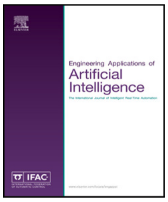
Engineering Applications of Artificial Intelligence 181 (2026) 115285

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Research paper

A unified causality-enhanced separable physics-informed neural network for predicting beam and plate dynamics

Anastasios Stamou [a],*, Taniya Kapoor [b], Michalis Fragiadakis [a]

[a] National Technical University of Athens, 9 Iroon Polytechniou, Zografou Campus, Athens, 15780, Greece
[b] Wageningen University & Research, 4 Droevendaalsesteeg, Wageningen, 6708 PB, The Netherlands

ARTICLE INFO

Keywords:
Separable physics-informed neural networks
Causal training
Auxiliary variables
Structural dynamics
High-order partial differential equations
Inverse identification

ABSTRACT

A thorough understanding of the dynamic behavior of structural elements, such as beams, plates, and membranes, is crucial for reliable structural analysis. This behavior is typically governed by higher-order partial differential equations (PDEs), which accurately capture their spatiotemporal response. However, the presence of high-order differential operators and inherent high-frequency components, especially over large spatial domains, poses challenges for Physics-Informed Neural Networks (PINNs). To tackle this challenge, we propose a causality-informed Separable Physics-Informed Neural Network (SPINN) framework augmented with auxiliary variables (Causal Aux SPINN) to enhance the performance of PINNs in structural engineering applications. The proposed framework integrates three key components. First, the SPINN formulation decomposes the spatiotemporal solution into low-rank components, allowing efficient representation of high-dimensional dynamic fields with reduced computational cost. Second, the causality-informed loss function re-weights the training objective to respect the temporal evolution of structural dynamics and mitigate error accumulation over time. Third, the auxiliary-variable formulation introduces additional outputs for second-order fields, such as curvature-related quantities, enabling accurate enforcement of fourth-order differential operators without relying solely on successive differentiation of displacement fields. The proposed "Causal Aux SPINN" algorithm is evaluated on both forward and inverse benchmarks in structural dynamics, including Euler–Bernoulli and Timoshenko beams, membranes, and Kirchhoff–Love plates. Across all benchmarks, the Causal Aux SPINN method consistently reduces error and improves robustness, while maintaining computational efficiency comparable to the Baseline SPINN formulation. Overall, coupling causal weighting with auxiliary-variable learning is shown to provide an effective framework for structural engineering applications.

1. Introduction

Beams, membranes, and plates constitute the fundamental load-bearing components of modern infrastructure. Accurately predicting their dynamic response and load-carrying capacity is essential to ensure the strength, safety, and serviceability of large-scale structures (Feng et al., 2021; Liu et al., 2022). The dynamic response of these elements depends strongly on boundary conditions, geometry, and loading characteristics, and is governed by high-order partial differential equations (PDEs). Because even millimeter-level discrepancies in displacements or their spatial and temporal derivatives can lead to unsafe underestimation or excessive conservatism, robust and precise simulation models are required. Such models enable engineers to design efficient and sustainable structural systems by minimizing material use while maintaining stability and reliability (Kapoor et al., 2023). Numerical methods, such as finite differences (Strikwerda, 2004), finite elements (Reddy, 1993), and finite volumes (Eymard et al., 2000), have long been employed to solve PDEs. Though highly versatile and accurate, these approaches face notable limitations. For instance, mesh generation for complex geometries is challenging (Karniadakis et al., 2021), while even minor changes in geometry or boundary conditions typically require the complete reformulation and recomputation of the simulation (Kapoor et al., 2024a). Such requirements hinder generalization and increase the computational cost. To overcome these drawbacks, recent research has increasingly explored data-driven and machine learning-based methods that can learn mappings between inputs and solutions without explicit meshing or full model reformulation.

Physics-Informed Machine Learning (PIML), which integrates governing physical laws into deep learning architectures, has been gaining increasing attention for approximating the solution field of PDEs (Raissi et al., 2019; Karniadakis et al., 2021; Cuomo et al., 2022). Within this branch of deep learning, Physics-Informed Neural Networks (PINNs)

---

* Corresponding author.
E-mail addresses: stamou@mail.ntua.gr (A. Stamou), taniya.kapoor@wur.nl (T. Kapoor), mfrag@mail.ntua.gr (M. Fragiadakis).

https://doi.org/10.1016/j.engappai.2026.115285
Received 9 November 2025; Received in revised form 27 May 2026; Accepted 29 May 2026

(Raissi et al., 2019) have emerged as a viable approach, demonstrating success across diverse engineering fields, including aerospace (Wang et al., 2025), structural engineering (Yu et al., 2020; Kapoor et al., 2024b), and complex fluid and thermal systems such as magnetohydrodynamic flows (Ahmad and Zafar, 2025; Jawad et al., 2025) and nanofluid heat transfer (Ahmad et al., 2026). Despite notable advances, conventional PINNs still face difficulties in accurately reproducing high-frequency structural responses, particularly over large spatial and temporal domains. These limitations have constrained their practical use mainly to small-scale beam deformation problems (Kapoor et al., 2023; Bazmara et al., 2023). As the spatial or temporal extent increases, the efficiency of PINNs deteriorates, resulting in a trade-off between accuracy and computational cost (Moseley et al., 2023; Karniadakis et al., 2021).

Several recent studies have sought to improve the accuracy, robustness, and efficiency of Baseline PINNs through modifications to the network architecture, the training strategy, or the loss formulation. Adaptive learning (McClenny and Braga-Neto, 2023), transformer-inspired architectures (Zhao et al., 2023), diffusion-based formulations (Shysheya et al., 2024), domain decomposition strategies (Hu et al., 2023), and sequential time-marching schemes (Mattey and Ghosh, 2022) have been proposed, each addressing specific bottlenecks of PINNs. Recent work has also explored the role of hybrid activation functions in improving PINN training performance (Ahmad et al., 2025). Within this broader landscape, several directions are particularly relevant to this work. First is the enforcement of temporal causality. In the Baseline PINN formulation of Raissi et al. (2019), the PDE residual is minimized over collocation points sampled throughout the space–time domain without explicitly enforcing temporal ordering. Consequently, unresolved errors at early time instances may propagate into the learning of later-time dynamics. Causality-enforcing weighting schemes (Wang et al., 2024) address this by down-weighting later-time residuals until earlier ones have been sufficiently resolved. This formulation has recently been adapted to long-time-horizon predictions in beam dynamics (Kapoor et al., 2024a), albeit with a significantly higher computational overhead. A second issue is the accurate enforcement of high-order differential operators. A-PINN formulations (Yuan et al., 2022) introduced auxiliary output variables to recast nonlinear integro-differential equations as purely differential systems, replacing integral discretization with automatic differentiation of the auxiliary outputs. While developed for integral operators rather than high-order derivatives, the auxiliary-variable principle is directly transferable to the fourth-order operators arising in beam and plate equations, where it can be used in the opposite direction, to decompose high-order derivatives into systems of lower-order ones.

When it comes to structural engineering applications, energy-based PINNs for graded porous beams (Eshaghi et al., 2025) and non-dimensionalized residual approaches (Kapoor et al., 2023; Fallah and Aghdam, 2024) have been explored. Leitao & Kang introduced early mesh-free formulations for plates and membranes (Leitao, 2001; Kang and Lee, 2000), which have since been extended to PINN frameworks (Haghighat et al., 2021). In parallel, several efforts have targeted the computational cost of PINNs through hybrid and low-rank formulations. Hybrid approaches, such as the deep finite element method that integrates PINNs with classical finite element method discretizations, have been proposed to address the computational and stability challenges of PINNs by leveraging established numerical solvers (Xiong et al., 2025). Of particular relevance to this work, Cho et al. (2023) introduced SPINNs, which approximate spatio-temporal solutions using low-rank decompositions to reduce training cost. Extensions to other model architectures, such as separable deep operator networks (Mandl et al., 2025), generalize the concept of SPINN to operator learning, further highlighting the potential of low-rank structures for high-dimensional PDEs.

Despite these advances, Baseline PINNs and many existing variants remain limited in two respects for high-order structural dynamics. First, temporal causality is either not explicitly enforced or is enforced at considerable computational cost. This allows unresolved early-time errors to propagate through the learned dynamics. Second, the repeated automatic differentiation required to evaluate high-order operators can amplify approximation errors, reducing the accuracy of derivative-dependent quantities in models such as the Euler–Bernoulli beam and Kirchhoff–Love plate equations.

To address the limitations discussed above, this paper introduces a novel causality-enhanced separable physics-informed neural network framework for high-order structural dynamics problems. The framework leverages non-trivial interactions between separable architectures, causal temporal weighting, and auxiliary-variable decompositions to jointly address challenges that no single component resolves alone. These limitations are temporal error accumulation, deep automatic-differentiation graphs in fourth-order operators, and error amplification in design-critical derived quantities. Specifically, (*i*) causal weighting (Wang et al., 2024), originally developed for standard PINNs, is adapted to the SPINN separable architecture to enforce temporal ordering under low-rank approximations. Its interaction with the shared branch structure and rank parameter r governs how early-time dynamics are resolved before later times and introduces behavior not previously studied. (*ii*) An auxiliary-variable formulation (Yuan et al., 2022) decomposes fourth-order differential operators into second-order, reducing automatic differentiation depth and mitigating the error amplification that arises in high-frequency modal content. Unlike A-PINN formulations designed for integral operators, this decomposition addresses the differential structure of beam and plate equations and interacts with both the separable architecture and causal weighting in ways that have no counterpart in the original formulation. (*iii*) The framework examines not only displacements but also design-critical derived quantities such as bending moments ($M = EIu_{xx}$) and shear forces ($V = EIu_{xxx}$), making it directly applicable to structural assessment and digital-twin updating.

The proposed method is validated through six benchmark problems involving Euler–Bernoulli and Timoshenko beams, free-vibrating membranes, and Kirchhoff–Love plates, in both forward and inverse settings. Across all cases, Causal SPINN and Causal Aux SPINN demonstrate consistent improvements in accuracy and stability over Baseline SPINN, with less than 8% additional training time and reduced GPU memory when adding the auxiliary variable. The proposed enhanced SPINN formulation remains robust under noisy initial conditions and learns unknown structural parameters, such as flexural rigidity for the Kirchhoff–Love plate and bending stiffness for the Euler–Bernoulli beam, from sparse sensor data. The obtained results highlight that the proposed enhanced SPINN architecture provides a scalable and efficient framework for simulating high-order structural dynamics.

The rest of the manuscript is structured as follows. Section 2 presents the Baseline PINN-based methods and the proposed methodology in detail. Section 3 presents the numerical experiments for beams, and Section 4 presents the numerical experiments related to membranes and plates, to validate the proposed methodology. The main conclusions drawn from this study are collated in Section 5.

2. Proposed methodology

This section presents the proposed methodology in four parts, beginning with a brief overview of PINNs and their separable extension, which utilize low-rank tensor decompositions to represent high-dimensional PDE solutions efficiently. Next, forward-mode automatic-differentiation is described. This approach facilitates fast and accurate evaluation of the higher-order derivatives required in structural dynamics. Subsequently, a causality-informed loss function is introduced to enforce the temporal ordering of errors. In addition, SPINN is augmented with an auxiliary-variable formulation to robustly capture high-order PDE operators, leveraging works such as those of Wang and Yuan (Wang et al., 2024; Yuan et al., 2022). Throughout this paper, we refer to the complete framework as **Causal Aux SPINN** (Causality-enhanced Auxiliary-variable Separable Physics-Informed Neural Network). When the auxiliary variable is not used, we write **Causal SPINN**; the version without causal weighting is termed **Baseline SPINN**; and SPINN with the auxiliary variable but without the causal loss is referred to as **Aux SPINN**.

2.1. PINNs and SPINNs

PINNs are Neural Networks that embed physical laws, expressed as differential equations, into their learning process to solve scientific and engineering problems. Raissi et al. (2019). A PINN approximates the solution of a PDE, denoted by $u(\mathbf{X})$, where $\mathbf{X} = [x_1, \dots, x_d, t]^T$ represents the spatial and temporal coordinates. Here $d \in \mathbb{N}$ represents the number of spatial dimensions. The approximation is obtained by training a neural network $\hat{u}(\mathbf{X}) = f(\mathbf{X}; \phi)$ where $f(\mathbf{X}; \phi)$ denotes the neural network mapping parameterized by the trainable weights and biases ϕ. This function approximates the underlying physical field $u(\mathbf{X})$, e.g., displacement, temperature, or pressure. The network parameters are obtained by minimizing a composite loss function $\mathcal{L}$ that enforces the governing physics:

$$\mathcal{L}_{\text{PINN}} = \lambda_{res}\mathcal{L}_{\text{res}} + \lambda_{bc}\mathcal{L}_{\text{bc}} + \lambda_{ic}\mathcal{L}_{\text{ic}}, \tag{1}$$

where $\mathcal{L}_{\text{res}}$ penalizes the residual of the governing PDE, $\mathcal{L}_{\text{bc}}$ enforces the boundary conditions, and $\mathcal{L}_{\text{ic}}$ enforces the initial conditions. The coefficients $\{\lambda_{\text{res}}, \lambda_{\text{bc}}, \lambda_{\text{ic}}\}$ act as *regularization weights* that balance the contributions of each term. They may be selected heuristically, through grid search, or by more general adaptive weighting strategies during training. These loss terms are obtained by substituting the Neural Network approximation $\hat{u}(\mathbf{X})$ into the governing equations and constraints, with the required derivatives evaluated through automatic differentiation. Despite their flexibility and ability to incorporate both data and physical constraints, standard PINNs often face high computational overhead and poor scalability to high-dimensional settings. This limitation arises because the entire space–time domain is represented by a single, typically wide and deep, monolithic network.

The SPINN (Cho et al., 2023) architecture is an efficient, tensor-structured variant of classical PINNs (Karniadakis et al., 2021). Instead of feeding the full coordinate-time tuple $\mathbf{X} = [x_1, \dots, x_d, t]^T$ through a single, wide multilayer perceptron, SPINN deploys an independent body network for every input axis $x_i, 1 \le i \le d, i \in \mathbb{N}$.

For time-dependent problems, the time is treated as an additional independent variable with its own body network. The partial features are then fused by a weight-free outer-product layer that implements a canonical polyadic (CP) decomposition:

$$u_{NN}(\mathbf{X}) = \sum_{j=1}^{r} \prod_{i=1}^{d} f_j^{(i)}(x_i), \tag{2}$$

where $f_j^{(i)} : \mathbb{R} \to \mathbb{R}$ denotes the jth output feature of the ith body network and r is the *decomposition rank*. A choice of $r = 1$ forces an entirely separable solution, whereas moderate ranks match or surpass a monolithic PINN with up to roughly 60× lower runtime and about 3-5% of the memory footprint on specific reported benchmarks by Cho et al. (2023). The single additional hyperparameter, r, can be tuned by monitoring the PDE residual during training. As a moderate rank, r values in the range of 32-128 are considered in this work, consistent with empirical observations, which helps achieve the best trade-off between error and computational overhead.

Evaluating Eq. (2) on a Cartesian grid of N points per axis requires $\mathcal{O}(N^d)$ evaluations overall. Since SPINN factorizes the computation, the body networks supply r-dimensional feature vectors at each of the d axes for all grid points in $\mathcal{O}(dN)$. The outer-product layer then broadcasts and multiplies these vectors without additional inference passes. Automatic differentiation benefits from the same reduction in network propagations, e.g., via forward-mode Jacobian Vector Products (JVPs). As a result, the number of network/derivative evaluations scales as $\mathcal{O}(dN)$ rather than $\mathcal{O}(N^d)$. According to Cho et al. (2023), the current implementation re-computes feature representations $f^{(\theta_i)}$ during gradient calculation, raising the practical complexity of network propagations to $\mathcal{O}(Nd^2)$. Since $Nd^2 \ll N^d$ for the problem dimensions considered here ($d = 2, 3$), the computational advantage over PINNs is preserved. Empirical evidence has shown that the wall-clock time and the memory requirements grow nearly linearly with the number of collocation points.

Cho et al. (2023) also evaluated PINN/SPINN variants that use the "modified MLP" backbone introduced by Wang et al. (2021). Since Cho et al. (2023) reported consistent accuracy gains with this backbone without noticeable computational overhead, we have also adopted this backbone in our implementation. Another benefit of SPINN is that every axis is handled by its own network, allowing SPINN to extend naturally to anisotropic settings. Thus, different widths, depths, or activation functions can be assigned to each coordinate when prior knowledge suggests unequal difficulty across axes.

2.2. Forward-mode automatic differentiation and Jacobian-vector products

When training a SPINN, spatial derivatives, sometimes up to fourth order, must be evaluated to enforce the governing PDEs. Automatic Differentiation (AD) provides these derivatives while avoiding the truncation errors of numerical schemes. AD applies the chain rule systematically to elementary operations and comes in two primary differentiating strategies: *reverse mode* and *forward mode*. Reverse-mode AD is the standard choice in Machine Learning (ML). However, *forward-mode* AD (Baydin et al., 2018) is often more efficient for SPINN because its computational cost scales with the number of *inputs* (one per body network), whereas reverse-mode scales with the number of *outputs*. Since each body network in SPINN has a scalar input but produces an r-dimensional feature vector, forward-mode becomes advantageous in both time and memory. This property aligns naturally with SPINN's per-axis decomposition, where spatial derivatives with respect to each input coordinate are required. Forward-mode automatic differentiation augments each primal value with a tangent component. This tangent tracks the sensitivity of the output to perturbations along a chosen input direction $\mathbf{v}$. Both primal and tangent values propagate layer by layer through the computation graph, yielding what is known as a Jacobian-Vector Product (JVP). A JVP corresponds to the directional derivative of f along $\mathbf{v}$ and is expressed as:

$$\text{JVP}(f, \mathbf{x}; \mathbf{v}) = \mathbf{J}_f(\mathbf{x})\mathbf{v} \tag{3}$$

where $\mathbf{J}_f(\mathbf{x}) \in \mathbb{R}^{m\times n}$ is the Jacobian of $f : \mathbb{R}^n ! \to ! \mathbb{R}^m$ at the point $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{v} \in \mathbb{R}^n$ is an arbitrary direction. Reseeding the input successively with the standard basis vectors $\{\mathbf{e}_i\}_{i=1}^n$, where $\mathbf{e}_i$ denotes the ith unit vector in $\mathbb{R}^n$—recovers the full Jacobian. Nesting JVPs, i.e., applying one JVP inside another, yields higher-order derivatives of any order (Griewank and Walther, 2008). For instance, the mixed derivative u_{xxyy} is obtained by first seeding twice with $\mathbf{e}_x$ and then re-applying two more times the JVP with $\mathbf{e}_y$. In an d-dimensional setting, high-order derivatives can be systematically recovered via repeated forward-mode sweeps along the coordinate directions. Since SPINN evaluates a shared one-dimensional body network on each axis, the *number of network/derivative evaluations* required to generate all spatial derivatives over an N^d grid remains $\mathcal{O}(dN)$. This cost is comparable to that of a single forward pass. The subsequent broadcast to the N^d lattice is parameter-free and inexpensive. Forward-mode stores only the current primal and tangent values, so its memory footprint is $\mathcal{O}(1)$ in network depth. Reverse-mode, on the other hand, requires storing a full computation trace of intermediate activations for back-propagation. The low-rank, separable structure of SPINN further limits the intermediate tensor sizes and the Jacobian-vector products produced by forward-mode remain in the same compact format. This approach ensures that the computational and memory savings of both techniques are preserved throughout the entire training pipeline. Hence, the combination of forward-mode AD and SPINN's separable structure enables the efficient evaluation of high-order derivatives.

