Physics-informed machine learning: from methods to beam structures

Kapoor, T.

DOI
10.4233/uuid:b0dac776-9c30-4f97-a8da-acb9b40e9579

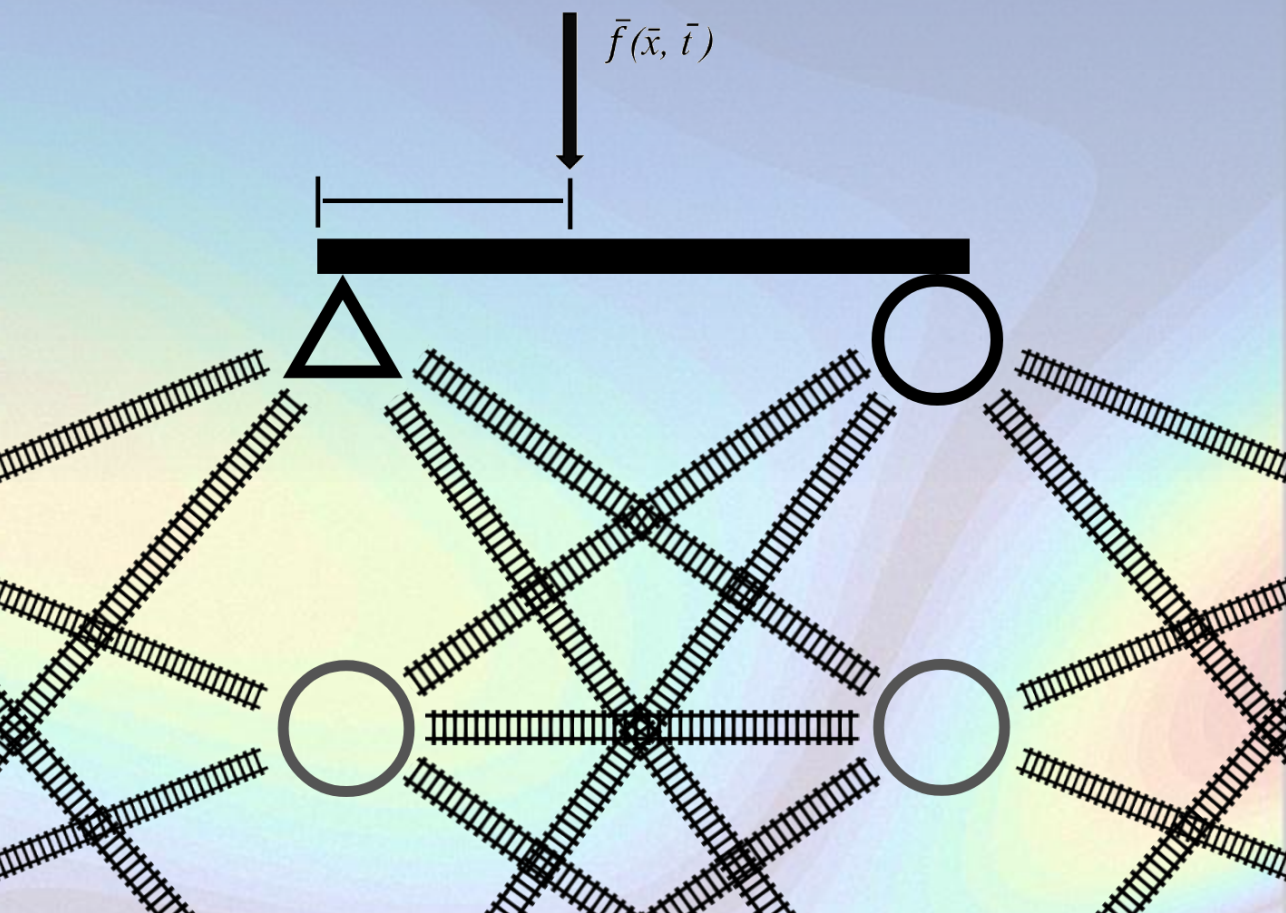
Publication date
2024

Document Version
Final published version

Citation (APA)
Kapoor, T. (2024). *Physics-informed machine learning: from methods to beam structures*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:b0dac776-9c30-4f97-a8da-acb9b40e9579

Taniya Kapoor

Physics-informed Machine Learning: From Methods to Beam Structures



Physics-informed machine learning: from methods to beam structures

Physics-informed machine learning: from methods to beam structures

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Tuesday 29 October 2024 at 12.30 o'clock

by

Taniya KAPOOR

Master of Science in Applied Mathematics, South Asian University, India
born in New Delhi, India

Dit proefschrift is goedgekeurd door de
Promotor: Prof. dr. ir. R.P.B.J. Dollevoet
Copromotors: Dr. H. Wang, Dr. A.A. Nunez Vicencio

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. R.P.B.J. Dollevoet,	Technische Universiteit Delft, promotor
Dr. H. Wang,	Technische Universiteit Delft, copromotor
Dr. A.A. Nunez Vicencio,	Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Prof. dr. ir. C. Vuik,	Technische Universiteit Delft
Prof. dr. S. Mishra,	ETH Zürich, Switzerland
Prof. dr. S. Roberts,	University of Oxford, UK
Prof. dr. D.M. Tartakovsky,	Stanford University, USA

Keywords: Physics-informed machine learning, Physics-informed neural networks, Beam dynamics, Causality, Generalization, Neural ordinary differential equations.
Printed by: Gildeprint
Cover by: Anahat Kelkar

ISBN 978-94-6496-258-1

An electronic copy of this dissertation is available at
https://repository.tudelft.nl/.

Dedicated to my beloved family.

CONTENTS

SUMMARY

Beams are the fundamental structural engineering element, supporting and stabilizing various structures ranging from suspension bridges to buildings and railways. Modeling and analyzing these structures necessitates a comprehensive understanding of the underlying beam dynamics constituting the structures. Simulating and predicting the beam dynamics is pivotal in ensuring structural integrity, optimizing structure design, and selecting appropriate materials. For instance, in railways, tracks and catenary contact wires are conceptualized as beams, allowing for the application of renowned beam theories like Euler-Bernoulli and Timoshenko. These theories provide a foundation for formulating partial differential equations (PDEs) that govern the dynamic behaviors of these beam systems.

These PDEs could be leveraged to simulate the underlying scenarios. The dissertation introduces physics-informed machine learning (PIML) based approaches tailored to simulate the dynamics of beam structures. The aim is to incorporate the physical laws in the neural networks training for more accurate and realistic simulations, handle noisy data effectively, and improve prediction accuracy while mitigating challenges such as multiscale problems and generalization. Chapter 1 outlines the primary challenges tackled in the dissertation. Chapters 2 through 5 detail the methodologies developed to address each challenge.

Chapter 2 presents a physics-informed neural network (PINN) based methodology to simulate complex beam systems with real-world material properties. In addition, inverse problems are solved in the presence of noisy data to predict unknown parameters, including force acting on the beam systems. It is essential to consider the real-world material parameters to simulate the dynamics of the modeled system and ensure the digital model represents the ground truth. However, incorporating material characteristics leads to multiscale PDE coefficients in the physical model, posing difficulty in training for PINNs. Subsequently, a framework is proposed to incorporate nondimensional PDEs into the PINN loss function. This approach facilitates efficient forward and inverse simulations while robust to noise and uncertainty in measurement data. The efficacy of this approach is demonstrated through simulations of Euler-Bernoulli and Timoshenko beam systems, contributing to the challenge of simulating large-scale systems with multiple interconnected components.

