu, C., Nemani, V., Thelen, A., Webster, K., Darr, M., Sidon, J., Kenny, S., 2021. A physics-informed deep learning approach for bearing fault detection. Eng. Appl. Artif. Intell. 103, 104295.

Shukla, K., Jagtap, A.D., Karniadakis, G.E., 2021. Parallel physics-informed neural networks via domain decomposition. J. Comput. Phys. 447, 110683.

Tsudik, E., 2012. Analysis of Structures on Elastic Foundations. J. Ross Publishing.

Uddin, Z., Ganga, S., Asthana, R., Ibrahim, W., 2023. Wavelets based physics informed neural networks to solve non-linear differential equations. Sci. Rep. 13 (2882).

Viana, F.A., Nascimento, R.G., Dourado, A., Yucesan, Y.A., 2021. Estimating model inadequacy in ordinary differential equations with physics-informed neural networks. Comput. Struct. 245, 106458.

Wang, S., Sankaran, S., Perdikaris, P., 2022. Respecting causality is all you need for training physics-informed neural networks. arXiv preprint arXiv:2203.07404.

Whittaker, T., Janik, R.A., Oz, Y., 2023. Neural network complexity of chaos and turbulence. Eur. Phys. J. E 46 (57).

Xu, C., Cao, B.T., Yuan, Y., Meschke, G., 2023. Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios. Comput. Methods Appl. Mech. Engrg. 405, 115852.

Yildiz, A., Öztürk, N., Kaya, N., Öztürk, F., 2003. Integrated optimal topology design and shape optimization using neural networks. Struct. Multidiscip. Optim. 25, 251–260.

Younesian, D., Hosseinkhani, A., Askari, H., Esmailzadeh, E., 2019. Elastic and viscoelastic foundations: a review on linear and nonlinear vibration modeling and applications. Nonlinear Dynam. 97, 853–895.

Yu, J., Lu, L., Meng, X., Karniadakis, G.E., 2022. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. Comput. Methods Appl. Mech. Engrg. 393, 114823.

Zhai, W., Tao, D., Bao, Y., 2023. Parameter estimation and modeling of nonlinear dynamical systems based on runge–kutta physics-informed neural network. Nonlinear Dynam. 111, 21117–21130.

Zobeiry, N., Humfeld, K.D., 2021. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. Eng. Appl. Artif. Intell. 101, 104232.