2.3. Causality-informed SPINN

The proposed method, called Causal SPINN, augments the Baseline SPINN framework with a causality-informed training objective. This mechanism reinforces the correction of errors present at early time steps. Inaccuracies at the initial condition tend to propagate and amplify over subsequent time steps. The training objective is reshaped so that early residuals contribute more strongly to the gradient. This encourages the optimizer to repair predictions at early time steps first. This practice mitigates downstream error growth and propagation, thereby improving overall accuracy and convergence.

Let the time-dependent PDE loss term be evaluated at a discrete set of N_T time instances, i.e., $\mathbf{t} = [t_1, t_2, \dots, t_{N_T}]$. For every t_i, the squared residual is computed and then integrated over the spatial domain. In other words, it is averaged over the spatial dimension(s) to obtain a scalar loss L_t at time instant t_i. These losses form vector $\mathbf{L} = [L_1, L_2, \dots, L_{N_T}]^\top$. To encode temporal causality, we construct the strictly *lower-triangular* matrix:

$$\mathbf{W} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \end{bmatrix}, \tag{4}$$

whose action accumulates all errors that precede every time step. Multiplying $\mathbf{W}$ and $\mathbf{L}$ yields the vector of running sums, i.e $\mathbf{S} = \mathbf{WL}$. Each component of $\mathbf{S}$ is:

$$S_i = \sum_{\tau=1}^{i-1} L_\tau, \qquad i = 2, 3, \dots, N_T, \qquad S_1 = 0. \tag{5}$$

The loss term of every row, L_{t_i}, is weighted by an exponential factor that depends on the temporal history. The result is then averaged over the domain:

$$\mathcal{L}_{\text{Causal}} = \frac{1}{N_T} \sum_{t=1}^{T} L_t \exp\bigl(-\varepsilon\, S_t\bigr). \tag{6}$$

The scalar hyper-parameter $\varepsilon > 0$ controls the strength of the causal emphasis. As $\varepsilon \to 0$, the exponential weights approach one, and Eq. (6) reduces to the uniform-in-time objective of Baseline SPINN. For larger values of ε, the influence of early time-step errors is amplified, guiding the optimizer to suppress them before gradually shifting attention to later time instances. For the dynamic problems examined in this work, a ε value between 3 and 5 was found suitable. This represents a deviation from the original formulation by Wang et al. (2024), who employ an annealing schedule with values $\varepsilon = [10^{-2}, 10^{-1}, 1, 10, 100]$ to handle chaotic and multi-scale dynamics. For the linear structural dynamics problems studied here, the smooth and monotonic decay of PDE residuals during training, renders a fixed ε sufficient, as the causal weights self-regulate without the need for progressive ramping.

All experiments in this work use a fixed temporal resolution between $N_t = 64 - 128$. Finer discretizations would distribute the cumulative sum S_i over more time steps, potentially requiring re-tuning of ε or the number of training epochs. The exponential form is stable by construction: because the cumulative loss sum S_i is non-negative, the weights are bounded in $[0, 1]$, preventing numerical overflow regardless of the choice of ε. The lower-triangular structure of $\mathbf{W}$ is the key mechanism: by definition, the weight $w_i = \exp(-\varepsilon\, S_i)$ for time step t_i depends only on the cumulative residual loss at earlier time steps $t_1, \dots, t_{i-1}$. This ensures that later time steps receive meaningful gradient signal only after earlier ones are sufficiently resolved, directly enforcing the temporal causality of the initial-value problem in the optimization process.

Causal weighting is especially important for the separable SPINN architecture. In a standard PINN, each prediction $u(x_i, t_j)$ is computed independently from the concatenated input $[x_i, t_j]$. In SPINN, however, the time-branch output $f(t_j) \in \mathbb{R}^r$ is shared across *all* spatial points, while the spatial-branch output $g(x_i) \in \mathbb{R}^r$ is shared across all time points. This parameter-sharing structure means that gradient signals from large residuals at later, unresolved time steps directly affect the shared internal features of the time branch. This interference can degrade predictions at earlier, already well-resolved time steps. Causal weighting addresses this by suppressing gradient contributions from later time steps where residuals are large. The time branch first learns accurate representations for early times, where initial and boundary conditions provide strong supervision. These features are then progressively refined as the causal window opens to later time steps.

In practice, this creates a self-regulating, curriculum-like training effect. Fig. 1 illustrates this behavior for the Euler–Bernoulli beam benchmark ($r = 128$, $\varepsilon = 5$, 250k epochs). At epoch 10k, only the first ∼2% of the time domain carries non-negligible weight. As early-time residuals decrease, the causal window progressively expands, covering the entire domain by epoch 75k. The remaining training epochs proceed with near-uniform weights across all time steps.

SPINN outputs a tensor of shape $\{N_T, N_x, N_y, \dots\}$, proportional to the dimensionality of the examined PDE. Therefore, no additional reshaping operations are required to assemble the losses over time. Moreover, since the exponential weights in Eq. (6) are positive and differentiable, the modified objective preserves the smoothness assumptions needed by standard gradient-based optimizers. Existing SPINN training pipelines can thus adopt the proposed Causal SPINN method with only minimal code changes.

2.4. Auxiliary-variable causality-informed SPINN

To complement the causal weighting described above, an auxiliary field v is introduced to enhance SPINN accuracy for *high-order* PDEs common in structural mechanics. This field approximates key second-order derivatives, inspired by the auxiliary-variable formulation of Yuan et al. (2022). This formulation is denoted in the remainder of the paper as *Causal-Aux-SPINN*. Depending on the problem examined, accurately predicting second-order spatial derivatives of the displacement field is often essential. The guiding principle is to decompose a fourth-order spatial operator into a pair of second-order relationships that are easier for a Neural Network to learn and less susceptible to vanishing-gradient pathologies.

The auxiliary variable v depends on the form of the PDE considered and thus the definition may differ according to the PDE at hand. For the one-dimensional Euler–Bernoulli beam problem considered in this study, the auxiliary variable v is defined as:

$$v = u_{xx}, \tag{7}$$

where $u_{xx} = \partial^2 u / \partial^2 x$. For the two-dimensional Kirchhoff–Love plate, the auxiliary variable is defined as

$$v = u_{xx} + u_{yy}, \tag{8}$$

The double partial derivatives with respect to the x and y coordinates provide the fourth-order operator:

$$\nabla v = v_{xx} + v_{yy} \; = \; u_{xxxx} + 2\, u_{xxyy} + u_{yyyy}. \tag{9}$$

The auxiliary variable eliminates the need for successive automatic differentiation through four network layers to compute u_{xxxx}. Instead, v_{xx} requires only second-order differentiation of the auxiliary output, while the consistency constraint $u_{xx} - v = 0$ similarly involves only second-order derivatives of u. This decomposition mitigates the amplification of numerical errors inherent in successive higher-order differentiation. In particular, neural networks exhibit a well-known bias toward learning low-frequency functions (Rahaman et al., 2019), and computing u_{xxxx} via four successive differentiations amplifies high-frequency errors at each stage. By introducing v as an explicit network output with the consistency constraint, the network is provided with

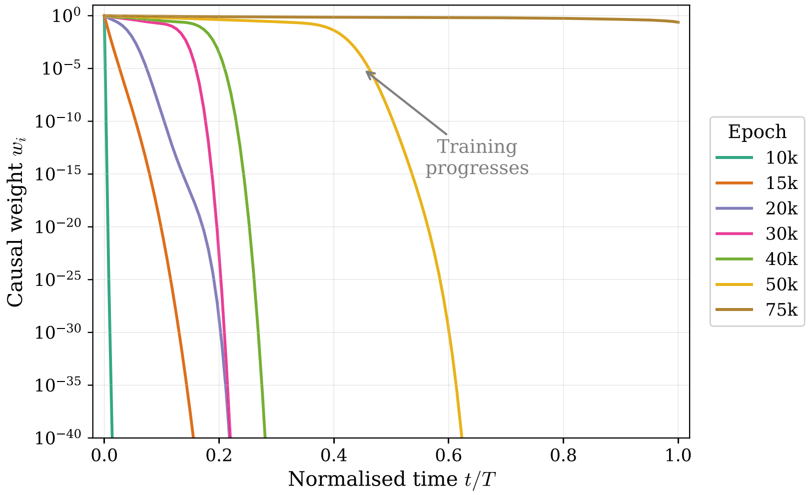


Fig. 1. Causal weight profiles w_i versus normalized time t/T at selected training epochs for the Euler–Bernoulli beam (Causal SPINN, $r = 128$, $\varepsilon = 5$).

a direct learning target for the curvature field, bypassing the cascaded differentiation that amplifies this spectral bias.

By introducing the auxiliary variable, the separable networks extend their predictions from a single output u to (u, v). Since both u and v share spatial and temporal features, they can be approximated by employing the same backbone with a lightweight branch head for v, adding less than 8% additional training time while actually reducing GPU memory by replacing fourth-order AD graphs with second-order ones.

In the SPINN separable representation, the auxiliary variable naturally inherits the separable form: each spatial body network need only capture second-order spatial features rather than fourth-order ones, which is a meaningful reduction in per-network complexity. For a given rank r, this means each rank-one component must represent simpler spatial features, effectively increasing the expressiveness of the low-rank approximation without requiring a larger r.

The auxiliary term should also be accounted for in the loss term that is to be minimized by the training process. The causal loss on the primary field u is augmented with a consistency residual that enforces agreement between v and the appropriate second-order derivative of u. The additional loss term is computed as a point-wise mismatch term c that depends on the problem at hand. For an Euler–Bernoulli beam problem, $c_v = v - u_{xx}$, while for a Kirchhoff–Love plate $c_v = v - (u_{xx} + u_{yy})$. The mismatch term is treated as a typical loss term, so the squared mean over the spatial domain is extracted to obtain the per-time-step penalty $L_v(t)$. The total PDE residual loss becomes:

$$L_{\mathrm{res,tot}}(t) = L_{\mathrm{res}}(t) + \lambda_v \, L_v(t), \tag{10}$$

where the hyperparameter $\lambda_v > 0$ balances the other loss terms against the consistency term so that both contribute meaningfully to the optimization. Substituting $L_{\mathrm{res,tot}}(t)$ with $L(t)$ in Eq. (6) couples the auxiliary mechanism with the causality-based learning. During the training of the Neural Network, the gradients from L_v provide an auxiliary supervisory signal that biases the network toward learning accurate second-order features. This remains effective even when the fourth-order residual is too weak to exert a meaningful influence on the early layers.

The value of λ_v controls the efficiency of the loss function. A grid search over λ_v $(10^{-1} - 10^{1})$ and over the causal strength ε quickly identifies a region over which the auxiliary term improves the training without dominating it. Once λ_v is tuned, the same value transfers well between beam and plate problems of comparable spatial resolution, suggesting a degree of robustness. Since the auxiliary parameter v is used only for the first and the second derivatives, the computational cost in terms of memory is reduced because of the shallower AD graph.

The overall workflow of the proposed framework is illustrated in Fig. 2, and the complete training procedure is formalized in Algorithm 1. Temporal and spatial coordinates are first passed to the per-axis *SPINN* body networks. Their outputs are combined via an outer product to reconstruct the solution field $u(\mathbf{X}, t)$ and, in the auxiliary-variable setting, its companion derivative consistency field v. Forward-mode AD is then applied to obtain the required temporal and spatial derivatives. From these, per-time-step residuals are assembled for both the governing PDE and the v-consistency condition. These residuals are modulated by a causal weighting mask $\mathbf{W}$ (lower-triangular matrix in time) that prioritizes early-time accuracy. The resulting weighted loss is then combined with boundary and initial condition terms before being minimized through gradient-based optimization. Repeated iterations of

Algorithm 1 Causal Aux SPINN Training Procedure

```
Require: Collocation grids $\{x_i\}_{i=1}^{N_x}$, $\{t_j\}_{j=1}^{N_t}$; PDE order $p$; causal strength $\varepsilon$; loss weights $\lambda_{\mathrm{bc}}, \lambda_{\mathrm{ic}}, \lambda_v$; learning rate $\eta$; epochs $E$
Ensure: Trained parameters $\phi^*$
Initialize body network parameters $\phi$
useAux ← $(p \geq 4)$
for epoch $= 1, \dots, E$ do
    Forward pass: Evaluate body networks $f^{(i)}(x_i; \phi)$ and $f^{(t)}(t_j; \phi)$
    Reconstruct $u(\mathbf{X}, t)$ via outer product (Eq. (2))
    if useAux then
        Reconstruct auxiliary field $v(\mathbf{X}, t)$ via outer product
    end if
    Differentiation: Compute spatial and temporal derivatives of $u$ via forward-mode AD (JVPs)
    PDE residuals: Evaluate per-time-step residual $\ell_j^{\mathrm{res}}$ for $j = 1, \dots, N_t$
    BC loss: $\mathcal{L}_{\mathrm{bc}} = \frac{1}{N_{\mathrm{bc}}} \sum_i \big(u(x_i^{\mathrm{bc}}, t_i^{\mathrm{bc}}) - g_{\mathrm{bc}}(x_i^{\mathrm{bc}}, t_i^{\mathrm{bc}})\big)^2$
    IC loss: $\mathcal{L}_{\mathrm{ic}} = \frac{1}{N_{\mathrm{ic}}} \sum_i \big(u(x_i^{\mathrm{ic}}, 0) - u_0(x_i^{\mathrm{ic}})\big)^2$
    if useAux then
        Auxiliary residual: $\ell_j^v = \frac{1}{N_x} \sum_i (v_i - u_{xx,i})^2$ at each time step $j$
    else
        $\ell_j^v \leftarrow 0$ for all $j$
    end if
    Causal weights: $w_j = \exp\big(-\varepsilon \sum_{k=1}^{j-1} \ell_k^{\mathrm{res}}\big)$
    Weighted loss: $\mathcal{L}_{\mathrm{res}} = \frac{1}{N_t} \sum_j w_j \, \ell_j^{\mathrm{res}}$;   $\mathcal{L}_v = \frac{1}{N_t} \sum_j w_j \, \ell_j^v$
    $\mathcal{L} = \mathcal{L}_{\mathrm{res}} + \lambda_{\mathrm{bc}} \mathcal{L}_{\mathrm{bc}} + \lambda_{\mathrm{ic}} \mathcal{L}_{\mathrm{ic}} + \lambda_v \mathcal{L}_v$
    Update $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}$
end for
return $\phi^* = \phi$
```

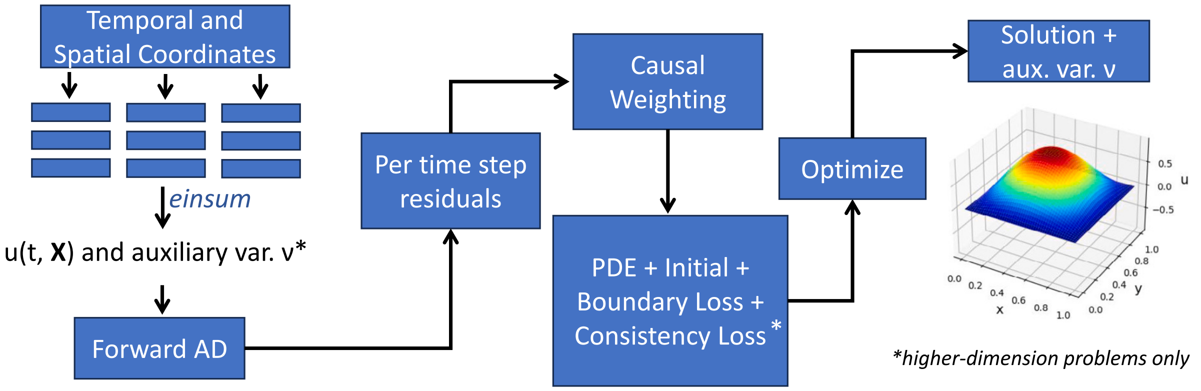


Fig. 2. Causal Aux SPINN framework. Per-axis body networks process spatial coordinates and time independently. An outer-product layer fuses their outputs into the solution field u and auxiliary field v. Forward-mode AD calculates the required derivatives. Per-time-step PDE and auxiliary residuals are then assembled and modulated by the causal weight mask $\mathbf{W}$. The resulting weighted losses are minimized jointly with boundary and initial condition terms.

this procedure yield the final approximation of the solution field u together with the auxiliary variable v.

Augmenting causal SPINN with auxiliary variables, therefore, preserves the temporal stability and provides high-order accuracy. The formulation is readily generalizable: any high-order PDE can be addressed without structural modifications to the algorithm. This makes the method a versatile extension of the SPINN framework.

2.5. Error metrics

To quantify the performance of the neural solutions $\hat{u}(\mathbf{X}, t)$ with respect to the ground truth, reference field $u(\mathbf{X}, t)$, three complementary error measures are adopted, evaluated over N test points $(\mathbf{X}_i, t_i), i = 1, \ldots, N$. The *Root-Mean-Squared Error (RMSE)* reports the average point-wise deviation in the original physical units:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(u_{\text{NN},i} - u_{\text{ref},i}\right)^2}. \tag{11}$$

RMSE weights larger outliers more heavily than absolute errors. However, it retains an intuitive interpretation as a "typical" error amplitude throughout the domain. The *Mean Absolute Error (MAE)* complements RMSE by treating all deviations linearly:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|u_{\text{NN},i} - u_{\text{ref},i}|. \tag{12}$$

Since this metric does not square the residuals, MAE is less sensitive to isolated spikes. It therefore reflects the median performance more faithfully when occasional significant errors occur, such as near singularities or boundaries. Finally, the *relative* L^2 (rel. L2) error captures the discrepancy in the energy norm commonly adopted in structural mechanics:

$$\varepsilon_{L^2} = \frac{\|u_{\text{NN}} - u_{\text{ref}}\|_2}{\|u_{\text{ref}}\|_2} = \frac{\sqrt{\sum_{i=1}^{N}(u_{\text{NN},i} - u_{\text{ref},i})^2}}{\sqrt{\sum_{i=1}^{N} u_{\text{ref},i}^2}} \tag{13}$$

This metric is dimensionless and thus directly indicates the percentage mismatch in global energy. In summary, RMSE gives a well-known "root-mean-square" magnitude, MAE supplies a robust check against the influence of outliers, and the L^2 norm links directly to energy considerations. All metrics are computed for every realization of the stochastic training procedure and then averaged over the prescribed set of random seeds. This yields an *ensemble* estimate that smooths out the variability from run to run. Reporting all three error metrics $\{\text{RMSE}, \text{MAE}, \varepsilon_{L^2}\}$ provides a balanced picture regarding the performance of the model.