Chapter 3 investigates beam dynamic simulations on Winkler founda-

tions for large spatiotemporal domains using PIML. Predictions on expansive spatiotemporal domains are vital for structural integrity, design optimization, and control mechanisms. A causality-respecting PINN framework is introduced, enhancing prediction accuracy. Furthermore, integrating transfer learning addresses the need to re-train the network for different initial conditions and computational domains. Numerical experiments based on Euler-Bernoulli and Timoshenko theories validate the methodology for respecting the causality and generalizing the beam dynamics across similar problems. The approach efficiently predicts beam dynamics under diverse engineering scenarios, reducing computational costs and improving convergence.

Chapter 4 explores the generalization abilities of PIML, essential for practical applications requiring accurate predictions in unexplored regions. The proposed framework exploits the inherent causality in the PDE solutions by merging PIML models with recurrent neural architectures, namely neural oscillators. The neural ordinary differential equations in the form of neural oscillators effectively handle long-time dependencies and address gradient-related issues, fostering improved generalization in PIML tasks. Benchmark equations like viscous Burgers, Allen-Cahn, Schrödinger, and biharmonic Euler-Bernoulli beam equations are used to demonstrate the effectiveness of the proposed approach. Through extensive experimentation with time-dependent nonlinear PDEs, the study showcases superior performance compared to existing state-of-the-art methods. The proposed method provides accurate solutions for extrapolation and prediction beyond the training data by enhancing the generalization capabilities of PIML, promising advancements in complex system simulations.

Chapter 5 follows up on generalization of beam dynamics beyond PIML-based approaches. Computer-aided simulations are crucial for advancing engineering industries, but existing simulators often struggle to generalize beyond their training domain. The chapter proposes a two-stage methodology to tackle this challenge. Firstly, it utilizes specialized simulators tailored to the application, such as causal PINNs and black-box finite element simulations. Secondly, it integrates predictions from the first stage into a recurrent neural architecture, incorporating ordinary differential equations to capture intrinsic dynamics and enhance generalization. The approach efficiently captures causality and generalizes dynamics across various data sources. Numerical experiments cover fundamental structural engineering scenarios, including real-world catenary contact wire uplift predictions, and demonstrate superior performance compared to conventional methods, and promise for diverse industrial applications. This dissertation concludes with Chapter 6.

In particular, this dissertation introduces PIML methodologies for simulating complex beam structures, addressing key challenges such as incorporating real material properties, handling noisy data, and improving

prediction accuracy. Chapter 2 introduces a PINN-based methodology that efficiently simulates beam systems and predicts unknown parameters, mitigating the difficulties posed by multiscale PDE coefficients. Chapter 3 tackles the challenge of large-domain beam dynamics predictions on the Winkler foundations by using causality-respecting PINNs and integrating transfer learning to reduce computational costs. Chapter 4 addresses the challenge of out-of-domain predictions in PIML by introducing neural oscillators. Chapter 5 proposes a two-stage methodology to generalize beam dynamics simulations, integrating beam dynamics solvers and recurrent neural-based architectures, showcasing its efficacy in real-world applications such as catenary contact wire uplift predictions.

SAMENVATTING

Balken zijn fundamentele bouwkundige elementen die zorgen voor ondersteuning en stabilisatie van uiteenlopende constructies en kunstwerken, variërend van bruggen tot gebouwen en de bovenbouw van spoorwegen. Het modelleren en analyseren hiervan vereist een alomvattend begrip van de onderliggende dynamica die bepalend is voor deze constructies. Simulatie en prognosticeren van balkdynamica is van cruciaal belang bij het waarborgen van: de constructieve betrouwbaarheid , optimalisatie van constructieontwerpen en de selectie van geschikte bouwmaterialen. Een voorbeeld hiervan is de railinfrastructuur waarin spoorbanen en rijdraden van de bovenleiding kunnen worden weergegeven als balken, waardoor toepassing van alom bekende balktheorieën, zoals Euler-Bernoulli en Timosjenko, mogelijk wordt gemaakt. Deze theorieën bieden een basis voor het formuleren van partiële differentiaalvergelijkingen (PDVn) waarmee het dynamische gedrag van deze balkconstructies kan worden gemodelleerd. De PDVn kunnen worden gebruikt om onderliggende scenario's te simuleren.

Dit proefschrift introduceert op fysica-gebaseerde machine learning (Physics Informed Machine Learing of PIML) benaderingen die zijn toegesneden op het simuleren van de dynamica van balkconstructies. Het doel ervan is om de natuurkundige wetten op te nemen in de machine learning ten behoeve van: het trainen van neurale netwerken voor nauwkeurige en realistische simulaties, doelgericht gebruik van dataruis en verhoging van de accuraatheid van voorspellingen, terwijl problemen met multischaal-modellering en generalisatie worden beperkt.

Hoofdstuk 1 beschrijft de belangrijkste vraagstukken die in het proefschrift worden behandeld. De hoofdstukken 2 tot en met 5 gaan in op de methodologieën die zijn ontwikkeld om deze vraagstukken aan te pakken. Hoofdstuk 2 beschrijft de methodologie gebaseerd op fysica geënte neuraal netwerken (Physics-informed neural networks of PINNs) om complexe balkconstructies te simuleren aan de hand van de echte materiaaleigenschappen uit de praktijk. Tevens wordt een oplossing aangereikt voor inverse problemen door dataruis bij prognostiek van onbekende parameters, zoals de krachten die op balkconstructies werken. Het is van belang om rekening te houden met materiaalgebonden parameters uit de praktijk bij simulatie van de dynamica van een gemodelleerde constructie en ervoor te zorgen dat het digitale model de ware toestand beschrijft. Het opnemen van materiaalkenmerken leidt echter tot multischaal-PDV-coëfficiënten in het fysieke model, hetgeen proble-

men oplevert bij de training van PINNs. Daarnaast wordt in hoofdstuk 2 een raamwerk voorgesteld om non-dimensionale PDVn op te nemen in de PINN-verliesfunctie. Deze aanpak maakt efficiënte voorwaartse en inverse simulaties mogelijk, terwijl deze robuust zijn voor ruis en meetonzekerheden.

De doeltreffendheid van deze aanpak wordt gedemonstreerd door simulaties van Euler-Bernoulli en Timoshenko-balkconstructies die bijdragen aan de condities voor het simuleren van omvangrijke constructies met diverse onderling verbonden componenten. Hoofdstuk 3 gaat in op de simulaties van balkdynamica bij Winkler-funderingen voor grote tijdruimtelijk domeinen met behulp van PIML. Prognostiek van uitgestrekte tijdruimtelijke domeinen zijn van cruciaal belang voor constructieve betrouwbaarheid, ontwerpoptimalisatie en de bepaling van controlemechanismen. In hoofdstuk 3 wordt ook een PINN-framework voorgesteld waarin de causaliteit wordt beschouwd, hetgeen de nauwkeurigheid van de prognostiek verbetert. Daarbij komt de integratie van transfer learning tegemoet aan de noodzaak om het netwerk opnieuw te trainen voor verschillende initiële omstandigheden en computationele domeinen. Numerieke experimenten gebaseerd op de theorieën van Euler-Bernoulli en Timoshenko valideren de methodologie voor het beschouwen van de causaliteit en het generaliseren van de balkdynamica in vergelijkbare vraagstukken. De aanpak prognosticeert op efficiënte wijze de balkdynamica in diverse technische scenario's, waardoor computationele kosten voor het uitvoeren van berekeningen worden verlaagd en de convergentie wordt verbeterd.

Hoofdstuk 4 gaat nader in op de generalisatiemogelijkheden van PIML die noodzakelijk zijn voor praktische toepassingen en die nauwkeurige voorspellingen in andere onontgonnen gebieden van toepassing vereisen. Het voorgestelde raamwerk voor generalisatie maakt gebruik van de inherente causaliteit in de PDV-oplossingen door PIML-modellen samen te voegen met een terugkerende neurale architectuur en dan met name neurale oscillatoren. De neurale standaard differentiaalvergelijkingen in de vorm van neurale oscillatoren gaan effectief om met langdurige afhankelijkheden en zijn gericht op gradiëntgerelateerde problemen, waardoor verbeterde generalisatie in PIML-taken wordt bevorderd. Aan de hand van een benchmark wordt de effectiviteit aangetoond van de voorgestelde aanpak voor Burgersvergelijkingen, Allen-Cahn, Schrödinger en biharmonische Euler-Bernoulli-balkvergelijkingen. Door uitgebreide experimenten met tijdsafhankelijke niet-lineaire PDVn laat de studie veelbelovende prestaties zien in vergelijking met bestaande, conventionele methoden.

De voorgestelde methode biedt nauwkeurige oplossingen voor extrapolatie en prognoses die verder gaan dan de trainingsgegevens als gevolg van verbetering van generalisatiemogelijkheden van PIML, hetgeen een veelbelovende vooruitgang bij complexe systeemsimulaties. Hoofd-

stuk 5 gaat in op de generalisatie van balkdynamica die verder raakt dan de op PIML gebaseerde benaderingen. Computersimulaties zijn cruciaal in de ontwikkeling van technische en industriële sectoren, maar bestaande simulatie-methoden hebben vaak moeite om buiten hun trainingsdomein te generaliseren. In hoofdstuk 5 wordt een tweestapsmethodologie voorgesteld om deze problematiek aan te pakken.

In de eerste stap wordt gebruik gemaakt van gespecialiseerde simulatoren die zijn afgestemd op de toepassing, zoals causale PINNs en black-box-eindige-elementensimulaties. In de tweede stap, integreert de methode prognoses uit de eerste fase in een terugkerende neurale architectuur, waarbij gewone differentiaalvergelijkingen worden geïntegreerd om de intrinsieke dynamiek vast te leggen en de generalisatie te verbeteren. De benadering in de methode legt op efficiënte wijze de causaliteit vast en generaliseert daarbij de dynamiek tussen verschillende gegevensbronnen. Numerieke experimenten beslaan fundamentele bouwkundige scenario's, inclusief prognoses over de bovenleiding van de rijdraad van spoorwegen in de praktijk. De experimenten leiden tot zeer goede resultaten in vergelijking met conventionele methoden en zijn dan ook veelbelovend voor diverse industriële toepassingen. Dit proefschrift wordt afgesloten met Hoofdstuk 6.

In het bijzonder introduceert dit proefschrift PIML-methodologieën voor het simuleren van complexe balkconstructies, waarbij belangrijke vraagstukken worden aangepakt zoals het opnemen van echte reële materiaaleigenschappen in de beschouwingen, het omgaan met dataruis en het verbeteren van de nauwkeurigheid van prognoses. Hoofdstuk 2 introduceert een op PINN gebaseerde methodologie die op efficiënte wijze balkconstructies simuleert en onbekende parameters prognosticeert , waardoor de problemen van multischaal PDV-coëfficiënten worden gereduceerd. Hoofdstuk 3 behandelt het vraagstuk van prognostiek van balkdynamica van omvangrijke constructies op de Winkler-funderingen door gebruik te maken van causaliteit beschouwende PINNs en de integratie van transfer learning om zo de computationele kosten voor berekening te verlagen. Hoofdstuk 4 gaat in op de vraagstukken bij prognoses buiten het domein in PIML door de introductie van neurale oscillatoren. Hoofdstuk 5 stelt een tweetrapsmethodologie voor om simulaties van balkdynamica te generaliseren, waarbij oplossingen van de balkdynamica en terugkerende, op neurale gebaseerde architecturen worden geïntegreerd. Hiermee wordt de doeltreffendheid ervan gedemonstreerd in toepassingen in de praktijk, zoals prognoses van opwaartse krachten in bovenleidingrijdraden.

Hoofdstuk 3 behandelt het vraagstuk van prognostiek van balkdynamica van omvangrijke constructies op Winkler funderingen door gebruik te maken van causaliteit beschouwende PINNs en de integratie van transfer learning om zo de computationele kosten voor berekening te verlagen.

1

INTRODUCTION

1

1.1. BACKGROUND

Structural engineering ensures safe and stable built environments through precise design and analysis of load-bearing structures. One such pivotal load-bearing structure is the beam, serving as foundational elements that support and stabilize a wide range of structures. Progress in structural engineering depends on advancement in beam-based structures, and comprehending the underlying beam dynamics within these structures is imperative for accurate modeling and analysis. The development of methodologies simulating the dynamics of these beam systems is crucial to optimizing the efficiency, safety, and resilience of structural designs.

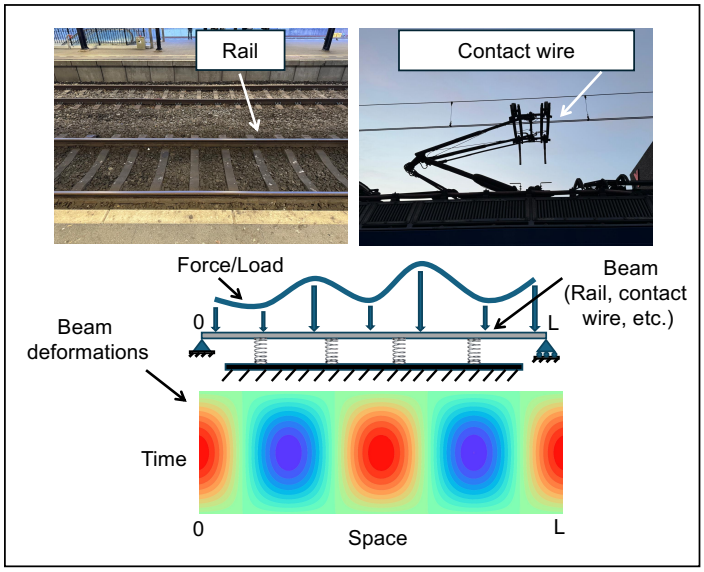


Figure 1.1: Examples of beams under varying transverse force. The direction of force on the rail is from above, whereas for the contact wire, it is from below. Different contour colors represent distinct space-time deformations of the beam and are not subject to scale.