3. Dynamics of beam problems

The Euler–Bernoulli and Timoshenko beam theories (Bauchau and Craig, 2009; Dolph, 1954; Oniszczuk, 2000; Goerguelue, 2009; He et al., 2017) provide the fundamental governing equations for predicting the dynamic response of beams under various loading scenarios. The dynamic behavior of such systems is described by high-order PDEs that involve complex boundary conditions and exhibit high-frequency vibration modes. These features make beam dynamics an ideal testbed for evaluating PINNs and their variants, which are known to face challenges such as spectral bias when representing high-frequency components. In this section, both Euler–Bernoulli and Timoshenko beam formulations are examined for cases of beams resting on a Winkler foundation. The proposed Causal-SPINN framework is then evaluated on these benchmarks in terms of its ability to predict forward dynamics and solve inverse problems, with particular emphasis on accurately recovering high-mode responses.

3.1. Euler–Bernoulli beam

Euler–Bernoulli beam theory offers a simplified yet effective framework for modeling the transverse vibrations of slender beams subjected to bending loads. When resting on a Winkler foundation, the beam experiences an additional linear reaction proportional to the local deflection. Here, both forward and inverse problems are considered. In the forward setting, the goal is to predict the displacement field $u(x, t)$, as well as all the derivatives of interest. In the inverse setting, sparse displacement measurements are used to recover both the displacement field and the unknown structural parameters of interest, such as the bending stiffness EI or the foundation stiffness k_w. The latter is particularly relevant for structural identification and health monitoring applications .

3.1.1. Non-dimensional formulation

The equation of motion of a prismatic Euler–Bernoulli (Fig. 3) beam of length L with bending stiffness EI, linear mass density ρA, resting on a Winkler foundation with stiffness k_w, and subjected to an external transverse load $f(x, t)$ is defined as follows:

$$\rho A u_{tt} + EI u_{xxxx} + k_w u = f(x, t) \tag{14}$$

For ML applications, a dimensionless formulation of Eq. (14) is preferred, since it enhances convergence and stability. In order to normalize Eq. (14), we define parameters U_{EB} and T_{EB} as follows:

$$T_{EB} = \sqrt{\frac{\rho A h^4}{EI}} \tag{15}$$

$$U_{EB} = \frac{h^4 f_0}{EI} \tag{16}$$

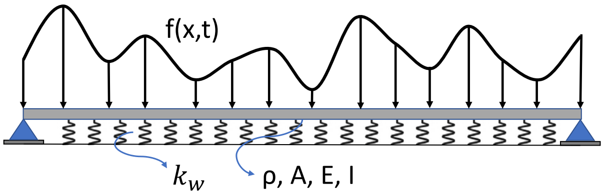


Fig. 3. Simply supported Euler–Bernoulli beam resting on a Winkler foundation.

The first parameter T_{EB} is obtained by balancing the inertia and the bending stiffness and is known as the *bending time scale* parameter, while U_{EB} is the *load-based displacement scale* parameter, obtained by balancing the load against bending. Moreover, f_0 is a characteristic amplitude of the external loading and h is a characteristic length scale, i.e., the beam's length or thickness. After the normalization, the normalized variables are obtained: $\tilde{x} = x/h$, $\tilde{t} = t/T_{EB}$, $\tilde{u} = u/U_{EB}$, $\tilde{f} = f/f_0$. Applying the chain rule, the derivatives of the solution field with respect to the dimensional coordinates can be expressed in terms of the normalized ones $\tilde{x}$, $\tilde{t}$, thus:

$$u_{tt} = \frac{U_{EB}}{T_{EB}^2}\,\tilde{u}_{\tilde{t}\tilde{t}}, \quad \text{and} \quad u_{xxxx} = \frac{U_{EB}}{h^4}\,\tilde{u}_{\tilde{x}\tilde{x}\tilde{x}\tilde{x}}. \tag{17}$$

Substituting into Eq. (14) and dividing with EIU_{EB}/h^4 gives the normalized equation of motion:

$$\left(\frac{\rho A h^4}{EIT_{EB}^2}\right) u_{tt} + u_{xxxx} + \left(\frac{k_w h^4}{EI}\right) u = f(x,t) \tag{18}$$

Note that we have dropped the tildes for clarity, and $f(x,t)$ now denotes the normalized load. Since T_{EB} is defined by balancing the stiffness and the inertia, the term $\rho A h^4/(EIT_{EB}^2)$ is equal to 1. The *dimensionless Winkler stiffness* is also defined as $k = k_w h^4/EI$. Following the above reformulation, the equation of motion reduces to:

$$u_{tt}(x,t) + u_{xxxx}(x,t) + k\ u(x,t) = f(x,t). \tag{19}$$

where $x \in [0, L/h]$.

3.1.2. Forward problem

In the forward problem, the goal is to predict the displacement field of a well-posed initial–boundary value problem. To demonstrate the performance of the proposed Causal SPINN algorithm, a benchmark Euler–Bernoulli beam is considered that is left to oscillate freely from its $16th$ mode.

Also, all material parameters are assumed as equal to one, i.e., $\rho A = 1$, $EI = 1$, so that $k = k_w h^4$. The normalized time domain is set to $\tilde{t} \in [0, 1]$ and the normalized spatial domain is set to $\tilde{x} \in [0, 16\pi]$. Following Kapoor et al. (2024a), the dimensionless foundation stiffness is chosen as $k = \pi^2 - 1$, which corresponds to $k_w = (\pi^2 - 1)/h^4$.

For a simply supported Euler–Bernoulli beam of length L, the transverse eigenmodes are given by

$$u_n(x) = \sin\left(\frac{n\pi x}{L}\right), \qquad n = 1, 2, \ldots \tag{20}$$

which satisfy the boundary conditions $u(0) = u(L) = 0$ and $u_{xx}(0) = u_{xx}(L) = 0$. Setting $L = 16\pi$ selects the sixteenth mode as $u_{16}(x) = \sin\left(\frac{16\pi x}{L}\right) = \sin(x)$. The beam is released from rest with this mode as the initial displacement and zero initial velocity:

$$u(x,0) = \sin(x), \tag{21}$$

$$u_t(x,0) = 0. \tag{22}$$

Simply supported boundary conditions are imposed such that both the deflection u and the bending moment u_{xx} are zero at the two ends. Eqs. (19)–(21) constitute a fully nondimensional, well-posed, initial–boundary value problem. The normalization practice (Eqs. (15) and (16)) ensures that all coefficients are $\mathcal{O}(1)$, yielding excellent numerical conditioning for both conventional solvers and modern machine-learning approaches such as PINNs.

Table 1
Euler–Bernoulli beam: mean error metrics (mean ± standard deviation over 10 seeds).

Method	RMSE	MAE	rel. L_2 (%)
Baseline SPINN	$(2.13 \pm 1.23) \times 10^{-3}$	$(9.26 \pm 0.38) \times 10^{-4}$	$(0.426 \pm 0.247)\%$
Causal SPINN	$(0.92 \pm 0.52) \times 10^{-3}$	$(3.81 \pm 0.15) \times 10^{-4}$	$(0.184 \pm 0.105)\%$

The analytical solution of the problem is given by:

$$u(x,t) = \sin(x)\cos(\pi t). \tag{23}$$

For structural-capacity checks, we also require the internal bending moment and shear force:

$$M(x,t) = -EI\,u_{xx}(x,t) \tag{24}$$

$$V(x,t) = -EI\,u_{xxx}(x,t) \tag{25}$$

Assuming $EI = 1$, the expressions reduce to $M = -u_{xx}$ and $V = -u_{xxx}$.

To address the Euler–Bernoulli beam problem, the proposed causal SPINN method is compared against the original Baseline SPINN formulation (Cho et al., 2023). Two "body" networks are defined as the architecture of the Neural Network. More specifically, both architectures use the modified MLP (Wang et al., 2021) with five hidden layers, each with 128 features. The loss function is defined as

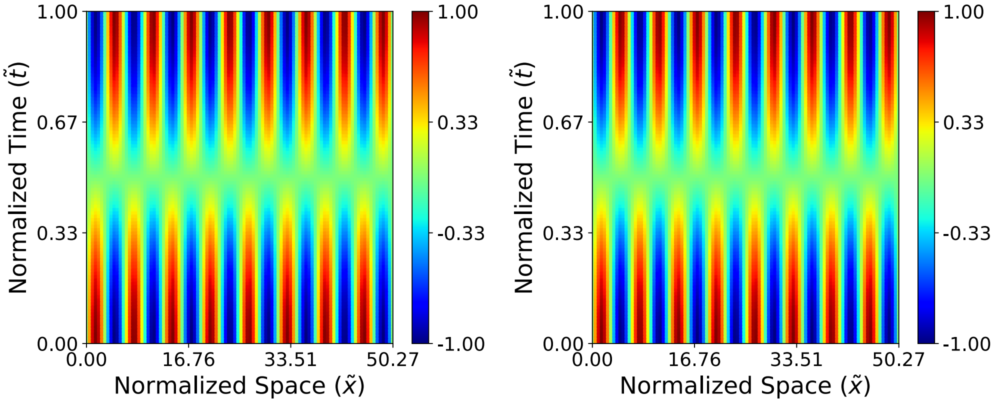
$$\mathcal{L}_{\text{SPINN}} = \lambda_{\text{res}}\,\mathcal{L}_{\text{res}} + \lambda_{\text{BC}}\,\mathcal{L}_{\text{BC}} + \lambda_{\text{IC}}\,\mathcal{L}_{\text{IC}} \tag{26}$$

For the proposed Causal SPINN method, the PDE loss term includes the loss modifications discussed in Section 2.3. The models are trained with Adam (Kingma and Ba, 2015) with a learning rate of 10^{-5} for 250000 epochs. Collocation points for PDE residuals, boundaries, and initial conditions are fixed to 128. Loss weights are set to $\lambda_{BC,IC} = 1$ for boundary/initial terms and 0.1 for the PDE residual, λ_{res}. Rank, r, of the approximated tensor is set to 128. In addition, the proposed Causal SPINN method employs residual sampling on a uniform grid with a causality weight of $\epsilon = 5$, whereas the Baseline SPINN method samples residual points from uniform random distributions. All experiments are performed using ten random seeds, and the error metrics are evaluated at the epoch with the lowest total loss.

Table 1 shows the values of the three metrics considered, RMSE (Eq. (11)), MAE (Eq. (12)), and rel. L_2 norm error (Eq. (13)). The mean and the standard deviation across all 10 seeds are reported. As can be seen, incorporating causal loss weighting reduces the relative L_2 error by more than 50%, with similar improvements in RMSE and MAE.

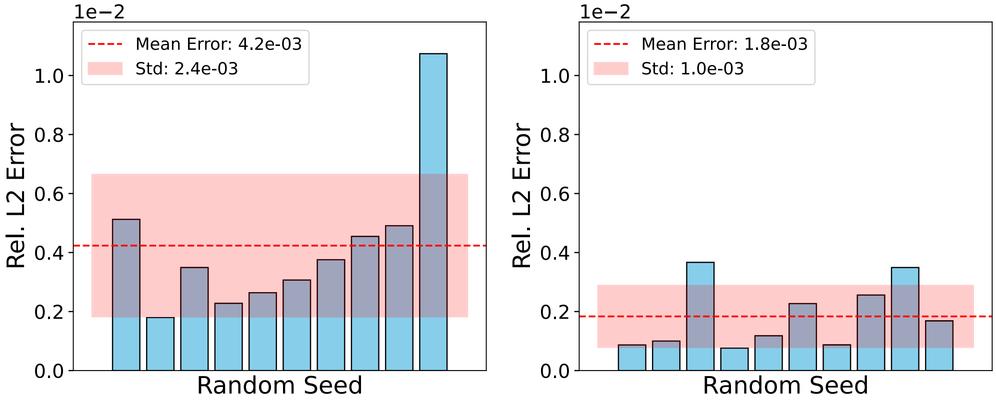
Fig. 4 presents a comparison between the exact solution and the response predicted by the proposed Causal SPINN method, while Fig. 5 reports the relative L_2 error across ten random seeds. The proposed method achieves both a lower mean error and a reduced standard deviation compared to the Baseline SPINN. It also demonstrates greater robustness, as the majority of seeds yield lower errors with reduced spread compared to the Baseline SPINN (Fig. 5a).

Moreover, Fig. 6 compares the mean RMSE distributions for the two methods across the examined seeds. The MAE heatmaps (not shown) exhibit the same spatial trends. For the Baseline SPINN (Fig. 6a), errors form a combined pattern of vertical bands at modal peaks ($x = k\pi$) and a clear horizontal gradient that intensifies at later time steps ($t > 0.5$). This is consistent with temporal error accumulation: without causal weighting, the optimizer distributes effort uniformly across all time steps, allowing early-time errors to propagate and grow, producing the most severe hotspots where late-time and high-curvature regions intersect. In contrast, the Causal SPINN (Fig. 6b) exhibits a substantially reduced and more uniform error profile. The temporal gradient visible in the Baseline SPINN is suppressed, confirming that causal weighting



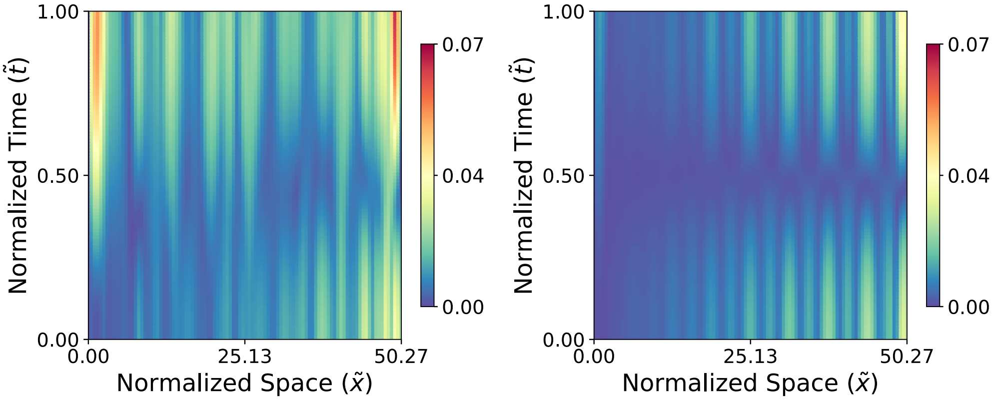
(a) **(b)**

Fig. 4. Euler–Bernoulli beam: comparison of deflection fields. (a) Exact solution (ground truth); (b) causal SPINN prediction.



(a) **(b)**

Fig. 5. Euler–Bernoulli beam: seed-wise relative L_2 error. (a) Baseline SPINN; (b) Causal SPINN.



(a) **(b)**

Fig. 6. Euler–Bernoulli beam: RMSE map. (a) Baseline SPINN; (b) Causal SPINN.

prevents temporal accumulation by ensuring that the network first resolves early-time dynamics correctly before progressing to later times. Residual vertical banding at modal peaks persists at reduced amplitude, but the RMSE heatmap is markedly smoother. This spatial error pattern is directly relevant for engineering applications, as the error hotspots in the Baseline SPINN coincide with regions of maximum curvature, where bending moment predictions ($M = EI \cdot u_{xx}$) are most critical for structural assessment.

For engineering design, accurately evaluating higher spatial derivatives is essential. To this end, Table 2 reports the relative L_2 errors for the first four derivatives of the displacement $u(x,t)$. Both methods achieve an error of less than 5% for the first two derivatives, but the accuracy deteriorates for the third and fourth derivatives. This behavior reflects the compounding effect of automatic differentiation, whereby each successive differentiation tends to amplify noise and approximation errors.

Table 2
Euler–Bernoulli beam: relative L_2 error in derivatives (%) (mean ± standard deviation over 10 seeds).

Method	u_x	u_{xx}	u_{xxx}	u_{xxxx}
Baseline SPINN	$(1.03 \pm 0.77)\%$	$(5.00 \pm 4.73)\%$	$(17.3 \pm 12.1)\%$	$(78.2 \pm 75.8)\%$
Causal SPINN	$(0.46 \pm 0.29)\%$	$(2.35 \pm 1.78)\%$	$(21.0 \pm 16.2)\%$	$(91.5 \pm 47.0)\%$

Table 3
Euler–Bernoulli beam: sensitivity of Causal SPINN to rank r and causality parameter ε. Relative L_2 error (mean ± std over 5 seeds). Bold denotes best ε per rank.

r	$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$	$\varepsilon = 7$
32	1.05e-3 ± 8.39e-4	8.18e-4 ± 1.88e-4	**7.08e-4 ± 8.63e-5**	1.74e-3 ± 1.32e-3
64	7.28e-4 ± 1.45e-4	7.20e-4 ± 1.75e-4	**6.84e-4 ± 1.46e-4**	8.83e-4 ± 1.80e-4
128	1.10e-3 ± 2.99e-4	**9.43e-4 ± 1.82e-4**	1.20e-3 ± 3.22e-4	1.17e-3 ± 5.47e-4
256	1.04e-3 ± 4.74e-4	1.47e-3 ± 5.83e-4	**9.16e-4 ± 1.32e-4**	1.25e-3 ± 1.32e-4

Therefore, while the displacements u and the slopes u_x can be recovered with high precision, and the curvature (u_{xx}) remains within acceptable bounds, the fourth derivative exhibits errors that are too large for practical use. This confirms that without sophisticated architectures or regularization, high-order derivatives tend to be unstable and dominated by propagated error.

In summary, this numerical experiment shows that for applications that focus solely on deflection profiles, the Causal SPINN outperforms Baseline SPINN by reducing errors in u, u_x, and u_{xx}. However, many engineering design and inverse problems require the accurate prediction of higher-order response quantities, which depend on the second, third, and fourth derivatives of the governing equations. The error amplification observed in the fourth derivative underscores the need for dedicated strategies to stabilize high-order derivative evaluation and to enable reliable estimation of structural parameters such as the bending stiffness EI.

3.1.3. Sensitivity to rank r and causality parameter ε

To assess the robustness of the proposed Causal SPINN method with respect to its two key hyperparameters, a joint sensitivity analysis is performed over rank $r \in \{32, 64, 128, 256\}$ and causality parameter $\varepsilon \in \{1, 3, 5, 7\}$, with 5 random seeds per configuration. Table 3 reports the relative L^2 error (mean ± standard deviation). For reference, the Baseline SPINN at $r = 128$ achieves $(4.26 \pm 2.47) \times 10^{-3}$ (10 seeds).

All Causal SPINN configurations outperform the Baseline SPINN, with the best configuration ($r = 64$, $\varepsilon = 5$) achieving over 6× lower error. Moderate values of $\varepsilon \in [3, 5]$ consistently yield the lowest error and variance across all tested ranks, while extreme values ($\varepsilon = 1$ or $\varepsilon = 7$) tend to increase both. Accuracy is stable across ranks $r = 32$–256: the best error per rank varies by a factor of only 1.4×. This stability arises because the benchmark solution ($u = \sin(x)\cos(\pi t)$) is rank-1 separable, so moderate ranks suffice. The rank-1 separability of the benchmark solutions is a consequence of the manufactured solution approach used for closed-form validation, which is standard practice in the PINN literature (Cho et al., 2023; Wang et al., 2024; Mattey and Ghosh, 2022). However, real engineering systems with geometric or material nonlinearity, parametric excitation, or complex loading produce solutions with limited separability, where higher rank r provides additional capacity to approximate more complex spatiotemporal interactions. Since the effective rank is problem-dependent and not known *a priori*, $r = 128$ is adopted as a conservative default that balances expressiveness and computational cost, consistent with Cho et al. (2023) who demonstrated that rank higher than 128 provides diminishing returns in terms of accuracy. Runtime is nearly rank-independent: all configurations complete in approximately 823 ± 5 s per seed (250k epochs), confirming that computational cost is dominated by automatic differentiation, not the separable tensor product.