For instance, the railway system incorporates diverse beam-based subsystems spanning several engineering domains. These include the interactions between catenaries and pantographs [1, 2] and the dynamics of wheels and tracks [3, 4]. The passage of trains exerts substantial loads on tracks [5, 6], and pantograph contact wire, causing deformations as shown in Fig. 1.1. Predicting deformations in the form of displacement assists in safety, infrastructure maintenance, and operational efficiency. In general, predicting the deformations for

beam-type structures under the action of varying forces contributes beyond railway engineering and is crucial in structural engineering, including designing buildings and bridges, among other applications [7].

These beam deformation could be studied in several ways, including data-driven machine learning, laboratory prototypes, and commercial software simulations. However, these approaches face their own set of challenges. For instance, machine learning-based data-driven methods for simulation necessitate substantial amounts of data, which is particularly challenging given the extensive spread of rail tracks covering thousands of kilometers, with each location equally significant.

Further, the cost of developing and maintaining prototypes for lab-based experiments is high, and these prototypes may not fully represent real-world conditions due to their limited exposure to varied environmental factors and restricted speed ranges. Moreover, understanding and accounting for material variations is important but restricted for prototype-based testing. Commercial software simulations, while potent, often need intricate meshing, increasing computational cost and time [8]. These challenges underline the need for improved methodologies that generalize across large domains and diverse conditions.

Mathematically, beam structures are described through partial differential equations (PDEs) [9]. Various theories have been developed to model beams and interconnected beam systems, considering beam characteristics like length, width, and rotatory effects. Euler-Bernoulli and the Timoshenko beam theories are widely recognized and utilized among these theories. These theories yield specific PDEs, namely the Euler-Bernoulli beam PDE and the Timoshenko PDE. Hence, rather than relying on prototype-based, experimental, data-driven methods or complex finite-element commercial software, these PDEs could be simulated to determine deflection profiles for different beam configurations.

The traditional approach to solving PDEs involves analytical methods to obtain closed-form solutions [10]. However, analytical solutions can only be obtained for a handful of PDEs and are often unavailable for complex PDEs, including those governing the beam dynamics [9]. Instead, numerical methods, such as finite difference, finite element, finite volume, and spectral methods, are often utilized to approximate the solutions of such PDEs. However, employing these methods requires specific expertise. For instance, mesh creation, the foundation of numerical-based methods, is computationally expensive and becomes even more challenging for complex geometries [8].

Additionally, several method-dependent parameters, such as the mesh size and the relation between time and space step sizes [10], exacerbate the challenge of employing numerical methods. The iterative process can be computationally costly, and the solutions are difficult

to generalize outside the computational domain. Also, solving inverse problems with numerical methods is particularly challenging. Given some solution data, the ill-posed inverse problems require estimating the unknown model parameters [11]. However, numerical-based methods require multiple iterative runs, with each run requiring solving the PDE and estimating the unknown parameters, increasing the computational cost and challenge of the problem.

Alternatively, deep learning could be utilized as a surrogate to approximate the solutions of PDEs. However, deep learning thrives on data, necessitating a pre-existing dataset of PDE solutions to train neural networks effectively [12] and develop the surrogate model. While solutions obtained from numerical methods may assist in training the neural network efficiently, data-driven machine learning approaches might disregard the underlying physical principles governing the system [13]. An alternative approach involves incorporating the governing physical equations into the learning process and integrating them with the expressive capabilities of deep neural networks, known as physics-informed machine learning [8]. Physics-informed neural networks [14], one of the cornerstones of physics-informed machine learning, have emerged as a paradigm for simulating intricate physical phenomena by training neural architectures solely on the boundary and initial conditions of the problem along with the PDE in the loss function of the neural network [15].

PINNs employ a unified framework to efficiently simulate forward and inverse problems by directly embedding physical laws into the deep learning architecture. This integration significantly reduces the reliance on extensive datasets, as the surrogate models utilize embedded physical principles to enhance predictions and infer parameters from observed data, even for complex problems with noisy datasets. Additionally, PINNs eliminate the need for mesh generation, a common requirement in numerical methods like finite element analysis. This meshless approach simplifies the simulation process and provides flexibility in handling geometric complexities.

The advantages of PIML for physical simulations motivate its applicability for developing the digital twin and simulating the beam dynamics in railway systems. In particular, beam structure simulations in railway engineering have largely been unexplored through the lens of physics-informed machine learning [16], and this dissertation aims to lay a foundation for physics-informed simulations for railway and structural engineering in general. However, as presented in this dissertation, applying PIML to predict beam dynamics is not straightforward. Several advancements in the PIML framework are required to advance the beam simulations. In particular, challenges like the multiscale coefficient of the PDEs governed by the material properties, large space-time domain, and generalization remain open problems in physics-informed machine

learning, which are fundamental for simulating beam dynamics in the context of railway engineering.

This dissertation presents PIML-based methodologies to simulate beam dynamics in structural engineering including railways. The developed methodologies, particularly on beam-based systems, aim to contribute multidisciplinary insights bridging the domains of physical simulations, machine learning, and structural engineering. The research addresses challenges inherent in PIML, including multiscale and large domain simulations. Furthermore, the research investigates temporal and parametric generalizations for canonical problems, including the deformation analysis of beams and real-world catenary contact wires under varied loading conditions.

1.2. CHALLENGES FOR BEAM SIMULATIONS THROUGH PHYSICS-INFORMED MACHINE LEARNING

This section discusses the major challenges in beam dynamics simulations through physics-informed machine learning that will be addressed in this dissertation.

1.2.1. COMPLEX SYSTEM SIMULATIONS

Beam dynamics in railway and structural engineering are complex and challenging to predict. Beams undergo deflections under acting dynamic loads. Further, many complex systems are based on multiple interconnected beams [17]. Simulating complex beam systems involves multiple PDEs describing intricate phenomena. PDEs governing the complex beam dynamics are characterized by higher-order derivatives, typically up to fourth order, and possess multiscale coefficient values arising from the material properties of the beam.

PINNs offer a promising approach for solving such complex PDE systems through developing a simulation-based surrogate model. By embedding physical equations in the loss function and leveraging neural networks universal function approximation property, PINNs can approximate high-fidelity solutions, providing well-posed PDEs. However, PINNs also suffer from challenges relevant to simulating complex PDEs.

In particular, one challenge for PINNs is learning relevant physical phenomena with large coefficients in the physical equation [18]. For instance, real-world beams in railway engineering, usually made of aluminum, are examples of such complex physical systems. Accurate simulations necessitate simulating PDEs with real-world coefficient values. Such simulations are crucial for developing robust methodologies to predict beam deformation across various material properties.

In essence, developing PIML-based methodologies for solving multi-scale issues in complex beam systems must be investigated. What

changes should be made to improve the performance of the physics-informed method for multiscale problems? Improving the performance of PINNs for multiscale problems and higher-order derivatives remains an open challenge and is not limited to beam dynamics.