Table 4
Euler–Bernoulli beam: error metrics for primary displacement u with auxiliary variable (mean ± standard deviation over 10 seeds).

Method	RMSE	MAE	rel.L_2 (%)
Baseline SPINN (aux.)	$(2.38 \pm 1.14) \times 10^{-3}$	$(1.17 \pm 0.40) \times 10^{-3}$	0.475 ± 0.228
Causal SPINN (aux.)	$(1.68 \pm 0.64) \times 10^{-3}$	$(8.36 \pm 2.37) \times 10^{-4}$	0.335 ± 0.127

Regarding transferability, the same moderate range $\varepsilon \in [3, 5]$ performs well across all four benchmark problems despite different temporal characteristics and PDE orders. This insensitivity arises because the causal weights are self-regulating: the cumulative residual $\sum_{k<i} \mathcal{L}_k$ decreases as training progresses, automatically widening the causal window regardless of the specific ε value. For chaotic or multi-scale problems, the annealing strategy of Wang et al. (2024) is recommended.

3.1.4. Use of auxiliary variables

To enhance the capacity of the proposed Causal SPINN algorithm on predicting high-order derivatives, we introduce an auxiliary output field $v(x,t)$, trained alongside $u(x,t)$. This field, in the given problem, explicitly represents curvature by approximating the second derivative: $v(x,t) = u_{xx}(x,t)$. By learning u and v simultaneously, the network acquires the capacity to capture the curvature as a separate field rather than relying solely on successive differentiations of the displacement field u. In practice, to enforce consistency between v and the true second derivative, we augment the forward loss defined in Eq. (26) with a penalty term:

$$\mathcal{L}_v = \frac{1}{N_{\text{cons}}} \sum_{i=1}^{N_{\text{cons}}} \left(u_{xx}(x_i, t_i) - v(x_i, t_i) \right)^2 \tag{27}$$

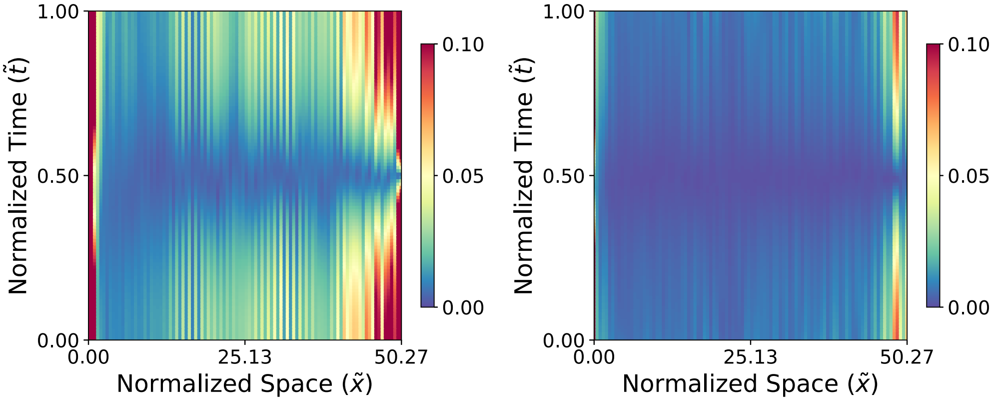
And the loss function becomes:

$$\mathcal{L}_{\text{SPINN,AUX}} = \lambda_{\text{res}} \mathcal{L}_{\text{res}} + \lambda_v \mathcal{L}_v + \lambda_{\text{bc}} \mathcal{L}_{\text{BC}} + \lambda_{\text{ic}} \mathcal{L}_{\text{IC}} \tag{28}$$

The auxiliary-loss weight is found, through a grid search, to yield the best results when set to $\lambda_v = 10$ for Causal SPINN and $\lambda_v = 1$ for Baseline SPINN with auxiliary variables. It is also found that reducing the learning rate to 10^{-6} improves the convergence. All other training settings, e.g., network architecture, optimizer, r, epochs, and collocation point sampling, remain unchanged.

Table 4 shows the error metrics for the displacement u. Causal SPINN with auxiliary variables reduces both the mean and standard deviation of the relative L_2 error by about 30% compared to the Baseline Aux SPINN, while also reducing RMSE and MAE. Table 5 shows that all spatial derivatives, including the fourth-order term u_{xxxx}, achieve errors below 5%, with the Causal SPINN runs exhibiting consistently lower variability. This confirms that the auxiliary-variable formulation effectively suppresses error propagation from automatic differentiation, making high-order derivative recovery feasible.

Fig. 7 visualizes the pointwise RMSE of u_{xxxx} across the normalized spatiotemporal domain. In the Causal SPINN case without auxiliary variables (Fig. 7(a)), elevated errors spread over a large portion of the domain, with high concentrations near both spatial boundaries. By contrast, the Causal Aux SPINN (Fig. 6(b)) confines significant errors almost exclusively to thin boundary strips at $\tilde{x} = 0$ and $\tilde{x} = 50.27$, while the domain interior remains at near-zero error levels. The key observation is that the auxiliary-variable formulation, combined with causal training, effectively reduces error propagation into the domain interior—a failure mode clearly visible in the non-auxiliary case. The third derivative u_{xxx}, corresponding to shear force, exhibited similar but slightly less pronounced patterns, consistent with Table 5. This is particularly relevant from an engineering standpoint, as accurate shear force recovery is essential for reliable strength verification and capacity checks in structural assessment.



(a) **(b)**

Fig. 7. Euler–Bernoulli beam: RMSE map of u_{xxxx}. (a) Causal SPINN; (b) Causal Aux SPINN.

Table 5
Euler–Bernoulli beam: relative L_2 error in derivatives with auxiliary variable (%) (mean ± standard deviation over 10 seeds).

Method	u_x	u_{xx}	u_{xxx}	u_{xxxx}
Baseline SPINN (aux.)	$1.13 \pm 0.52\%$	$1.85 \pm 0.58\%$	$2.39 \pm 0.67\%$	$4.42 \pm 2.63\%$
Causal SPINN (aux.)	$0.82 \pm 0.30\%$	$1.76 \pm 0.54\%$	$2.25 \pm 0.56\%$	$4.20 \pm 1.44\%$

Table 6
Euler–Bernoulli beam: sensitivity of Causal Aux SPINN to the auxiliary-loss weight λ_v. Relative L_2 error (mean ± std over 5 seeds).

λ_v	Mean rel. L_2 error	Std	CoV (%)
0.1	2.28e-2 ± 1.40e-2	1.40e-2	61.4
1.0	4.29e-3 ± 1.39e-3	1.39e-3	32.4
10.0	**3.37e-3 ± 6.41e-4**	**6.41e-4**	**19.0**
50.0	4.76e-3 ± 1.07e-3	1.07e-3	22.5

3.1.5. *Sensitivity to auxiliary-loss weight λ_v*

To validate the choice of λ_v for the Causal Aux SPINN, a sweep over $\lambda_v \in \{0.1, 1, 10, 50\}$ is performed with $r = 128$, $\epsilon = 5$, and $\mathrm{lr} = 10^{-6}$ fixed (5 seeds each). Table 6 reports the results.

The ablation confirms that $\lambda_v = 10$ achieves both the lowest mean error and the lowest coefficient of variation (19.0%). Too small a value ($\lambda_v = 0.1$) renders the consistency constraint ineffective, resulting in 6.7× worse mean error with a CoV of 61.4%. Too large a value ($\lambda_v = 50$) overweights the auxiliary constraint at the expense of the PDE residual, leading to 41% higher error than the optimal. The optimal λ_v balances two objectives: enforcing the kinematic relation $u_{xx} \approx v$ while leaving the optimizer sufficient freedom to minimize the PDE residual.

3.1.6. *Comparison with PINN methods*

To isolate the contribution of each component in the proposed framework, three Baseline PINN variants are evaluated on the same Euler–Bernoulli beam benchmark:

- *Causal PINN* (Wang et al., 2024), is a standard fully-connected PINN with the same causal weighting ($\epsilon = 5$). It uses a single MLP that takes concatenated (x, t) inputs, with 4 hidden layers of 128 features. Collocation points are sampled on a full $128^2 = 16{,}384$ grid.
- *AUX PINN* (Yuan et al., 2022), is a standard PINN with two outputs (u, v) where $v \approx u_{xx}$, and $\lambda_v = 10$. No causal weighting is applied. The auxiliary decomposition reduces the maximum AD order from 4th to 2nd.
- *Adaptive-domain PINN* (Mattey and Ghosh, 2022), refers to a standard PINN with progressive time-domain expansion over $N = 10$ stages, each covering a growing fraction of $[0, 1]$. At every stage transition, early stopping (patience=3 000) and learning rate scheduling are reset.

Table 7
Euler–Bernoulli beam: comparison of SPINN and PINN methods. SPINN results are mean ± std over 10 seeds; PINN results are single seed.

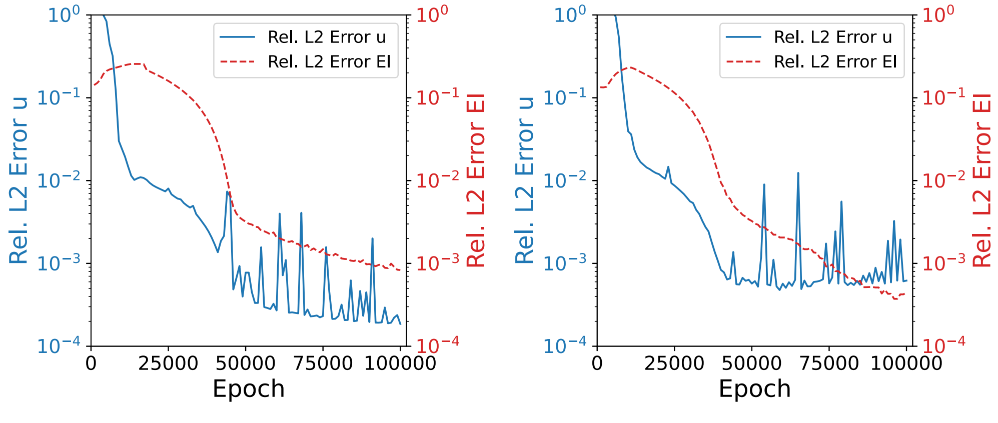
Method	rel. L_2 error	Epochs	Runtime (s)	ms/iter	GPU (MiB)
Baseline SPINN	(4.26 ± 2.47)e-3	250k	899	3.59	330
Causal SPINN	(1.84 ± 1.05)e-3	250k	761	3.04	394
Causal Aux SPINN	(3.35 ± 1.27)e-3	250k	822	3.29	308
AUX PINN	3.69e-3	250k	4629	18.5	1328
Causal PINN	6.83e-1	∼20k	738	36.4	3248
Adaptive-domain PINN	7.29e-3	∼206k	7402	35.9	3248

All PINN variants use matching network depth and width (4 layers × 128 features) and undergo a hyperparameter sweep over $\lambda_{\text{res}} \in \{0.01, 0.1, 1, 10\}$ at 10k epochs, with the best value selected for full training. These comparisons isolate the benefit of separability (Causal PINN vs. Causal SPINN), of causality combined with separability (AUX PINN vs. Causal Aux SPINN), and of explicit temporal marching vs. causal weighting (Adaptive PINN vs. Causal SPINN). Table 7 reports accuracy, training time, and GPU memory for all methods.

As shown in Table 7, Causal PINN without auxiliary decomposition fails entirely on this problem (rel. $L_2 = 0.683$) as it diverges, whereas AUX PINN achieves reasonable accuracy (3.69×10^{-3}). This provides independent evidence that auxiliary variable decomposition is valuable for 4th-order PDEs regardless of the network architecture. Adaptive-domain PINN also achieves reasonable accuracy but with much higher runtime. The separable architecture provides a further advantage: Causal SPINN outperforms the best PINN variant (AUX PINN) by 2× in accuracy while being 5.6× faster and using 3.4× less GPU memory, since PINNs evaluate the full 128^2 collocation grid per iteration whereas SPINN evaluates only $128 + 128$ separable points. Causal Aux SPINN achieves slightly higher displacement error than Causal SPINN ($(3.35 \pm 1.27) \times 10^{-3}$ vs. $(1.84 \pm 1.05) \times 10^{-3}$). We attribute this to the additional auxiliary output head sharing expressive capacity with the primary field, marginally reducing displacement accuracy while substantially improving the prediction of derived quantities (M, V) as exhibited in Section 3.1.4. Regarding computational resources, SPINN memory is nearly rank-independent whereas PINN memory is dominated by the collocation grid and AD graph. AUX PINN requires less memory than Causal/Adaptive PINN, reflecting the shallower AD graph from auxiliary decomposition.

3.1.7. *Inverse parameter estimation*

Inverse identification of structural parameters from sparse measurements is a crucial task in structural health monitoring and digital-twin



(a) **(b)**

Fig. 8. Euler–Bernoulli beam: training curves for inverse bending stiffness estimation. (a) Causal SPINN; (b) Causal SPINN with auxiliary variable.

updating. The ability of the proposed framework to recover the uniform bending stiffness EI of a supported beam solely from limited spatiotemporal displacement data is examined here. For this purpose, EI is treated as an additional trainable scalar parameter, initialized uniformly in the range $[0.5, 1.5]$ to reflect typical uncertainty in engineering problems.

Synthetic "sensor" time series are generated from the exact solution of Eq. (23) at four fixed locations along the beam. These positions are:

$$x_i = \{3.75\pi,\ 7.5\pi,\ 11.25\pi,\ 15\pi\}, \quad i = 1, \ldots, 4$$

each recording $u(x_i, t)$ has a sampling frequency of 100 Hz and total duration 1 s Both the Baseline Causal SPINN and the Causal SPINN with auxiliary variables, as introduced in Sections 3.1 and 3.1.4, are evaluated.

For the inverse problem, the total loss function contains an additional data-fitting term and is defined as:

$$\mathcal{L}_{\text{SPINN,Inverse}} = \lambda_{\text{res}}\,\mathcal{L}_{\text{res}} + \lambda_v\,\mathcal{L}_v + \mathcal{L}_{\text{BC}} + \mathcal{L}_{\text{IC}} + \lambda_{\text{data}}\,\mathcal{L}_{\text{data}} \tag{29}$$

Where the L_v term is used only with the auxiliary variable enhanced architecture. The data-fitting term is defined as:

$$\mathcal{L}_{\text{data}} = \frac{1}{T}\sum_{i=1}^{4}\sum_{j=1}^{T}\left(u_{\text{NN}}(x_i, t_j) - u_{\text{exact}}(x_i, t_j)\right)^2 \tag{30}$$

For the auxiliary-variable model, the auxiliary loss term follows Eq. (27). A small grid search gives the regularization terms that yielded the best results, $\lambda_{\text{data}} = 1$ and $\lambda_v = 10$. The network architecture, optimizer, and training schedule were kept the same as in Sections 3.1 and 3.1.4, while the learning rate is set to 10^{-5} for both architectures.

Fig. 8 shows, for both methods, the epoch-by-epoch evolution of: (i) the relative L^2 error in the displacement, and (ii) The relative error in the inferred bending stiffness. Both models reduce the displacement error to below 1% within a few thousand epochs, while also accurately identifying the bending stiffness of the beam. The auxiliary-variable model finally achieves a reduction in the final parameter error almost by half, i.e. 4.4×10^{-2}% compared to 8.3×10^{-2}% produced by the Baseline Causal SPINN. In all, both architectures rapidly recover the beam displacement and bending stiffness with high accuracy. However, incorporating the auxiliary curvature variable into the Causal framework yields a more precise estimate of EI. This demonstrates the benefit of auxiliary-variable regularization in stabilizing and improving inverse parameter learning.

3.2. Timoshenko beam

Timoshenko beam theory extends the classical Euler–Bernoulli formulation by incorporating both shear deformation and rotary inertia effects, making it well-suited for beams with low length-to-thickness ratios. In addition to the transverse displacement field $u(x, t)$, the theory considers the cross-sectional rotation $\theta(x, t)$ as an independent unknown, yielding a coupled system of second-order PDEs. This coupling increases both the dimensionality and the stiffness of the problem, since the shear term $\kappa GA(\ theta - u_x)$ must be determined accurately and alongside the bending effects.

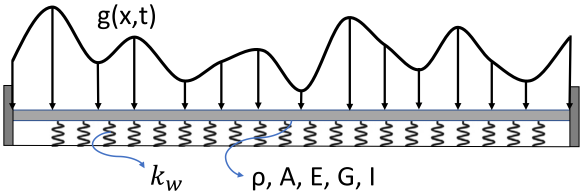


Fig. 9. Fixed Timoshenko beam on a Winkler foundation.

3.2.1. Non-dimensional formulation

The equation of motion of a prismatic Timoshenko beam of length L, with bending stiffness EI, and shear modulus G is given by a pair of coupled PDEs. Assuming that κ is the shear-correction factor, these equations are:

$$\begin{aligned} &\rho I\,\theta_{tt} - EI\,\theta_{xx} + \kappa GA\big(\theta - u_x\big) = 0, \\ &\rho A\,u_{tt} + \big[\kappa GA\big(\theta - u_x\big)\big]_x + k_w\,u = g(x,t) \end{aligned} \tag{31}$$

where $g(x, t)$ is an external transverse load (Fig. 9). The remaining parameters have already been defined for the EB problem. Unlike the Euler–Bernoulli formulation, the equation of motion of the Timoshenko beam involves derivatives of at most second order. Therefore, only derivatives up to second order need to be approximated, thus, there is no need to introduce an auxiliary variable into the network architecture, since, as shown in the EB example, Causal SPINN can accurately approximate second-order derivatives .