1.2.2. INVERSION OF COMPLEX SYSTEM WITH NOISY DATA

In addition to solving the forward problem and predicting beam dynamics, another challenge in beam dynamic computations is solving the inverse problem. The inverse problem refers to inferring the unknown parameters and loadings acting on the beam systems. Solving such inverse problems requires collecting spatio-temporal data, possibly through sensor measurements. However, data collection through sensors is expensive and yields noisy observations, which must be utilized to provide robust parameter approximations.

PIML algorithms have been proposed to solve inverse problems and have shown success across the domain [19, 20]. Solving inverse problems with PINNs requires data at certain locations to simulate an ill-posed physical system. Ideally, PINNs simulate the ill-posed physical equations and approximate unknown parameters, providing some data at certain locations. However, the difficulty of the inverse problem increases for complex systems using noisy data. The challenge is whether PIML algorithms can cope with noise to simulate the underlying complex beam systems and estimate unknown parameters and functions.

1.2.3. LARGE DOMAIN SIMULATIONS

Simulating in a large domain refers to predicting and analyzing the behavior and interactions of systems across extensive spatial and temporal domains rather than focusing on smaller, localized segments. This approach is particularly relevant in fields like structural engineering, where the structures under investigation, such as bridges or railway tracks, cover vast distances. For instance, an example of a vast spatial domain includes railway tracks, which must be analyzed throughout the thousands of kilometers of the spatial domain. Many PIML algorithms are only tested in small domains, but extending PIML to larger spatial domains is crucial in structural engineering.

Mathematically, applying PIML to large domains involves training over an extensive space-time domain, which is complex and challenging [21, 22]. The difficulty arises from the inherent biases in conventional PIML methods like PINNs, which prioritize resolving later time levels due to implicit gradient biases. This bias violates temporal causality and results in inaccurate predictions, especially in systems heavily dependent on initial conditions. Overcoming these challenges is crucial for PIML to effectively simulate large-domain structural behaviors, mitigating the

challenge of multiple computations for smaller domains or domain decomposition strategies.

1.2.4. GENERALIZATION

Generalization refers to the ability of a neural network model to predict scenarios not included in the training dataset. This prediction is critical for applying a trained model in real-world conditions where it might encounter settings or situations different from those on which it was originally trained. Out-of-domain prediction is essential for models intended to be used in practical applications, such as structural engineering, where the modeled systems must operate reliably under a range of untested conditions. Achieving this capability requires the model to learn from the data it was trained on and deeply understand and apply the underlying physical principles governing the system. The generalization ability ensures that the model predictions remain valid even when extrapolating beyond the training domain, making it both a challenging and a vital aspect of PIML development.

Generalization is not only a challenge for PIML but, in general, for machine learning, at least for regression problems [23]. While PIML integrates physical laws into its frameworks to enhance robustness, it still struggles with accurately predicting outcomes outside the training domain [24]. This limitation arises because minimizing the PDE residuals during training does not effectively control the generalization error, meaning the model can perform well on familiar data but may not provide accurate predictions on new, unseen data. Additionally, embedding physical constraints into the model does not guarantee genuine physical comprehension or robustness beyond trained scenarios. Achieving reliable predictions in new and varied situations remains an open problem. This complexity underscores the need for developing methodologies that improve the ability of PIML to generalize beyond their training data, which is crucial for their practical application in modeling complex physical systems.

1.3. RESEARCH OBJECTIVES AND QUESTIONS

This dissertation explores the potential of PIML methodologies for simulating fundamental beam dynamics prevalent in railway and structural engineering. The research objectives encompass simulating the behavior of complex beams, solving inverse problems for estimating the quantity of interest, including noisy data, developing an approach for simulation in large domains, and reducing computational costs. Additionally, out-of-domain prediction will be sought, and physics-informed methodologies will be designed to generalize the existing PIML and numerical solvers.

The following research questions are explored to address the research objectives:

1. How to predict beam deformations through PIML-based methodologies, mitigating the challenge of multiscale coefficients?
2. How effectively do physics-informed algorithms tackle inverse beam dynamic problems and predict underlying dynamics and unknown parameters from noisy data?
3. How to simulate beam deformations within large spatiotemporal domains using PIML?
4. How to accelerate and generalize PIML-based methodologies for simulating similar beam dynamic problems in large spatiotemporal domains?
5. How to develop PIML-based frameworks capable of predicting out-of-domain for canonical problems, including nonlinear and high-order problems?
6. How to generalize beam dynamic solver to predict out-of-domain scenarios, particularly for beam systems like catenaries with varying train speeds?

1.4. CONTRIBUTIONS

This section outlines the main contributions of this dissertation. The theoretical or methodological advancements presented in the dissertation are as follows:

- This dissertation integrates structural engineering, computational mechanics, and machine learning by proposing machine learning-based solutions for mechanical problems encountered in the form of beam dynamics in railway and structural engineering.
- This dissertation proposes simulating the railway or structural dynamics modeled as beams through physics-informed machine learning. Specifically, Chapter 2 presents that simulating multiscale PDEs encapsulating the specific material behavior is non-trivial using PINNs. Chapter 2 proposes a framework using nondimensional equations in the loss function. The proposed framework is lossless and has no loss of features, converting the dimensional equations to nondimensional and vice versa. The proposed framework is also employed to address multiple connected complex beam systems and ill-posed inverse problems to identify the unknown model parameters and the applied force on the beam components. In addition, the presented methodology is robust to noise. It can

accommodate potential uncertainty in the measurement data and is suitable for real-world applications with incomplete or uncertain data.

- This dissertation introduces a causality-respecting physics-informed deep learning framework in Chapter 3 that simulates complex beam systems in large domains. A causality-respecting PINN loss function effectively addresses these limitations and enforces relevant physics. However, implementing this modified loss function requires a denser neural network with more parameters. This problem is exacerbated when simulating similar beam dynamics problems. A transfer learning-based framework is proposed within the causal PINN architecture to address the complexity and impracticality of simulating at every instant. By leveraging transfer learning, previously trained model parameters are used to initialize and train new models, reducing the computational cost and enabling faster convergence for subsequent tasks, improving the efficiency of simulating beam dynamics on elastic foundations for large-space time domains.
- This dissertation proposes a framework in Chapter 4 to mitigate the issue of out-of-domain predictions in deep learning for solving PDEs. In particular, a physics-informed neural architecture is utilized to learn the underlying dynamics in the training domain, followed by a neural oscillator to exploit the causality and learn temporal dependencies between the solutions at subsequent time levels. The proposed framework based on neural oscillators carries a hidden state that retains information from previous time steps, enabling the model to capture and leverage temporal dependencies in the data. This extension of a physics-informed architecture increases prediction accuracy in out-of-domain scenarios and efficiently extrapolates the dynamics in the time domain without using any data from the untrained time domain. The performance of the proposed approach is evaluated on four different canonical problems, including beam dynamics, demonstrating superior performance compared to other recurrent architectures.
- This dissertation introduces a two-stage approach for generalizing the state-of-the-art (SOTA) traditional numerical-based methods or the experimental data collection-based methods for engineering dynamic simulations. First, dynamics are simulated using a SOTA simulator preferred for the application, followed by classical mathematical models (ODEs) infused in neural architecture to extrapolate the dynamics. Chapter 5 proposes a resolution invariant pipeline where both stages can process data at different resolutions. The proposed workflow efficiently extrapolates the dynamics in the time domain without using any data from the

untrained time domain. Furthermore, the framework eliminates tedious re-meshing and re-simulation in computer-aided simulation software for novel parameters belonging to the parameter space. The performance of the proposed approach is evaluated on four different dynamic simulation problems in structural engineering, including real-world catenary-pantograph systems, demonstrating superior performance compared to other recurrent architectures.