The problem can become nondimensional by balancing inertia with shear stiffness and the external load with the Winkler foundation stiffness. The resulting *bending time scale* parameter takes the form:

$$T_{TB} = \sqrt{\frac{\rho A\,h^2}{\kappa GA}} \tag{32}$$

The characteristic displacement scale is obtained by balancing the distributed load g_0 with the shear restoring force $\kappa GA\,u/h$.

$$U_{TB} = \frac{h\,g_0}{\kappa GA} \tag{33}$$

Additionally, the *associated rotation scale* is:

$$\Theta_{TB} = \frac{U_{TB}}{h} = \frac{g_0}{\kappa GA} \tag{34}$$

The following dimensionless quantities are obtained $\tilde{x} = x/h$, $\tilde{t} = t/T$, $\tilde{u} = u/U$, $\tilde{\theta} = \theta/\Theta$, $\tilde{g} = g/g_0$. Applying the chain rule, we arrive at:

$$\begin{aligned} u_{tt} &= \frac{U_{TB}}{T_{TB}^2}\,\tilde{u}_{\tilde{t}\tilde{t}}, u_x = \frac{U_{TB}}{h}\,\tilde{u}_{\tilde{x}}, \quad u_{xx} = \frac{U_{TB}}{h^2}\,\tilde{u}_{\tilde{x}\tilde{x}}, \\ \theta_{tt} &= \frac{\Theta_{TB}}{T_{TB}^2}\,\tilde{\theta}_{\tilde{t}\tilde{t}}, \theta_x = \frac{\Theta_{TB}}{h}\,\tilde{\theta}_{\tilde{x}}, \quad \theta_{xx} = \frac{\Theta_{TB}}{h^2}\,\tilde{\theta}_{\tilde{x}\tilde{x}} \end{aligned} \tag{35}$$

Inserting the above expressions into Eq. (31), the first row is divided by $\kappa GA\,\Theta$ and the second row by $\kappa GA\,\Theta/h$, yielding

$$\left(\frac{\rho I}{\rho A\,h^2}\right)\theta_{tt} - \left(\frac{EI}{\kappa GA\,h^2}\right)\theta_{xx} + \left(\theta - u_x\right) = 0, \tag{36}$$

$$u_{tt} + \left(\theta - u_x\right)_x + \left(\frac{k_w\,h^2}{\kappa GA}\right)u = g(x,t) \tag{37}$$

For clarity, the tildes have been omitted. Introducing the dimensionless groups

$$\alpha = \frac{EI}{\kappa GA\,h^2}, \qquad \beta = \frac{\rho I}{\rho A\,h^2}, \qquad k = \frac{k_w\,h^2}{\kappa GA} \tag{38}$$

The fully nondimensional Timoshenko–Winkler system is obtained as

$$\begin{aligned} &\beta\,\theta_{tt} - \alpha\,\theta_{xx} + \left(\theta - u_x\right) = 0 \\ &u_{tt} + \left(\theta - u_x\right)_x + k\,u = g(x,t) \end{aligned} \tag{39}$$

where $x \in [0, L/h]$, $t \geq 0$. In the equations above, α represents the bending-to-shear stiffness ratio, β the rotary-to-translational inertia ratio, and k the dimensionless Winkler stiffness.

3.2.2. *Forward problem*

To examine the network's ability to capture high-frequency spatial content and coupled fields, we examine the Timoshenko beam in its eleventh vibration mode. A high-frequency mode was chosen to highlight a known limitation of PINNs, the *spectral bias*, where Neural Networks tend to under-represent high-frequency components. The performance of the Baseline SPINN formulation against our causality-weighted extension is compared, without the use of an auxiliary variable, assessing not only the accuracy of the primary fields $u(x,t)$ and $\theta(x,t)$ but also the accuracy of the derived quantities such as bending moment and shear force under dynamic loading.

A benchmark problem is considered, where the normalized length of the spatial domain is $\tilde{x} = L/h = 11\pi$, and we solve over $\tilde{t} \in [0,1]$. Material and geometric parameters are chosen so that $\rho A = \kappa GA = 1$ and $\alpha = \beta = 1$. The Winkler spring stiffness parameter is set to $k = 1$, so that $k_w = 1/h^2$, and the transverse load is $g(x,t) = \cos(t)$. The beam is fully fixed at both ends, allowing for investigating different boundary conditions compared to the EB example that was previously examined. The beam is deformed from rest by applying a transverse displacement profile proportional to the 11th mode, as given by Eq. (20). Rotation θ is given by $\theta(x,t) = [u'(x,t) + x - \frac{11\pi}{2}]cos(t)$.

The initial conditions at $t = 0$ are

$$u(x,0) = \frac{11\pi}{2}\sin x, \quad u_t(x,0) = 0, \quad \theta(x,0) = \left[\tfrac{11\pi}{2}\cos(x) + x - \tfrac{11\pi}{2}\right], \quad \theta_t(x,0) = 0. \tag{40}$$

The analytical solution for this benchmark is:

$$u(x,t) = \frac{11\pi}{2}\sin(x)\,\cos(t), \tag{41}$$

$$\theta(x,t) = \left[\tfrac{11\pi}{2}\cos(x) + x - \tfrac{11\pi}{2}\right]\cos(t), \tag{42}$$

Moreover, the bending moment and the shear force are obtained as:

$$M(x,t) = \theta_x(x,t) \tag{43}$$

$$V(x,t) = \theta(x,t) - u_x(x,t) \tag{44}$$

which are already non-dimensional since $EI = 1$ and $\kappa GAh^2 = 1$. Eq. (39), together with the boundary and initial data (Eq. (40)), form a fully non-dimensional well-posed initial–boundary value problem where all coefficients are equal to one.

Table 8

Timoshenko beam: mean error metrics for deflection u (mean ± standard deviation over 10 seeds).

Method	RMSE	MAE	rel. L_2 (%)
Baseline SPINN	$(6.96 \pm 2.92) \times 10^{-3}$	$(3.67 \pm 1.38) \times 10^{-3}$	0.0672 ± 0.0282
Causal SPINN	$(2.17 \pm 1.19) \times 10^{-3}$	$(1.39 \pm 0.42) \times 10^{-3}$	0.0210 ± 0.0115

Table 9

Timoshenko beam: mean error metrics for rotation θ (mean ± standard deviation over 10 seeds).

Method	RMSE	MAE	rel. L_2 (%)
Baseline SPINN	$(1.17 \pm 0.99) \times 10^{-2}$	$(4.50 \pm 2.35) \times 10^{-3}$	0.0877 ± 0.0747
Causal SPINN	$(2.30 \pm 0.93) \times 10^{-3}$	$(1.33 \pm 0.41) \times 10^{-3}$	0.0173 ± 0.0070

Table 10

Timoshenko beam: relative L_2 errors in derivatives of u (%) (mean ± standard deviation over 10 seeds).

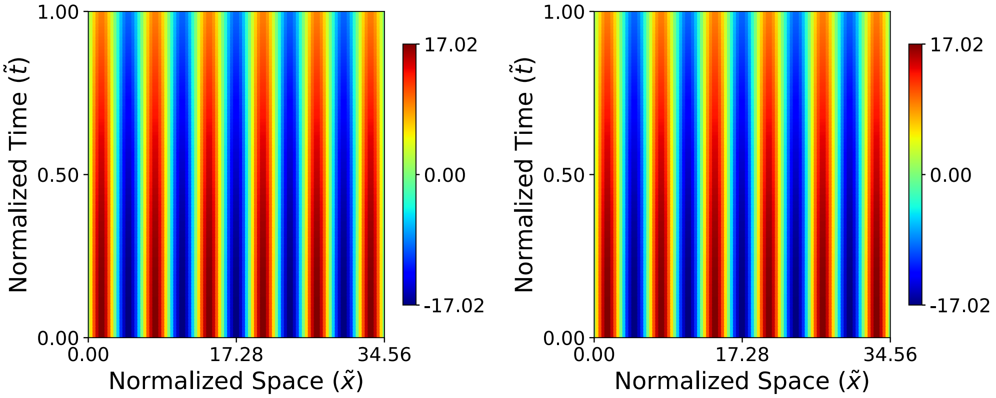
Method	u_x (%)	u_{xx} (%)
Baseline SPINN	0.230 ± 0.097	1.78 ± 0.93
Causal SPINN	0.137 ± 0.052	2.07 ± 0.79

Both architectures employ two "body" networks with joint output (u, θ) (Cho et al., 2023). These consist of modified MLPs (Wang et al., 2021) with four hidden layers of 128 units. Rank of the approximated tensor, r, is 128 as well. The training employs the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-5} for 150000 epochs and 128 collocation points for the PDE residual, the boundary, and the initial conditions. In Causal SPINN, all loss weights are set to $\lambda = 1$ and the Causality parameter is $\varepsilon = 3$. For comparison with the Baseline SPINN, the PDE residual weight is set to 0.1. The loss function follows Eq. (26), with $\mathcal{L}_{\text{res}}$ now including both coupled PDEs.

The residual points are sampled on an equally spaced grid for the Causal SPINN and on a uniformly random grid for the Baseline SPINN method. Ten random-seed experiments are performed, and the metrics reported correspond to the epoch yielding the minimum total loss. Figs. 10 and 11 illustrate the predicted and actual values of the displacements and rotations, respectively.

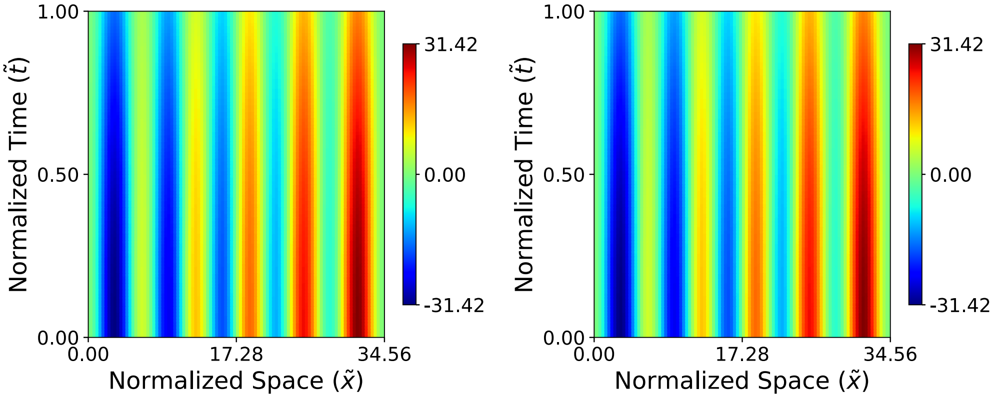
Tables 8 and 9 summarize the mean (± std) error metrics for $u(x,t)$ and $\theta(x,t)$. Causal SPINN reduces the relative L_2 error of the deflection by more than 60%, while for the rotation, the improvement is nearly an order of magnitude. Figs. 12 and 13 show the seed-wise dispersion, confirming the reduced mean error and the reduced error variability under Causal weighting.

Fig. 14 presents the spatio-temporal RMSE heatmap for the deflection $u(x,t)$. The MAE heatmaps (not shown) exhibit the same spatial trends. For the Baseline SPINN (Fig. 14a), the error field exhibits distinct vertical bands aligned with the spatial modal peaks of the 11th vibration mode. Similarly to the Euler–Bernoulli case, a clear temporal gradient is visible, with errors intensifying at later time steps, consistent with temporal error accumulation in the absence of causal weighting. In contrast, the Causal SPINN (Fig. 14b) produces a nearly uniformly low error field, with both the vertical band structure and the temporal gradient effectively suppressed. This error pattern is directly relevant for engineering applications: the modal-peak error concentrations in the Baseline SPINN coincide with regions of high curvature, where bending moment ($M = \theta_x$) and shear force ($V = \theta - u_x$) are largest, making accurate predictions at these locations critical for structural assessment.



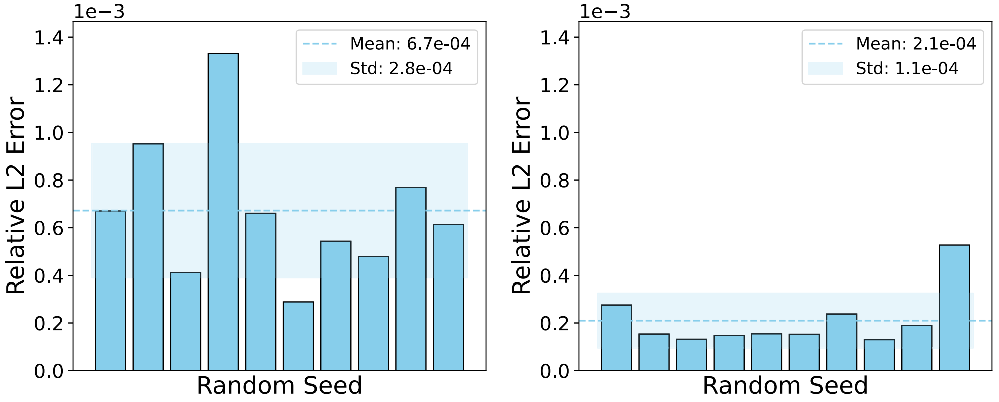
(a) **(b)**

Fig. 10. Timoshenko beam: deflection heatmaps. (a) Exact solution; (b) Causal SPINN.



(a) **(b)**

Fig. 11. Timoshenko beam: rotation heatmaps. (a) Exact solution; (b) Causal SPINN.



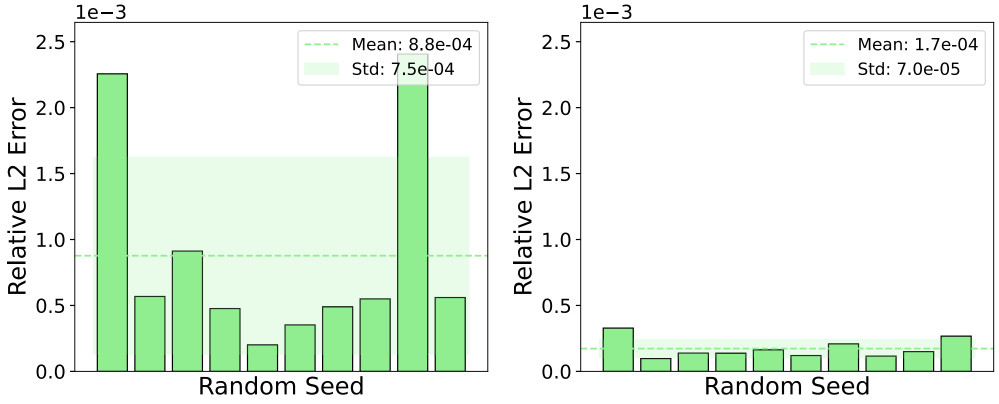
(a) **(b)**

Fig. 12. Timoshenko beam: seed-wise relative L_2 error for u. (a) Baseline SPINN; (b) Causal SPINN.

Table 11
Timoshenko beam: relative L_2 errors in derivatives of θ (%) (mean ± standard deviation over 10 seeds).

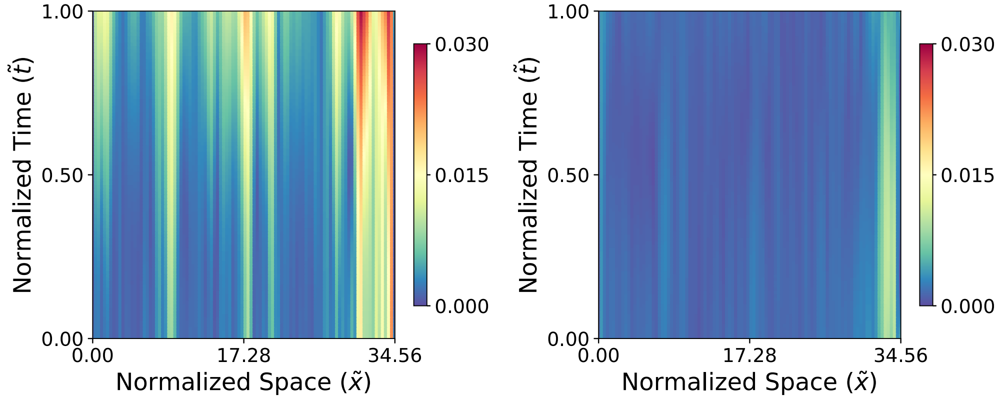
Method	θ_x (%)	θ_{xx} (%)
Baseline SPINN	0.542 ± 0.519	3.91 ± 3.79
Causal SPINN	0.216 ± 0.111	3.90 ± 1.09

Moreover, Fig. 15 shows the RMSE map of the rotation $\theta(x,t)$. The Baseline SPINN map (Fig. 15a) exhibits pronounced vertical error bands with peak RMSE values reaching approximately 0.12, concentrated near the left support ($x \approx 0$) and at several interior spatial positions. In contrast, the Causal SPINN map (Fig. 15b) is nearly uniformly dark, with these error bands effectively suppressed across the entire domain. The plots show that causal weighting not only reduces average errors but also suppresses localized spatial deviations in both primary fields.



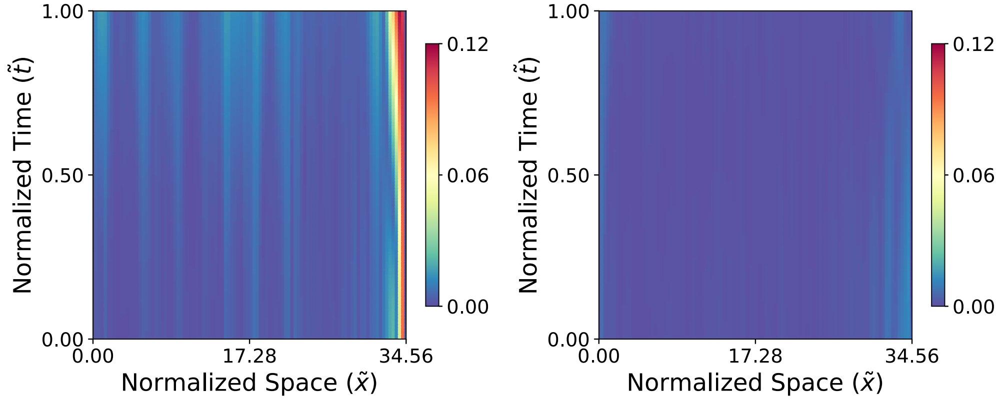
(a) **(b)**

Fig. 13. Timoshenko beam: seed-wise relative L_2 error for θ. (a) Baseline SPINN; (b) Causal SPINN.



(a) **(b)**

Fig. 14. Timoshenko beam: RMSE map for $u(x,t)$. (a) Baseline SPINN; (b) Causal SPINN.



(a) **(b)**

Fig. 15. Timoshenko beam: RMSE map for $\theta(x,t)$: (*a*) Baseline SPINN, (*b*) Causal SPINN.

Derivative accuracy is quantified in Tables 10 and 11, where first and second derivative errors remain below 2.5% for Causal SPINN, which is adequate for reliably estimating the bending moment and the shear force without introducing auxiliary variables. The MAE heatmaps (not shown) exhibit the same spatial trends.

In summary, for the Timoshenko beam example considered, Causal SPINN substantially improves both displacement and rotation accuracy relative to Baseline SPINN, while keeping first- and second-derivative errors below 2.5%. The high accuracy of these derivatives makes additional auxiliary-variable regularization unnecessary for Timoshenko beam problems.

3.2.3. Comparison with PINN methods

The Timoshenko system involves coupled PDEs with only second-order derivatives. This benchmark therefore tests whether the proposed framework provides gains for coupled PDEs when PINNs do not face

Table 12
Timoshenko beam: comparison of SPINN and PINN methods. SPINN results are mean ± std over 10 seeds; PINN results are single seed.

Method	u rel. L_2	θ rel. L_2	Epochs	Runtime (s)	ms/iter	GPU (MiB)
Baseline SPINN	(6.72 ± 2.82)e-4	(8.77 ± 7.47)e-4	150k	431	2.88	214
Causal SPINN	(2.10 ± 1.15)e-4	(1.73 ± 0.70)e-4	150k	425	2.83	310
Causal PINN	2.33e-2	2.18e-2	~85k	1663	19.5	1362
Adaptive-domain PINN	3.27e-3	1.49e-3	~126k	2443	19.4	1458

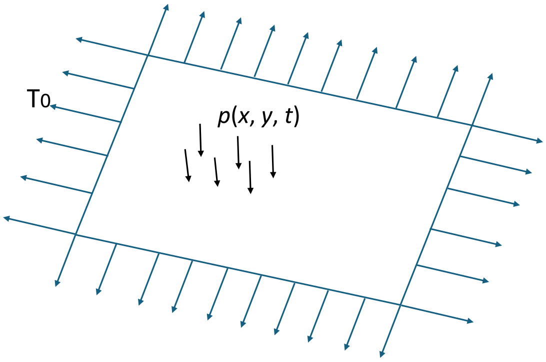


Fig. 16. Simply supported Membrane.

the high-order differentiation barrier encountered in the EB beam. Two PINN variants are evaluated, Causal PINN (Wang et al., 2024) ($\varepsilon = 3$) and Adaptive-domain PINN (Mattey and Ghosh, 2022), both following the same architecture and λ_{res} sweep protocol as in Section 3.1.6. No AUX PINN variant is needed since the system involves only second-order derivatives. Table 12 reports accuracy and training cost for all methods.

Even on this lower-order system where PINNs can converge, Causal SPINN outperforms the best PINN variant (Adaptive-domain PINN) by 16× in relative L_2 error for u, while being 5.7× faster. Adaptive-domain PINN outperforms Causal PINN by 7×, suggesting that progressive domain expansion is more effective than global causal weighting for coupled multi-field systems in a non-separable architecture. However, both remain over an order of magnitude less accurate than Causal SPINN. The contrast with the EB beam is instructive, there, auxiliary decomposition or gradual domain solution was the key enabler for PINN convergence on a 4th-order operator, whereas here standard Causal PINNs achieve reasonable accuracy. This suggests that derivative order, rather than system coupling, is a more significant barrier for PINNs on engineering problems.

4. Dynamics of membrane and plate problems

Thin, tensioned membranes and elastic plates are fundamental two-dimensional structural elements widely used in tensile roofs, bridge decks, aerospace panels, and mechanical housings. Although geometrically similar, they exhibit different mechanical behaviors due to their different load-resisting mechanism. Membranes (Haghighat et al., 2021) primarily resist in-plane tension, and two-dimensional wave equations govern their dynamics. They are commonly found in architectural fabrics, tension roofs, and diaphragms. In contrast, plates are designed to sustain bending moments, shear forces, and torsion. They are modeled by higher-order PDEs, such as the Kirchhoff–Love theory (Gorman, 1978; Ebrahimi and Rastgo, 2008; Haghighat et al., 2021), whose fourth-order bi-harmonic operator introduces considerable numerical stiffness.

Therefore, membranes and plates introduce two new challenges for PINNs: the membranes test the ability of PINNs to capture wave propagation, whereas the plates test the stability of high-order derivative recovery. To rigorously assess the proposed Causal-Aux-SPINN framework, we consider two benchmarks: a free vibrating membrane and a Kirchhoff–Love plate, each examined in its fundamental mode of vibration.

4.1. Free-vibrating membrane

The free-vibrating Membrane serves as a canonical two-dimensional hyperbolic PDE benchmark. The problem includes capturing wave propagation, reflections at supported boundaries, and coupled space–time oscillations. We focus on the fundamental mode, characterized by oscillations with spatial wavenumbers of π in both x and y directions and a temporal frequency of $\sqrt{2}\pi$, which presents a challenging test for the accuracy of PINNs and SPINNs.

4.1.1. Non-dimensional formulation

A uniformly stressed membrane of areal mass density ρ and constant in-plane tension T_0 satisfies the two-dimensional wave equation (Haghighat et al., 2021):

$$\rho\, u_{tt} - T_0\left(u_{xx} + u_{yy}\right) = p(x, y, t), \tag{45}$$

where $u(x, y, t)$ is the transverse displacement and $p(x, y, t)$ the transverse load (Fig. 16).

Following the same nondimensionalization used for the Euler–Bernoulli beam, introduce a characteristic length L and the membrane time scale

$$T_{MB} = \sqrt{\frac{\rho L^2}{T_0}}. \tag{46}$$

Define the load-based displacement scale $U_{MB} = p_0 L^2 / T_0$ and the normalized variables

$$\tilde{x} = x/L, \quad \tilde{y} = y/L, \quad \tilde{t} = t/T_{MB}, \quad \tilde{u} = u/U_{MB}, \quad \tilde{p} = p/p_0.$$

With this choice, the inertial coefficient equals 1. The dimensionless equation reads

$$u_{tt} = u_{xx} + u_{yy} + p. \tag{47}$$

with the tildes omitted for clarity. For free vibration ($p = 0$), this reduces to

$$u_{tt} = u_{xx} + u_{yy}. \tag{48}$$

4.1.2. Forward problem

Following the above derivation, the normalized spatial and time domains are set to $(x, y, t) \in [0, 1] \times [0, 1] \times [0, 1]$. Moreover, a free-vibration test case is defined, and supported boundaries are adopted. The Membrane is released from rest with an initial displacement proportional to its first Eigenmode, given by the expression:

$$u(x, y, 0) = \sin(\pi x)\sin(\pi y) \tag{49}$$

The closed-form response for the problem considered is

$$u(x, y, t) = \sin(\pi x)\sin(\pi y)\cos\!\left(\sqrt{2}\,\pi\, t\right) \tag{50}$$

Eqs. (48)–(50) constitute a fully nondimensional well-posed initial–boundary value problem in which all coefficients are equal to one.

Unlike the Euler–Bernoulli beam (4th-order PDE), the membrane wave equation involves only second-order spatial derivatives. As

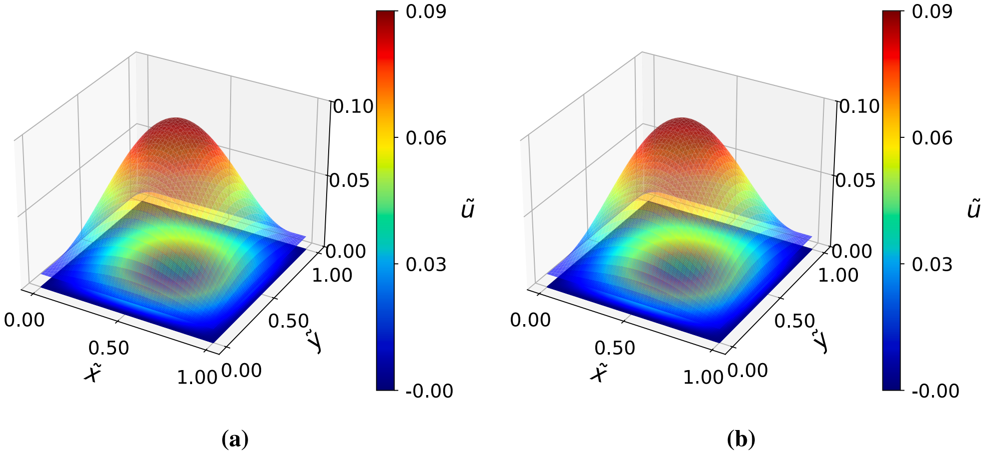


Fig. 17. Membrane: deflection heatmaps. (a) Exact solution; (b) Causal SPINN.

Table 13
Membrane: displacement $u(x, y, t)$ error metrics (mean ± standard deviation over 10 seeds).

Method	RMSE	MAE	rel. L_2 (%)
Baseline SPINN	$(4.95 \pm 1.10) \times 10^{-5}$	$(3.75 \pm 0.87) \times 10^{-5}$	0.0138 ± 0.0031
Causal SPINN	$(1.91 \pm 0.36) \times 10^{-5}$	$(1.47 \pm 0.27) \times 10^{-5}$	0.0053 ± 0.0010

demonstrated in Section 3.1, Causal SPINN can accurately approximate second-order derivatives, and there is therefore no need to introduce an auxiliary variable for this problem.

Both Baseline SPINN and Causal SPINN employ three body networks (one per spatial coordinate and one for time), each implemented as a modified MLP (Wang et al., 2021) with four hidden layers of 128 units (Cho et al., 2023). The approximation tensor rank is set to $r = 128$. The networks are trained with the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-5} for 100 000 epochs. Sixty-four collocation points are used for the PDE residual, boundary, and initial conditions. Loss weights are set to one for both the boundary and initial terms. The PDE-residual weight is $\lambda_{\mathrm{res}} = 0.1$ for Baseline SPINN and $\lambda_{\mathrm{res}} = 0.01$ for Causal SPINN. The causality parameter is fixed at $\varepsilon = 3$. The loss function follows Eq. (26), with $\mathcal{L}_{\mathrm{res}}$ now representing the 2D wave equation residual. The residual points are sampled on an equally spaced grid for the Causal SPINN and on a uniformly random grid for the Baseline SPINN method. Ten random-seed experiments are performed, and the response metrics are reported at the epoch corresponding to the minimum total loss.

Fig. 17 shows the predicted displacement fields. Table 13 summarizes the mean (± std) error metrics for the displacement $u(x, y, t)$. Causal SPINN reduces the relative L_2 error by over 60% compared to the Baseline, with comparable improvements in RMSE and MAE. Moreover, the standard deviations across all three metrics are roughly one-third of those for the Baseline model, indicating not only lower mean error but also greater run-to-run stability.

Fig. 18 reports the relative L_2 error across ten random seeds. The proposed method achieves both a lower mean error and a reduced standard deviation compared to the Baseline SPINN. It also demonstrates greater robustness, since most seeds lead to smaller errors compared to the Baseline SPINN shown on the left plot (Fig. 18a).

Fig. 19 presents the seed-averaged heatmap of the RMSE. For brevity, only the RMSE heatmap is shown, as the MAE exhibits the same spatial structure and ranking across methods, while MAE scalar values are reported in Table 13. For the Baseline SPINN (Fig. 19a), the RMSE field exhibits a localized concentration of error near the center of the spatial domain ($x \approx 0.5$, $y \approx 0.5$), coinciding with the peak amplitude of the fundamental mode shape $\sin(\pi x)\sin(\pi y)$. This indicates that error is concentrated in regions of maximum displacement, where the network must simultaneously track the largest transverse amplitudes and the fastest temporal oscillations. In contrast, the Causal SPINN (Fig. 19b) exhibits a notable reduction in both the magnitude and spatial extent of the high-error region. Peak RMSE values are lower, and the error field is more spatially uniform, with fewer pronounced localized concentrations. This spatial error pattern is directly relevant for engineering applications: the error concentration in the Baseline SPINN coincides with regions of maximum displacement and maximum in-plane stress, where accurate predictions are most critical for assessing membrane integrity.

Table 14 presents the relative L_2 norm error for all the relevant spatial derivatives. First-derivative errors are on the order of $10^{-2}\%$ and second-derivative errors on the order of $10^{-1}\%$. Since all required derivatives are already learned with high accuracy, the Causal formulation suffices, and there is no practical need to introduce additional auxiliary-variable architecture modifications for this membrane vibration problem.

4.1.3. *Sensitivity to causality parameter ε*

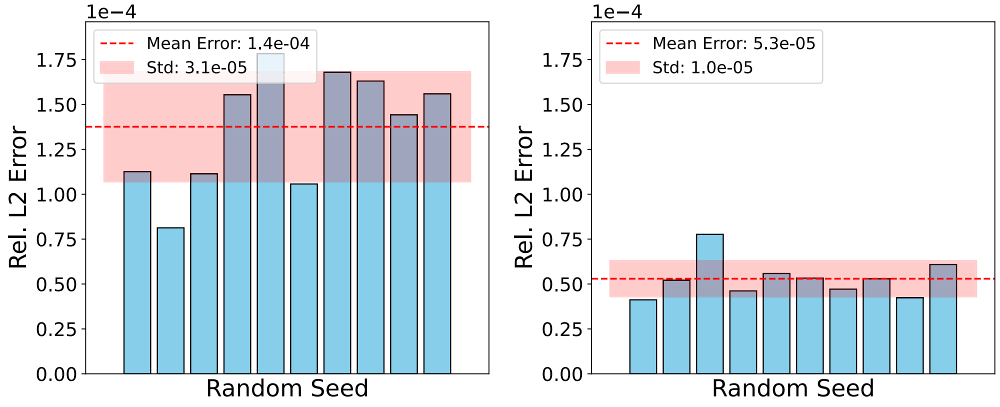
To confirm that the causality parameter used above is near-optimal, we sweep $\varepsilon \in \{1, 3, 5, 7\}$ under identical settings (architecture, sampling, schedule, and seeds) and report the mean ± std relative L_2 error over the same 10 seeds. Fig. 20(a) shows only modest variation across this range, with the best value at $\varepsilon = 5$ and a broad sweet spot spanning $\varepsilon = 3$–5. This confirms that the choice of ε in the main study is well-justified and close to optimal. The insensitivity of ε is consistent with the findings from the EB beam ablation (Section 3.1): moderate values $\varepsilon \in [3, 5]$ provide the best balance between temporal ordering and optimizer freedom, and this range transfers across problems with different temporal dynamics ($\omega = \pi \approx 3.14$ for the EB beam vs. $\omega = \sqrt{2}\,\pi \approx 4.44$ for the membrane).

4.1.4. *Noise robustness*

To replicate measurement uncertainty, the initial displacement is perturbed with zero-mean Gaussian noise (standard deviation 10% of the signal amplitude) while keeping all other settings fixed. The seed-wise dispersion in Fig. 20(b) and the metrics in Table 15 indicate that Causal SPINN maintains acceptable accuracy under noise (rel. $L_2 \approx 0.0416\%$, RMSE 1.5×10^{-4}, MAE 1.1×10^{-4}). Derivative errors remain low as well (Table 16).

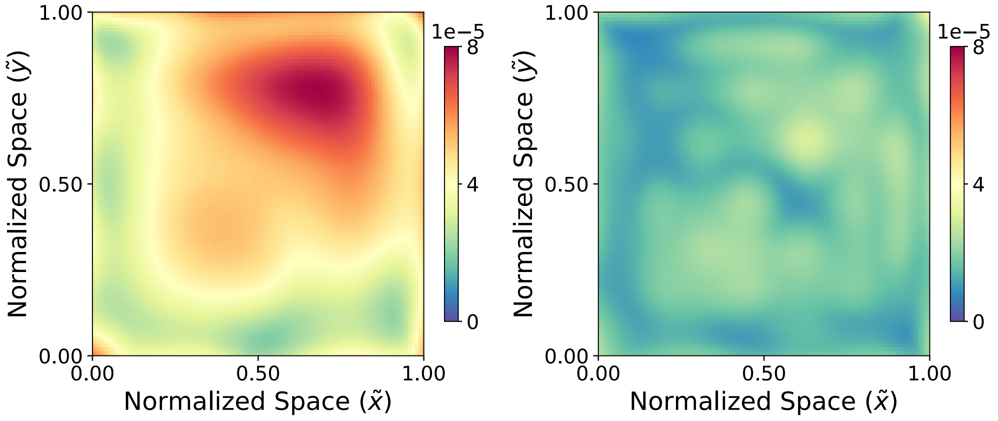
4.1.5. *Comparison with PINN methods*

The membrane is the first 3D benchmark, so the PINN comparison now examines how collocation scaling from nc^2 to nc^3 affects both memory and runtime. Two PINN variants are again evaluated. Causal



(a) **(b)**

Fig. 18. Membrane: seed-wise relative L_2 error. (a) Baseline SPINN; (b) Causal SPINN.

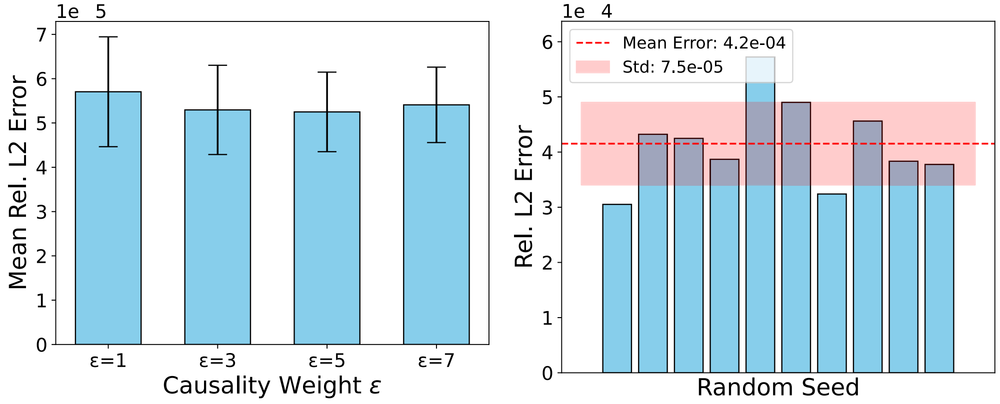


(a) **(b)**

Fig. 19. Membrane: RMSE heatmaps. (a) Baseline SPINN; (b) Causal SPINN.

Table 14
Membrane: spatial-derivative relative L_2 errors (mean ± standard deviation over 10 seeds).

Method	u_x	u_{xx}	u_y	u_{yy}
Baseline SPINN	$0.0247 \pm 0.0051\%$	$0.0798 \pm 0.0215\%$	$0.0253 \pm 0.0065\%$	$0.0827 \pm 0.0303\%$
Causal SPINN	$0.0143 \pm 0.0026\%$	$0.1070 \pm 0.0200\%$	$0.0140 \pm 0.0022\%$	$0.1090 \pm 0.0150\%$



(a) **(b)**

Fig. 20. Membrane: (a) mean relative L_2 error vs. causality weight ϵ (±1 standard deviation across 10 seeds); (b) relative L_2 error per seed under 10% Gaussian noise in the initial condition.

Table 15
Membrane with Gaussian noise: mean error metrics.

Method	rel. L_2 (%)	RMSE	MAE
Causal SPINN	0.0416	1.5×10^{-4}	1.1×10^{-4}

Table 16
Membrane with Gaussian noise: spatial-derivative relative L_2 errors (%) (mean ± standard deviation over 10 seeds).

Method	u_x	u_{xx}	u_y	u_{yy}
Causal SPINN	0.0534 ± 0.0093	0.134 ± 0.019	0.0547 ± 0.0118	0.133 ± 0.027

Table 17
Membrane: comparison of SPINN and PINN methods. SPINN results are mean ± std over 10 seeds; PINN results are single seed.

Method	rel. L_2 error	Epochs	Runtime (s)	ms/iter	GPU (MiB)
Baseline SPINN	(1.38 ± 0.31)e-4	100k	480	4.80	306
Causal SPINN	(5.3 ± 1.0)e-5	100k	454	4.54	306
Causal PINN	2.18e-3	100k	12,735	127	8690
Adaptive-domain PINN	3.59e-3	100k	12,923	129	8754

PINN (Wang et al., 2024) ($\varepsilon = 3$) and Adaptive-domain PINN (Mattey and Ghosh, 2022), both following the same architecture and λ_{res} sweep protocol as in Section 3.1.6, with nc = 64 per dimension ($64^3 = 262,144$ collocation points). No AUX PINN variant is required since the membrane equation involves only second-order derivatives. Table 17 reports accuracy and training cost for all methods.

The accuracy gap widens substantially in 3D: Causal SPINN outperforms the best PINN variant (Causal PINN) by 41× in relative L_2 error, compared with 2× on the EB beam and 16× on the Timoshenko beam. The runtime advantage also grows, from 6–11× per iteration on 2D beams to 28× here, reflecting the transition from nc^2 to nc^3 collocation scaling. GPU memory follows the same trend: PINNs require ~8.7 GiB versus 306 MiB for SPINN, a 28× gap driven by the $64^3 = 262,144$-point grid.