- All codes and datasets used in the research are provided open-source, facilitating the reproduction of the work and open science.

1.5. OUTLINE

This dissertation is organized into six chapters as shown in Fig. 1.2. Chapters 2-5 contribute to advancing the understanding and application of physics-informed machine learning for simulating fundamental components of railway infrastructure, i.e., beam deformation. A flowchart of the dissertation is shown in Figure 1.2, clarifying the relationships between the chapters. Chapter 2 introduces a physics-informed neural network-based methodology for simulating beam dynamics. Chapter 3 presents a causality and transfer learning-based approach for predicting beam deformation in large space and temporal domains. Chapter 4 introduces a methodology for generalizing canonical PDE problems through neural oscillators. Chapter 5 presents out-of-domain prediction, including moving load and catenary contact wire dynamics. Chapter 6 concludes the dissertation with the findings for the research questions and recommendations for future research. A brief description of the remaining chapters is as follows:

Chapter 2 considers the simulation of complex beam dynamics with multiscale coefficient challenge. Depending on the nature of the material, coefficients in the physical equation differ, leading to multiscale coefficient values. Incorporating large coefficient variations in the PDE-based loss function, like in PINN, is challenging to optimize. The chapter introduces nondimensional equations in the PINN loss function for simulating complex beam systems. In addition, the chapter proposes to simulate multiple connected beam systems and solve inverse problems in the presence of noisy data.

Chapter 3 focuses on predicting beam deformation in large space and temporal domains. The chapter proposes using causality-respecting PINN to achieve long rollouts in large spatial domains. In addition, to solve several similar problems and reduce the computational cost, a transfer learning-based methodology is proposed to generalize the dynamics. The numerical experiments are carried out to showcase faster convergence for similar problems, solving problems in larger space-time domains.

Chapter 4 deals with the problem of generalization in physics-informed machine learning. The chapter introduces a neural oscillator-based framework for predicting outside the training domain. The methodology is validated on the canonical PDEs, including beam dynamics, to showcase better generalization abilities than traditional recurrent neural architectures.

Chapter 5 extends the problem of generalization beyond physics-informed machine learning and tackles the prevalent problem for dynamic simulators in general. A two-stage framework is proposed to generalize the simulation dynamics through a neural ODE-based architecture. The numerical experiments, including the catenary contact wire deflection predictions, demonstrate the efficacy of the proposed method in predicting the quantity of interest outside the training domain.

Finally, Chapter 6 concludes this dissertation and recommends future research directions in physics-informed machine learning for engineering structures.

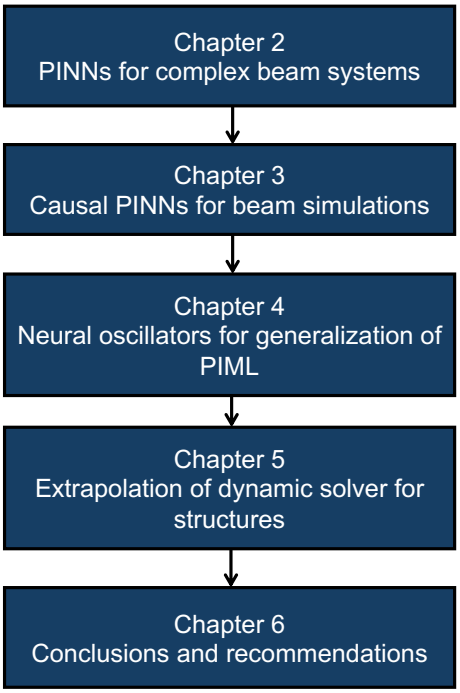


Figure 1.2: Structure of this dissertation.

REFERENCES

[1] Z. Liu, Y. Song, Y. Han, H. Wang, J. Zhang, and Z. Han. “Advances of research on high-speed railway catenary”. In: *Journal of Modern Transportation* 26.1 (2018), pp. 1–23. doi: 10.1007/s40534-017-0148-4.

[2] L. Drugge, T. Larsson, and A. Stensson. “Modelling and simulation of catenary-pantograph interaction”. In: *Vehicle System Dynamics* 33.sup1 (1999), pp. 490–501. doi: 10.1080/00423114.1999.12063106.

[3] J. J. Kalker. *Rolling contact phenomena: linear elasticity*. Springer, 2000. doi: 10.1007/978-3-7091-2782-7.

[4] Y.-B. Yang, J. Yau, Z. Yao, and Y. Wu. *Vehicle-bridge interaction dynamics: with applications to high-speed railways*. World Scientific, 2004. doi: 10.1142/5541.

[5] D. Younesian, A. Hosseinkhani, H. Askari, and E. Esmailzadeh. “Elastic and viscoelastic foundations: a review on linear and nonlinear vibration modeling and applications”. In: *Nonlinear Dynamics* 97.1 (2019), pp. 853–895. doi: 10.1007/s11071-019-04977-9.

[6] A. C. Lamprea-Pineda, D. P. Connolly, and M. F. Hussein. “Beams on elastic foundations–A review of railway applications and solutions”. In: *Transportation Geotechnics* 33 (2022), p. 100696. doi: 10.1016/j.trgeo.2021.100696.

[7] R. R. Craig Jr and A. J. Kurdila. *Fundamentals of structural dynamics*. John Wiley & Sons, 2006.

[8] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440. doi: 10.1038/s42254-021-00314-5.

[9] S. M. Han, H. Benaroya, and T. Wei. “Dynamics of transversely vibrating beams using four engineering theories”. In: *Journal of Sound and Vibration* 225.5 (1999), pp. 935–988. doi: 10.1006/jsvi.1999.2257.

[10] L. C. Evans. *Partial differential equations*. Vol. 19. American Mathematical Society, 2022.

[11] T. Kapoor, H. Wang, A. Nunez, and R. Dollevoet. “Physics-informed neural networks for solving forward and inverse problems in complex beam systems”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023). doi: 10.1109/TNNLS.2023.3310585.

[12] J. Han, A. Jentzen, and W. E. “Solving high-dimensional partial differential equations using deep learning”. In: *Proceedings of the National Academy of Sciences* 115.34 (2018), pp. 8505–8510. doi: 10.1073/pnas.1718942115.

[13] M. Alber, A. Buganza Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, *et al.* “Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences”. In: *Npj Digital Medicine* 2.1 (2019), p. 115. doi: 10.1038/s41746-019-0193-y.

[14] M. Raissi, P. Perdikaris, and G. E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational physics* 378 (2019), pp. 686–707. doi: 10.1016/j.jcp.2018.10.045.

[15] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. “Scientific machine learning through physics–informed neural networks: where we are and what’s next”. In: *Journal of Scientific Computing* 92.3 (2022), p. 88. doi: 10.1007/s10915-022-01939-z.

[16] W. Phusakulkajorn, A. Núñez, H. Wang, A. Jamshidi, A. Zoeteman, B. Ripke, R. Dollevoet, B. De Schutter, and Z. Li. “Artificial intelligence in railway infrastructure: current research, challenges, and future opportunities”. In: *Intelligent Transportation Infrastructure* 2 (2023), liad016. doi: https://doi.org/10.1093/iti/liad016.

[17] Z. Oniszczuk. “Forced transverse vibrations of an elastically connected complex simply supported double-beam system”. In: *Journal of Sound and Vibration* 264.2 (2003), pp. 273–286. doi: 10.1016/S0022-460X(02)01166-5.

[18] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. “Characterizing possible failure modes in physics-informed neural networks”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[19] B. Moseley. “Physics-informed machine learning: from concepts to real-world applications”. PhD thesis. University of Oxford, 2022.

[20] K. Shukla, P. C. Di Leoni, J. Blackshire, D. Sparkman, and G. E. Karniadakis. “Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks”. In: *Journal of Nondestructive Evaluation* 39 (2020), pp. 1–20. doi: 10.48550/arXiv.2005.03596.

[21] S. Wang, S. Sankaran, and P. Perdikaris. “Respecting causality for training physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 421 (2024), p. 116813. doi: 10.48550/arXiv.2203.07404.

[22] P. Lippe, B. S. Veeling, P. Perdikaris, R. E. Turner, and J. Brandstetter. “PDE-refiner: Achieving accurate long rollouts with neural pde solvers”. In: *arXiv preprint arXiv:2308.05732* (2023). doi: 10.48550/arXiv.2308.05732.

[23] F. Chollet. “On the measure of intelligence”. In: *arXiv preprint arXiv:1911.01547* (2019). doi: 10.48550/arXiv.1911.01547.

[24] J. Kim, K. Lee, D. Lee, S. Y. Jhin, and N. Park. “DPM: A novel training method for physics-informed neural networks in extrapolation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 8146–8154. doi: 10.48550/arXiv.2012.02681.

2

PINNS FOR COMPLEX BEAM SYSTEMS

This chapter proposes a new framework using physics-informed neural networks (PINNs) to simulate complex structural systems that consist of single and double beams based on Euler-Bernoulli and Timoshenko theory, where the double beams are connected with a Winkler foundation. In particular, forward and inverse problems for the Euler-Bernoulli and Timoshenko partial differential equations (PDEs) are solved using nondimensional equations with the physics-informed loss function. Higher-order complex beam PDEs are efficiently solved for forward problems to compute the transverse displacements and cross-sectional rotations with less than 1e-3 percent error. Furthermore, inverse problems are robustly solved to determine the unknown dimensionless model parameters and applied force in the entire space-time domain, even in the case of noisy data. The results suggest that PINNs are a promising strategy for solving problems in engineering structures and machines involving beam systems.

Apart from minor updates, this chapter has been published as: Kapoor, T., Wang, H., Núñez, A., & Dollevoet, R. (2023). Physics-informed neural networks for solving forward and inverse problems in complex beam systems. IEEE Transactions on Neural Networks and Learning Systems

2.1. INTRODUCTION

Complex engineering issues in real-life scenarios are often characterized by the connection between various subsystems and uncertainty in behavior caused by internal and external variables and their interactions. Furthermore, the design and maintenance of complex systems, such as engineering structures and machines, is made challenging by the unpredictable collective behaviors and properties of these concurrently operating and interacting components. These issues are typically difficult to analyze through conventional methods [1]. Most of these complex engineering systems are continuous, and partial differential equation (PDE) models are used to characterize and understand their behavior. These PDE models are used to simulate a wide range of engineering phenomena, ranging from multiple beam systems in suspension bridge cables (Timoshenko beam equations)[2] to catenary-pantograph interactions in railways (damped beam equations) [3] to simulating air turbulence that disrupts flight (Navier-Stokes equations) [4, 5], among many others [6–13]. Solutions to governing PDEs enable real challenges such as structural health monitoring [14–16] and optimal structural design [17, 18] to be addressed.

The development of algorithms for diagnostics and prognosis is an issue in maintaining complex engineering systems [1]. Insights could be obtained by solving the forward and inverse problems for the governing PDEs of interest to forecast the behavior of system and minimize unexpected downtimes of complex systems. These equations range in complexity from being extremely nonlinear (Navier-Stokes equation [19]) to incorporating intricate higher-order boundary conditions (fourth-order beam equations [20]). In practice, these equations are too complicated to be solved analytically and must be solved numerically. Numerical methods such as the finite-difference and finite-element methods have been used to approximate the solutions of these PDEs. Despite their success in practice, these methods encounter some difficulties, such as mesh creation, which is more difficult for complex geometries in higher dimensions [21, 22].

In recent years, scientific machine learning, which combines scientific computing with machine learning methodologies to estimate PDEs solutions, has made remarkable developments and has emerged as a viable alternative to the aforementioned numerical methods. The review papers [21, 23, 24] extensively discuss state-of-the-art breakthroughs in scientific machine learning, including works on real-world engineering problems. However, data-driven methods require a large amount of data, which is possibly computationally expensive and susceptible to noise in some engineering systems [25]. One possible way to mitigate the effects of these problems is to utilize the known physical knowledge of the underlying system in the learning procedure [26–28]. Prior physical knowledge could be incorporated into the learning procedure

by collocating the PDE residual at training points, similar to leveraging the physical equation in the training process. The underlying neural networks proposed in [25] are called physics-informed neural networks (PINNs).

PINNs utilize neural networks universal function approximation property [29] and embed the well-posed physical equations modeled by PDEs in the loss function. Prior knowledge of physical principles works as a regularization agent in neural network training, restricting the space of admissible solutions and improving function approximation accuracy. As a result, given some knowledge of the physical features of the problem and some training data, PINN can be utilized to identify a high-fidelity solution. PINNs have already proven to be a very effective paradigm for approximating solutions of PDEs for real-world problems [30, 31], as discussed in the review papers [21, 23].

However, several challenges for PINNs have also been found [32]. One such challenge for PINNs is to learn relevant physical phenomena for more complex problems with large coefficients in the physical equation [33]. A sequence-to-sequence learning task was proposed in [33] as a remedy to this problem. However, this can be computationally expensive when the scale is large. In [34], the importance of using nondimensional equations in the PINN framework was highlighted for cardiovascular blood flow. We build on these works and address the challenge of multiscale complex beam systems. Accordingly, this chapter uses nondimensional PDEs instead of dimensional PDEs in the loss function. This provides a way to simulate realistic physical equations with computational tractability.

Accurate prediction of the dynamics of structures [35] and structural elements, such as plates [36], and beams [37, 38], is crucial in the field of structural engineering. However, measuring quantities of interest in beam systems through lab experiments can prove to be difficult, as it necessitates specialized prototypes, training, and safety during the testing process, increasing the overall cost of the experiment. PINNs offer a simulation-based solution as a mesh-free method that does not require discretizing the domain into a finite number of elements, making it computationally inexpensive compared to numerical methods. PINNs can effectively integrate incomplete or noisy information with prior physical knowledge. The proposed framework converts dimensionalized PDEs to a nondimensionalized form, increasing the suitability for neural networks and enabling the prediction of deflections and rotations for any material, resulting in a more generalizable method.