Causal PINN outperforms Adaptive-domain PINN on this wave problem by 1.6×, reversing the pattern observed on the coupled Timoshenko system (Section 3.2), where Adaptive PINN was 7× better. This suggests that causal weighting suits single-field wave propagation, while progressive domain expansion benefits coupled multi-field systems. Across all membrane experiments, the proposed Causal SPINN method consistently outperforms both the Baseline SPINN and standard PINN methods. Its performance remains stable across a broad range of ε, indicating that tuning this additional hyperparameter is straightforward. It also remains robust under noisy initial conditions. Overall, enforcing causality yields more accurate and reliable solutions without the need for extensive tuning.

4.2. Kirchhoff–Love plate

The free vibration of a thin elastic plate is modeled using the Kirchhoff–Love plate theory. Unlike the membrane benchmark of Section 4.1, the governing PDE involves fourth-order spatial derivatives, which considerably increase numerical stiffness during training. Consequently, standard PINNs often become unstable, converge slowly, and can exhibit sensitivity to the placement of collocation points when representing high-frequency, sharply varying fields. Moreover, the plate oscillates at angular frequency $2\pi^2 \approx 19.7$ s, far beyond the membrane's $\sqrt{2}\pi \approx 4.4$ s. This rapid time-scale variation hinders the efficiency of PINNs, which must capture fine temporal features alongside stiff spatial gradients. To mitigate these difficulties, we enhance the proposed Causal SPINN weighting with an auxiliary variable formulation that enforces lower-order derivative consistency, thereby stabilizing training on the biharmonic operator and enabling the accurate recovery of both displacement and curvature.

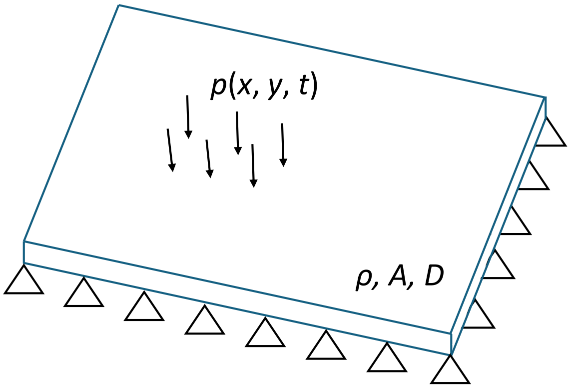


Fig. 21. Simply supported Kirchhoff–Love plate.

4.2.1. Nondimensional formulation

The transverse motion of a thin, isotropic Kirchhoff–Love plate of thickness h and areal mass density ρh is governed by

$$\rho h\, u_{tt} + D\, \nabla^4 u = p(x, y, t) \tag{51}$$

where $D = \dfrac{Eh^3}{12(1-\nu^2)}$ is the flexural rigidity and $\nabla^4 = \partial^4/\partial x^4 + 2\,\partial^4/\partial x^2 \partial y^2 + \partial^4/\partial y^4$ is the biharmonic operator (Fig. 21).

Following the Euler–Bernoulli nondimensionalization, we introduce a characteristic length L and the bending time scale

$$T_{KL} = L^2 \sqrt{\frac{\rho h}{D}} \tag{52}$$

together with a load-based displacement scale $U_{KL} p_0 L^4/D$ (used only when forcing is present), and form the corresponding normalized variables. With this scaling, the inertial prefactor satisfies $\frac{\rho h L^4}{D T_{KL}^2} = 1$ So the nondimensional Kirchhoff–Love equation reads

$$u_{tt} + \nabla^4 u = p \tag{53}$$

with the tildes omitted for clarity, and p now the normalized distributed load.

4.2.2. Forward problem

For the forward problem, we consider a supported plate on $(x, y, t) \in [0, 1] \times [0, 1] \times [0, 0.1]$ with $p \equiv 0$, reducing Eq. (53) to

$$u_{tt} + \nabla^4 u = 0, \tag{54}$$

subject to boundary conditions

$$u = \nabla^2 u = 0 \qquad \text{on } \partial([0, 1] \times [0, 1]). \tag{55}$$

The plate is released from rest in its fundamental mode,

$$u(x, y, 0) = \sin(\pi x)\sin(\pi y), \qquad u_t(x, y, 0) = 0, \tag{56}$$

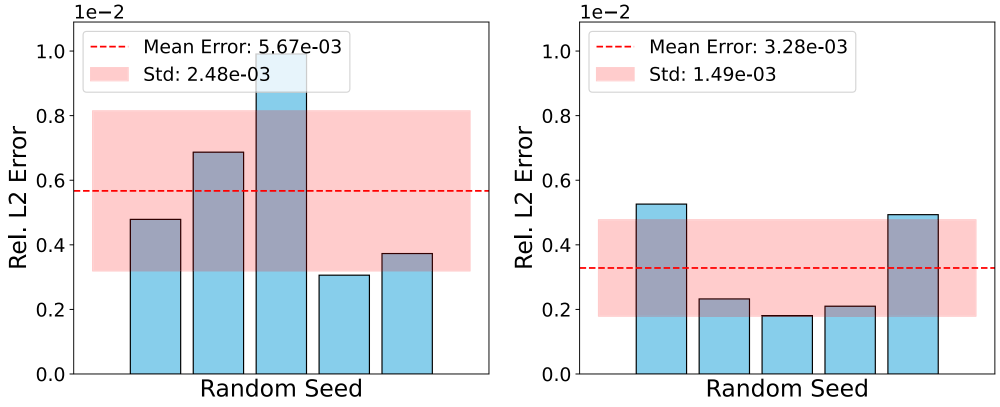
which yields the exact solution

$$u(x, y, t) = \sin(\pi x)\sin(\pi y)\cos\left(2\pi^2 t\right). \tag{57}$$

With $D = 1$ and $\nu = 0$ in this nondimensional setting, the bending moments and shear forces are defined as:

$$M_x = -u_{xx}, \quad M_y = -u_{yy}, \quad M_{xy} = -u_{xy}, \quad Q_x = -(u_{xxx} + u_{xyy}), \quad Q_y = -(u_{yyy} + u_{xxy}). \tag{58}$$

Both Baseline SPINN and Causal SPINN are examined in their auxiliary-variable forms, since without auxiliary regularization the



(a) **(b)**

Fig. 22. Kirchhoff–Love plate: seed-wise relative L_2 error. (a) Baseline SPINN; (b) Causal SPINN.

Table 18
Kirchhoff–Love plate: displacement $u(x, y, t)$ error metrics (mean ± standard deviation over 5 seeds).

Method	RMSE	MAE	rel. L_2 (%)
Baseline Aux SPINN	$(1.793 \pm 0.782) \times 10^{-3}$	$(1.092 \pm 0.472) \times 10^{-3}$	0.5654 ± 0.2465
Causal Aux SPINN	$(1.041 \pm 0.474) \times 10^{-3}$	$(4.386 \pm 1.531) \times 10^{-4}$	0.3283 ± 0.1494

Table 19
Kirchhoff–Love plate: second-order derivative relative L_2 errors (%) (mean ± standard deviation over 5 seeds).

Method	u_{xx} (%)	u_{yy} (%)	u_{xy} (%)
Baseline SPINN	1.078 ± 0.281	0.978 ± 0.167	0.6861 ± 0.2345
Causal SPINN	0.3365 ± 0.1359	0.3289 ± 0.1530	0.3059 ± 0.1346

Table 20
Kirchhoff–Love plate: third-order derivative relative L_2 errors (%) (mean ± standard deviation over 5 seeds).

Method	u_{xxx} (%)	u_{yyy} (%)	u_{xxy} (%)	u_{xyy} (%)
Baseline SPINN	3.477 ± 0.990	3.368 ± 0.645	1.083 ± 0.273	0.9181 ± 0.1882
Causal SPINN	0.5210 ± 0.1540	0.5542 ± 0.1157	0.3211 ± 0.1314	0.3252 ± 0.1369

models fail to converge on this high-order PDE. Each method uses three body networks (one per spatial coordinate and one for time), each implemented as a modified MLP (Wang et al., 2021) with four hidden layers of 64 units (Cho et al., 2023). The approximation tensor rank is set to $r = 128$. The networks are trained with the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-5} for 150 000 epochs. 128 collocation points are used for the PDE residual, boundary, and initial conditions. Loss weights are $\lambda_{\text{res}} = 0.01$ and $\lambda_{\text{BC}} = \lambda_{\text{IC}} = 10$ for Baseline SPINN, whereas Causal SPINN sets $\lambda_{\text{res}} = 0.01$, $\lambda_{\text{BC}} = \lambda_{\text{IC}} = 1$ and $\lambda_{v} = 10$. The loss function follows Eq. (28), with the Causal SPINN applying causal weighting to the PDE and consistency terms. The consistency loss enforces agreement between the auxiliary variable v and the Laplacian of the displacement field:

$$\mathcal{L}_{\text{cons}} = \frac{1}{N_{\text{cons}}} \sum_{k=1}^{N_{\text{cons}}} \Big(u_{xx}(x_k, t_k) + u_{yy}(x_k, t_k) - v(x_k, t_k) \Big)^2 . \tag{59}$$

The residual points are sampled on a uniformly random grid for the Baseline SPINN and on an equally spaced grid for the Causal SPINN method. The causality parameter is fixed at $\epsilon = 3$. Due to the computational cost, five random-seed experiments are performed for this benchmark instead of ten, and the response metrics are reported at the epoch corresponding to the minimum total loss.

As summarized in Table 18, Causal SPINN yields substantial gains over the Baseline SPINN across all displacement-error metrics, with reduced standard deviations indicating more consistent results across random initializations.

Fig. 22 reports the relative L_2 error across five random seeds. Causal Aux SPINN achieves both a lower mean error and a reduced standard deviation compared to the Baseline SPINN.

Heatmaps in Fig. 23 compare the exact deflection field with the Causal SPINN prediction. Fig. 24 visualizes the spatial distribution of RMSE; the MAE heatmap (not shown) exhibits the same spatial trends. For the Baseline Aux SPINN (Fig. 24a), the RMSE field exhibits a pronounced central hotspot coinciding with the peak amplitude of the fundamental mode shape $\sin(\pi x)\sin(\pi y)$, where the plate's midspan deflection is largest. This spatial pattern is consistent with the membrane case (Section 4.1), confirming that error concentrates where the network must track the largest transverse amplitudes. The Causal Aux SPINN (Fig. 24b) attenuates this peak and produces a more uniform error profile across the domain. This error concentration is directly relevant for engineering applications: the midspan region carries the largest bending moments ($M_x = -u_{xx}$, $M_y = -u_{yy}$), where accurate predictions are most critical for structural assessment.

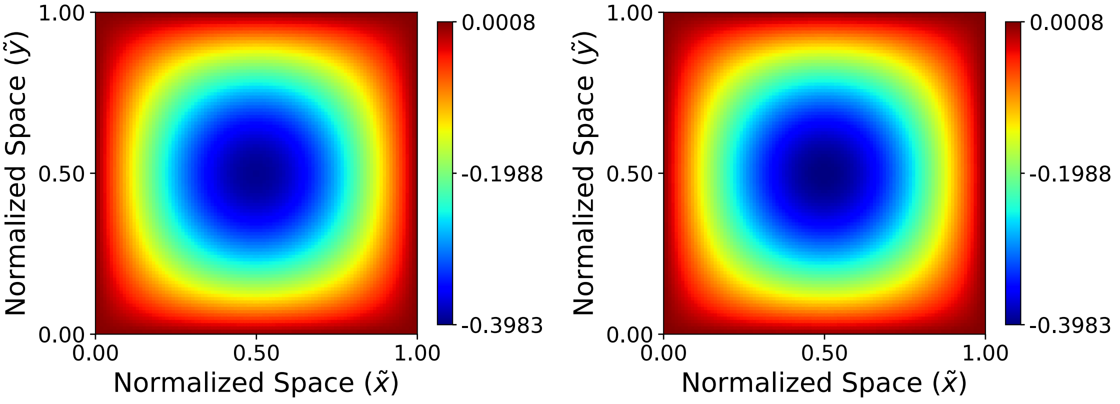
Tables 19 and 20 report the derivative errors. Causal Aux SPINN yields substantial reductions across all second- and third-order derivatives, with the improvements more pronounced for higher-order terms.

Neither Baseline SPINN nor Causal SPINN without the auxiliary variable converged on this high-order PDE, underscoring the importance of auxiliary-variable regularization for the Kirchhoff–Love plate problem. With this enhancement, Causal Aux SPINN delivers gains in both field and derivative accuracy, confirming that causal weighting, grid-based residual sampling, and explicit enforcement of derivative relations are essential for stable and accurate learning of biharmonic plate dynamics.

4.2.3. Comparison with PINN methods

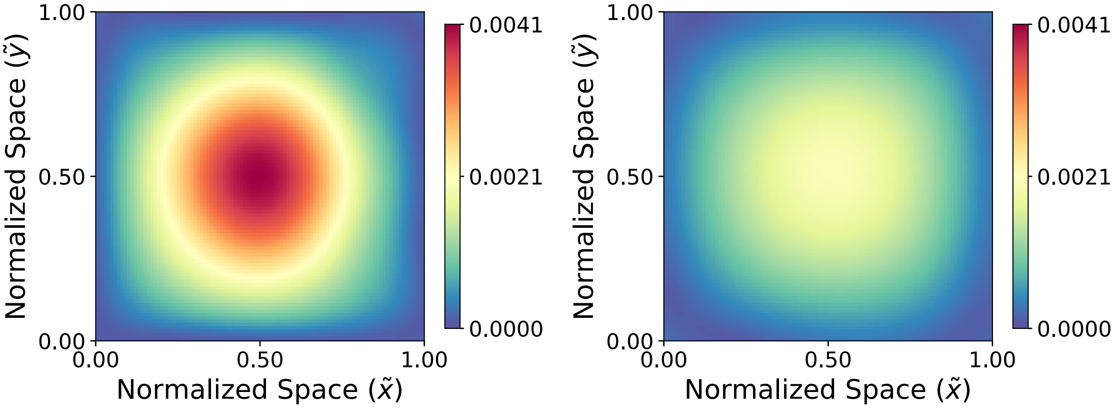
The Kirchhoff–Love plate combines 3D collocation with a 4th-order biharmonic operator, stressing both memory and AD depth simultaneously. Three PINN variants are evaluated as follows. *Causal PINN* (Wang et al., 2024) ($\epsilon = 3$), AUX PINN (Yuan et al., 2022) ($v \approx u_{xx} + u_{yy}$, $\lambda_v = 10$), and Adaptive-domain PINN (Mattey and Ghosh, 2022), all following the same λ_{res} sweep protocol as in Section 3.1.6, with 4 layers × 64 features to match the SPINN configuration. Table 21 reports accuracy and training cost.

At matching collocation density (nc = 128, i.e. $128^3 \approx 2\times10^6$ points), all three PINN variants exhaust GPU memory (41–77 GiB required vs. 16 GB available). Reducing to nc = 32 ($32^3 = 32{,}768$ points) allows training to proceed, but the coarse grid cannot resolve the biharmonic operator, yielding ∼87% relative L_2 error. By contrast,



(a) **(b)**

Fig. 23. Kirchhoff–Love plate: deflection heatmaps at $t = 0.1$. (a) Exact solution; (b) Causal SPINN.



(a) **(b)**

Fig. 24. Kirchhoff–Love plate: RMSE heatmaps. (a) Baseline SPINN; (b) Causal SPINN.

Table 21
Kirchhoff–Love plate: comparison of SPINN and PINN methods. SPINN results are mean ± std over 5 seeds; PINN results are single seed at reduced resolution (nc = 32).

Method	rel. L_2 error	Epochs	Runtime (s)	ms/iter	GPU (MiB)	nc
Baseline SPINN	(5.65 ± 2.47)e-3	150k	1355	9.03	562	128
Causal SPINN	(3.28 ± 1.49)e-3	150k	1277	8.51	692	128
Causal PINN[a]	8.68e-1	150k	–	–	OOM	128
AUX PINN[a]	8.51e-1	150k	–	–	OOM	128
Adaptive-domain PINN[a]	8.68e-1	150k	–	–	OOM	128

[a] OOM at nc = 128 (41–77 GiB required vs. 16 GB available). nc = 32 results shown for completeness.

Causal SPINN trains at nc = 128 within ~700 MiB and achieves sub-percent accuracy. This represents the limiting case of the scaling trend observed across all benchmarks: SPINN's $\mathcal{O}(Nd)$ collocation cost remains tractable where PINN's $\mathcal{O}(N^d)$ scaling becomes prohibitive, making the separable architecture essential for full-resolution 3D plate problems.

4.2.4. Inverse parameter estimation

Baseline SPINN and Causal SPINN, both augmented with the auxiliary-variable formulation, are then challenged with the inverse problem of identifying the uniform flexural rigidity D of a supported Kirchhoff–Love plate using only sparse displacement measurements. Four "sensors" are placed at $S_1 = (0.25, 0.25)$, $S_2 = (0.75, 0.75)$, $S_3 = (0.75, 0.25)$, and $S_4 = (0.25, 0.75)$ on the unit square, each recording $u(x_i, t)$ at $T = 100$ equally spaced instants (sampling rate $f_{sampling} = 100$ Hz) (see Fig. 25).

Synthetic time-series data are drawn from the exact solution. We then compare the convergence of the inferred D and the displacement-prediction error for both the Baseline and causally weighted models, thereby quantifying the impact of Causality and auxiliary regularization on parameter recovery under highly underdetermined measurement settings.

The models are treating D as an additional trainable scalar, initialized uniformly in $[0.5, 1.5]$ consistent with typical engineering uncertainty ranges for material parameters. The total loss combines the biharmonic PDE residual, boundary, and initial-condition terms, auxiliary consistency loss, and a data-fitting term on the sensor readings as derived in Eq. (29) with data loss defined as:

$$\mathcal{L}_{\text{data}} = \frac{1}{4} \sum_{i=1}^{4} \left(u_{\text{NN}}(x_i, y_i, t; D) - u_{\text{meas}}(x_i, y_i, t) \right)^2. \quad (60)$$

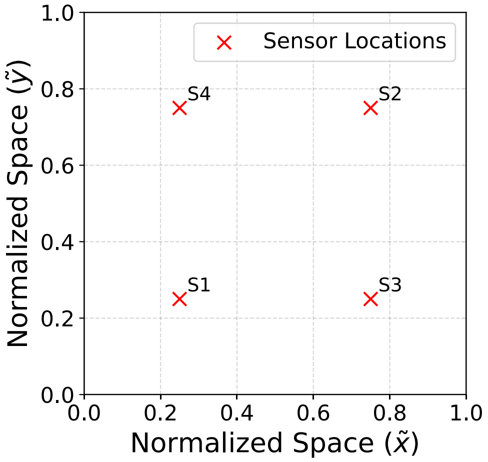


Fig. 25. Kirchhoff–Love plate: sensor locations on the unit-square plate used for inverse parameter estimation. Four sensors (S1-S4) are placed at $(0.25, 0.25)$, $(0.75, 0.75)$, $(0.75, 0.25)$, and $(0.25, 0.75)$.

where the consistency loss enforces the auxiliary variable v similarly to the forward problem, as defined in Eq. (59).

A value of $\lambda_{\text{data}} = 1$ is here used as a data loss regularization term. All other hyperparameters (network depth, rank of the approximated tensor, learning rate, collocation sampling, and epochs) have been kept the same as those of the forward-problem examined in Section 4.2.

Fig. 26 plots, for both architectures, the epoch-by-epoch evolution of (i) the relative L^2 error in the reconstructed displacement, $\|u_{\text{NN}} - u_{\text{exact}}\| / \|u_{\text{exact}}\|$, and ($ii$) the relative error in the inferred rigidity, $\left|D_{\text{est}} - D_{\text{true}}\right| / |D_{\text{true}}|$. Both models rapidly reduce the displacement error below 10^{-2}. Baseline Aux SPINN converges toward the actual value $D_{\text{true}} = 1$ but subsequently drifts upward due to accumulated error. In contrast, Causal SPINN exhibits minor early oscillations yet stabilizes around the actual value, yielding consistently lower error, which demonstrates that Causal weighting, combined with the introduction of the auxiliary variable, stabilizes inverse parameter learning.

In summary, both models successfully drive the displacement error below the 10^{-2} threshold, but only Causal Aux SPINN maintains a stable and accurate estimate of the flexural rigidity D throughout training. While Baseline Aux SPINN initially converges to the actual value, it subsequently drifts due to error accumulation. In contrast, the Causality-weighted model with auxiliary regularization remains tightly centered on $D = 1$ with minimal variance. This demonstrates that enforcing temporal Causality and derivative consistency is important for reliable inverse learning of structural stiffness from sparse, time-series measurements.

5. Conclusions

This work introduces Causal-Aux-SPINN for structural dynamics and demonstrates how coupling temporal reweighting with an auxiliary-variable head yields accurate and stable solutions for high-order PDEs that arise in beam, membrane, and plate models. The separable architecture factors the spatiotemporal field into low-rank components and exploits forward-mode automatic-differentiation, while the causal loss emphasizes early-time accuracy to curb error propagation. For fourth-order operators, an auxiliary branch predicts selected second-order fields and enforces consistency, thereby stabilizing higher derivatives without significantly complicating the training process.

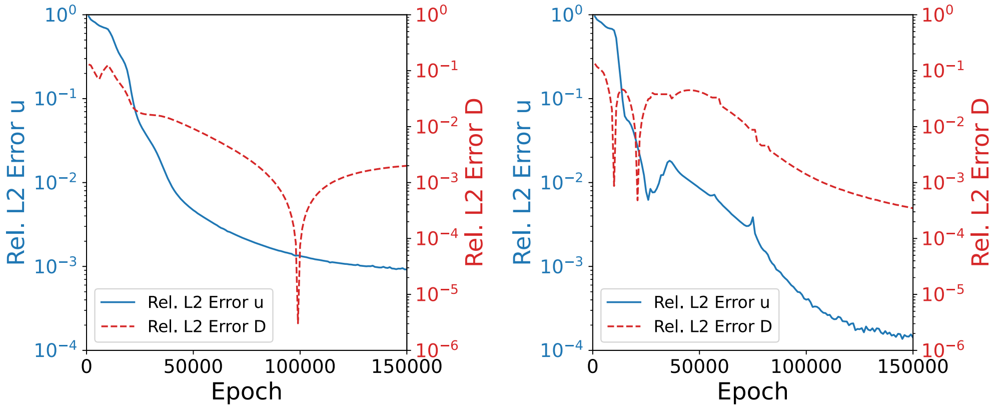
Across Euler–Bernoulli and Timoshenko beams, a membrane, and a Kirchhoff–Love plate, the proposed Causal-SPINN and its auxiliary variant consistently improved accuracy and robustness over Baseline SPINN, including inverse identification with sparse sensors and perturbed initial data. Results indicate that causality weighting reliably reduces global errors in forward dynamics and narrows seed-to-seed variability, while preserving a lightweight training pipeline compatible with the separable backbone. In the Euler–Bernoulli study, the auxiliary variable enabled accurate recovery of high-order responses, driving fourth-derivative errors to a regime suitable for engineering quantities such as bending moment and shear. In contrast, the Timoshenko case met derivative-accuracy requirements without auxiliary outputs, underscoring that the added branch can be deployed selectively when fourth-order stability is the bottleneck. For inverse problems, both the causal and auxiliary mechanisms facilitated the rapid and accurate recovery of stiffness parameters from limited time series, supporting their utility for identification tasks and digital-twin updates.

The framework remains general: the causal weighting is applicable to any time-dependent governing equation. It integrates with the same separable backbone used across all cases, and the auxiliary branch extends naturally to other higher-order operators by redefining the consistency target. Within the examined settings, these additions retained the practical advantages of SPINN while improving fidelity, suggesting a favorable balance between accuracy and computational efficiency for structural analyses. The spatial error analysis further shows that these improvements concentrate in regions of maximum curvature and displacement, precisely where accurate predictions are most critical for structural assessment.

Several directions merit further investigation. The present work adopts a fixed causality parameter ϵ, which proved adequate for the smooth temporal dynamics of the linear problems studied here. Extending the framework to chaotic or multi-scale regimes will likely require ϵ annealing, as demonstrated by Wang et al. (2024), and finer temporal resolution (N_t) may interact with the exponential decay in ways that warrant systematic study for stiff or rapidly oscillating dynamics. Beyond the linear setting, Cho et al. (2023) have already applied the separable SPINN backbone to nonlinear benchmarks including Navier–Stokes, suggesting that the causal and auxiliary mechanisms proposed here can be paired with the same architecture for nonlinear structural problems. The auxiliary variable formulation is particularly robust to such extensions, since $v = u_{xx}$ (or $v = \nabla^2 u$ for plates) is a purely kinematic quantity that remains valid regardless of constitutive nonlinearity and can be redefined to suit the governing operator at hand. Cho et al. (2023) have further demonstrated SPINN on non-smooth domains such as the L-shaped Poisson problem. Their approach combines a domain decomposition strategy, which splits irregular geometries into rectangular sub-domains trained separably. A masking approach handles boundary points in a non-separable manner while retaining separable interior collocation. These techniques should extend naturally to structural problems with geometric singularities such as cut-outs, sharp corners, or multi-material interfaces.

From an optimization perspective, the adoption of advanced second-order optimizers, such as quasi-Newton methods (e.g., L-BFGS), may provide faster and more stable convergence compared to first-order approaches. Moreover, further research on inverse problems will enable more robust identification of unknown parameters and the optimization of sensor placement, both in terms of number and spatial configuration, so as to achieve maximal observability with minimal instrumentation. Another promising direction involves improving the resilience of the model to noise in training data and to imperfections in the governing PDE itself, through probabilistic regularization, adaptive weighting strategies, or physics-constrained denoising techniques. These developments would broaden the applicability of Causal SPINN to real-world dynamical systems characterized by uncertainty and incomplete information.

In sum, causality-informed SPINN, with an optional auxiliary variable, offers a unified, easy-to-implement, and effective framework for



(a) **(b)**

Fig. 26. Kirchhoff–Love plate: training curves for inverse flexural rigidity estimation. (a) Baseline AUX SPINN; (b) Causal AUX SPINN.

forward and inverse dynamics of beams and plates. By combining temporal ordering with low-rank structure, the method delivers accurate fields and reliable derivatives at practical cost, and it provides a flexible foundation for future work on complex geometries, richer boundary conditions, and larger-scale structural applications.

CRediT authorship contribution statement

Anastasios Stamou: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Taniya Kapoor:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Michalis Fragiadakis:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization.

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.

Acknowledgments

The authors gratefully acknowledge the support of the European Union under Project **101169429 – FOURIER**, funded by the Marie Skłodowska-Curie Doctoral Networks (HORIZON-MSCA-2023-DN-01) within the framework of the European Commission's General Model Grant Agreement. This work was carried out as part of the doctoral project "**Advanced AI-METHODS for Inspection and Visualization (DC7)**".

Data availability

No data was used for the research described in the article.

References

Ahmad, Husna Zafar, Liu, Xiangyang, Sadiq, Muhammad Noveel, 2025. Optimizing physics-informed neural networks with hybrid activation functions: A comparative study on improving residual loss and accuracy using partial differential equations. Chaos Solitons Fractals 191, 115727. http://dx.doi.org/10.1016/j.chaos.2024.115727.

Ahmad, Husna Zafar, Jawad, Muhammad, Arif, Usama, Dar, Nayyar Ijaz, Sadiq, Muhammad Noveel, 2026. Scientific computing for thermal analysis in ternary hybrid nanofluid flow through cylinder with gyrotactic microorganisms: Thermal storage applications. J. Therm. Anal. Calorim. 151, 2617–2636. http://dx.doi.org/10.1007/s10973-025-15094-8.

Ahmad, Feroz Soomro, Zafar, Husna, 2025. Direct numerical simulation of magnetohydrodynamic slip-flow past a stretching surface using physics-informed neural network. Heat Transf. 1–9. http://dx.doi.org/10.1002/htj.70157.

Bauchau, Oliver A., Craig, James I., 2009. Euler-Bernoulli beam theory. In: Structural Analysis. Springer, pp. 173–221.

Baydin, Atilim Gunes, Pearlmutter, Barak A, Radul, Alexey Andreyevich, Siskind, Jeffrey Mark, 2018. Automatic differentiation in machine learning: a survey. J. Mach. Learn. Res. 18 (153), 1–43, URL http://jmlr.org/papers/v18/17-468.html.

Bazmara, Maziyar, Silani, Mohammad, Mianroodi, Mohammad, et al., 2023. Physics-informed neural networks for nonlinear bending of 3D functionally graded beam. In: Structures, vol. 49, Elsevier, pp. 152–162.

Cho, Junwoo, Nam, Seungtae, Yang, Hyunmo, Yun, Seok-Bae, Hong, Youngjoon, Park, Eunbyung, 2023. Separable physics-informed neural networks. Adv. Neural Inf. Process. Syst. 36, 23761–23788.

Cuomo, Salvatore, Di Cola, Vincenzo Schiano, Giampaolo, Fabio, Rozza, Gianluigi, Raissi, Maziar, Piccialli, Francesco, 2022. Scientific machine learning through physics–informed neural networks: Where we are and what's next. J. Sci. Comput. 92 (3), 88.

Dolph, C.L., 1954. On the Timoshenko theory of transverse beam vibrations. Quart. Appl. Math. 12 (2), 175–187.

Ebrahimi, Farzad, Rastgo, Abbas, 2008. An analytical study on the free vibration of smart circular thin FGM plate based on classical plate theory. Thin-Walled Struct. 46 (12), 1402–1408.

Eshaghi, Mohammad Sadegh, Bamdad, Mostafa, Anitescu, Cosmin, Wang, Yizheng, Zhuang, Xiaoying, Rabczuk, Timon, 2025. Applications of scientific machine learning for the analysis of functionally graded porous beams. Neurocomputing 619, 129119.

Eymard, Robert, Gallouët, Thierry, Herbin, Raphaèle, 2000. Finite volume methods. Handb. Numer. Anal. 7, 713–1018.

Fallah, Ali, Aghdam, Mohammad Mohammadi, 2024. Physics-informed neural network for bending and free vibration analysis of three-dimensional functionally graded porous beam resting on elastic foundation. Eng. Comput. 40 (1), 437–454.

Feng, Jiali, Liu, Zhijie, He, Xiuyu, Li, Qing, He, Wei, 2021. Vibration suppression of a high-rise building with adaptive iterative learning control. IEEE Trans. Neural Netw Learn. Syst. 34 (8), 4261–4272.

Goerguelue, Uemuet, 2009. Beam theories the difference between Euler-Bernoulli and timoschenko. Lect. Handouts.

Gorman, D.J., 1978. Free vibration analysis of the completely free rectangular plate by the method of superposition. J. Sound Vib. 57 (3), 437–447.

Griewank, Andreas, Walther, Andrea, 2008. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, second ed. SIAM, Philadelphia, ISBN: 978-0-89871-659-7.

Haghighat, Ehsan, Can Bekar, Ali, Madenci, Erdogan, Juanes, Ruben, 2021. Deep learning for solution and inversion of structural mechanics and vibrations. In: Modeling and Computation in Vibration Problems, Vol. 2, IOP Publishing, ISBN: 978-0-7503-3487-7, pp. 1–1 to 1–17. http://dx.doi.org/10.1088/978-0-7503-3487-7ch1, 2053-2563.

He, Wei, Meng, Tingting, Huang, Deqing, Li, Xuefang, 2017. Adaptive boundary iterative learning control for an Euler–Bernoulli beam system with input constraint. IEEE Trans. Neural Netw Learn. Syst. 29 (5), 1539–1549.

Hu, Zheyuan, Jagtap, Ameya D., Karniadakis, George Em, Kawaguchi, Kenji, 2023. Augmented physics-informed neural networks (APINNs): A gating network-based soft domain decomposition methodology. Eng. Appl. Artif. Intell. (ISSN: 0952-1976) 126, 107183. http://dx.doi.org/10.1016/j.engappai.2023.107183, URL https://www.sciencedirect.com/science/article/pii/S0952197623013672.

Jawad, Muhammad, Ahmad, Husna Zafar, Fu, Zhuojia, Alshehery, Sultan, Junaid, Muhammad, Khan, Ilyas, 2025. Physics-informed neural networks for rotating EMHD flow of jeffrey hybrid nanofluid with arrhenius activation energy and mass convections. Expert Syst. Appl. (ISSN: 0957-4174) 271, 126517. http://dx.doi.org/10.1016/j.eswa.2025.126517, URL https://www.sciencedirect.com/science/article/pii/S0957417425001393.

Kang, S.W., Lee, J.M., 2000. Application of free vibration analysis of membranes using the non-dimensional dynamic influence function. J. Sound Vib. 234 (3), 455–470.

Kapoor, Taniya, Wang, Hongrui, Núñez, Alfredo, Dollevoet, Rolf, 2023. Physics-informed neural networks for solving forward and inverse problems in complex beam systems. IEEE Trans. Neural Netw Learn. Syst. 35 (5), 5981–5995.

Kapoor, Taniya, Wang, Hongrui, Núñez, Alfredo, Dollevoet, Rolf, 2024a. Transfer learning for improved generalizability in causal physics-informed neural networks for beam simulations. Eng. Appl. Artif. Intell. 133, 108085.

Kapoor, Taniya, Wang, Hongrui, Stamou, Anastasios, El Sayed, Kareem, Núñez, Alfredo, Tartakovsky, Daniel M, Dollevoet, Rolf, 2024b. Neural differential equation-based two-stage approach for generalization of beam dynamics. IEEE Trans. Ind. Inform..

Karniadakis, George Em, Kevrekidis, Ioannis G, Lu, Lu, Perdikaris, Paris, Wang, Sifan, Yang, Liu, 2021. Physics-informed machine learning. Nat. Rev. Phys. 3 (6), 422–440.

Kingma, Diederik P., Ba, Jimmy, 2015. Adam: A method for stochastic optimization. In: International Conference on Learning Representations. URL https://arxiv.org/abs/1412.6980.

Leitao, Vitor M.A., 2001. A meshless method for Kirchhoff plate bending problems. Internat. J. Numer. Methods Engrg. 52 (10), 1107–1130.

Liu, Yu, Wu, Xiaoqi, Yao, Xiangqian, Zhao, Jingyi, 2022. Backstepping technology-based adaptive boundary ILC for an input–output-constrained flexible beam. IEEE Trans. Neural Netw Learn. Syst. 34 (11), 9314–9322.

Mandl, Luis, Goswami, Somdatta, Lambers, Lena, Ricken, Tim, 2025. Separable physics-informed DeepONet: Breaking the curse of dimensionality in physics-informed machine learning. Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 434, 117586. http://dx.doi.org/10.1016/j.cma.2024.117586, URL https://www.sciencedirect.com/science/article/pii/S0045782524008405.

Mattey, Revanth, Ghosh, Susanta, 2022. A novel sequential method to train physics informed neural networks for Allen Cahn and Cahn Hilliard equations. Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 390, 114474. http://dx.doi.org/10.1016/j.cma.2021.114474, URL https://www.sciencedirect.com/science/article/pii/S0045782521006939.

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

Moseley, Ben, Markham, Andrew, Nissen-Meyer, Tarje, 2023. Finite basis physics-informed neural networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. Adv. Comput. Math. 49 (4), 62.

Oniszczuk, Z., 2000. Free transverse vibrations of elastically connected simply supported double-beam complex system. J. Sound Vib. 232 (2), 387–403.

Rahaman, Nasim, Baratin, Aristide, Arpit, Devansh, Draxler, Felix, Lin, Min, Hamprecht, Fred, Bengio, Yoshua, Courville, Aaron, 2019. On the spectral bias of neural networks. In: Chaudhuri, Kamalika, Salakhutdinov, Ruslan (Eds.), Proceedings of the 36th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 97, PMLR, pp. 5301–5310, URL https://proceedings.mlr.press/v97/rahaman19a.html.

Raissi, Maziar, Perdikaris, Paris, Karniadakis, George 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.

Reddy, J.N., 1993. An Introduction to the Finite Element Method, second ed. McGraw-Hill, New York.

Shysheya, Aliaksandra, Diaconu, Cristiana, Bergamin, Federico, Perdikaris, Paris, Hernández-Lobato, José Miguel, Turner, Richard, Mathieu, Emile, 2024. On conditional diffusion models for PDE simulations. Adv. Neural Inf. Process. Syst. 37, 23246–23300.

Strikwerda, John C., 2004. Finite Difference Schemes and Partial Differential Equations. SIAM.

Wang, Sifan, Sankaran, Shyam, Perdikaris, Paris, 2024. Respecting causality for training physics-informed neural networks. Comput. Methods Appl. Mech. Engrg. 421, 116813.

Wang, Sifan, Teng, Yujun, Perdikaris, Paris, 2021. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J. Sci. Comput. 43 (5), A3055–A3081. http://dx.doi.org/10.1137/20M1318043.

Wang, Yueqing, Zhang, Peng, Liu, Yushuang, Zhao, Jianing, Lin, Jie, Chen, Yi, 2025. Aerodynamic coefficients prediction via cross-attention fusion and physical-informed training. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, (1), pp. 869–876.

Xiong, Wei, Long, Xiangyun, Bordas, Stéphane P.A., Jiang, Chao, 2025. The deep finite element method: A deep learning framework integrating the physics-informed neural networks with the finite element method. Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 436, 117681. http://dx.doi.org/10.1016/j.cma.2024.117681, URL https://www.sciencedirect.com/science/article/pii/S0045782524009356.

Yu, Yang, Yao, Houpu, Liu, Yongming, 2020. Structural dynamics simulation using a novel physics-guided machine learning method. Eng. Appl. Artif. Intell. (ISSN: 0952-1976) 96, 103947. http://dx.doi.org/10.1016/j.engappai.2020.103947, URL https://www.sciencedirect.com/science/article/pii/S0952197620302670.

Yuan, Lei, Ni, Yi-Qing, Deng, Xiang-Yun, Hao, Shuo, 2022. A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations. J. Comput. Phys. (ISSN: 0021-9991) 462, 111260. http://dx.doi.org/10.1016/j.jcp.2022.111260, URL https://www.sciencedirect.com/science/article/pii/S0021999122003229.

Zhao, Leo Zhiyuan, Ding, Xueying, Prakash, B. Aditya, 2023. Pinnsformer: A transformer-based framework for physics-informed neural networks. arXiv preprint arXiv:2307.11833.