This chapter provides a framework to simulate complex structural systems consisting of two or more basic structural systems connected by an elastic layer. In particular, the forced vibration of two elastically connected beams is studied, which is commonly encountered in the mechanical, construction, and aeronautical industries [6]. These double-

beam systems in engineering structures have received significant attention in the scientific community and are considered complex systems. Studies have been conducted to predict the dynamics of these systems under various loading and force conditions, such as those found in papers [39–47], among others. These studies include the use of analytical and closed-form solutions [43, 48–51]; however, analytical methods have limitations in applicability, as they may be useful only for specific types of problems and can become complex for systems with many variables or nonlinear equations. Other approaches, such as the state-space method presented in [45, 52], may also be computationally expensive for systems with a large number of states. Additionally, modal analysis methods as presented in [6, 53] have been used to study the natural frequencies and modes of vibration, but they do not provide information on the full response of the system and cannot be used to predict the time-domain response at any instant.

The considered governing equations are modeled using Euler-Bernoulli and Timoshenko theory. In addition to solving the forward problem and computing the physical quantities of interest, we also solve the inverse problem. For the inverse problem, one may not necessarily have complete information about the inputs to the PDEs, such as initial or boundary data, coefficients [54–56] or applied forces. This lack of knowledge makes the forward problem ill-posed, and subsequently, the forward problem cannot be solved uniquely. In this chapter, access to data for quantities of interest is leveraged to determine the PDEs unknown inputs, for instance, the model parameters and applied forces.

The main contributions of the current chapter are as follows,

- To the best of the authors knowledge, this is the first work to use physics-informed machine learning to solve the forward and inverse problems of Euler-Bernoulli and Timoshenko complex beam models.
- We address a challenge for PINNs in solving multiscale complex beam PDEs and propose a framework for using nondimensional equations in the loss function.
- The proposed nondimensional PINN framework is employed to address ill-posed inverse problems for complex systems and to identify the unknown model parameters and the applied force on the beam components. This is achieved by utilizing data from indirect measurements such as the displacement and cross-sectional rotations of the beams.
- The presented methodology is robust to noise and can accomodate potential uncertainty in the measurement data, making it well suited for real-world applications where data are incomplete or uncertain.

The rest of the chapter is organized as follows. In Section II, the PINN method is presented to simulate the dimensional Euler-Bernoulli beam equation. Due to the limitations of PINNs in simulating the dimensional Euler-Bernoulli beam equation, an alternative approach of using nondimensional equations in the PINN loss function is proposed and successfully used to solve the dimensionless Euler-Bernoulli equation in Section III. Section IV first applies the proposed framework to simulate the Timoshenko beam model for solving forward and inverse problems. The forward problem of the Euler-Bernoulli double-beam equation is then solved. Additionally, Section IV covers forward and inverse Timoshenko double-beam system problems. Section V concludes this chapter.

2.2. PINNS FOR DIMENSIONAL PDES

In this section, the method of PINNs to simulate PDEs is presented in brief using an abstract dimensional PDE. The method is then used to simulate the dimensional Euler-Bernoulli equation. The following abstract dimensional PDE is considered with implicit initial and boundary conditions:

$$\bar{\mathcal{K}}(\bar{x},\bar{t}) := \mathcal{D}[\bar{u}](\bar{x},\bar{t};\bar{\lambda}) - \bar{f}(\bar{x},\bar{t}) \quad \forall (\bar{x},\bar{t}) \in \bar{\Omega} \times \bar{T} \subset \mathbb{R}^{\mathrm{d}} \times \mathbb{R} \tag{2.1}$$

where $\mathcal{D}[\cdot]$ denotes the differential operator, $\bar{u}$ is the quantity of interest, $\bar{x} \in \bar{\Omega} \subset \mathbb{R}^{\mathrm{d}}$, $\bar{t} \in \bar{T} \subset \mathbb{R}$ for $d \geq 1$, $\bar{\Omega}$ denotes the spatial boundary contained in the d-dimensional Cartesian spatial space and $\bar{T}$ denotes the temporal domain, $\bar{\lambda} \in \mathbb{R}$ is the model parameter, $\bar{f}(\bar{x},\bar{t})$ is the external force, and $\bar{K}$ is the notation for the abstract physical equation.

Deep neural networks are the core for PINNs in which inputs $(\bar{x},\bar{t})$ map to output $(\bar{u})$ through an iterative composition of hidden layers. The composition consists of weights (w), biases (b), and linear or nonlinear activation function(s) (σ). The inputs undergo a linear composition within a neuron, where they are multiplied by respective weights and summed along with a bias term. Subsequently, this combined input is passed through a nonlinear activation function (σ) as presented in Fig. 2.4. This allows the neural network to introduce nonlinearity, enabling the network to capture intricate relationships between inputs and outputs.

To train the neural network, one needs training set (Δ), consisting of spatial boundary points (Δ_{b}), temporal boundary points (Δ_{i}) and interior points (Δ_{int}). As a result, the training set can be written as $\Delta = \Delta_{\mathrm{i}} \cup \Delta_{\mathrm{b}} \cup \Delta_{\mathrm{int}}$. In this work, Δ_{i}, Δ_{b}, and Δ_{int} are considered to have N_{i}, N_{b} and N_{int} training points respectively. The total number of training points is denoted by N_{train}. To approximate the quantity of interest $\bar{u}$, one needs to minimize the loss function containing the physical model in the form of a PDE with initial and boundary conditions of (2.1). No

additional data are required in the loss function for forward problems. The loss function $\bar{\mathcal{L}}$ is defined as follows:

$$\bar{\mathcal{L}}(\theta) = \underset{\theta}{\text{Min}}\left(\frac{1}{N_{\text{train}}}\sum_{n=1}^{N_{\text{train}}} ||\bar{\mathcal{K}}(\bar{x}_{\text{n}}, \bar{t}_{\text{n}})||^2\right) \tag{2.2}$$

where $(\bar{x}_{\text{n}}, \bar{t}_{\text{n}})$ represents the training tuple for each n. Minimizing this loss function using a suitable optimization algorithm provides optimal parameters $\theta = \{w, b\}$.

Now, we employ the PINN algorithm for the dimensional Euler-Bernoulli beam equation and evaluate the corresponding performance. The dynamic Euler-Bernoulli beam equation is given by

$$\rho A \bar{u}_{\bar{t}\bar{t}} + EI\bar{u}_{\bar{x}\bar{x}\bar{x}\bar{x}} = \bar{f}(\bar{x}, \bar{t}) \quad \bar{x} \in [0, \bar{l}], \bar{t} \in [0, t_{\text{end}}] \tag{2.3}$$

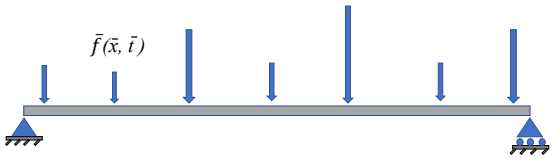


Figure 2.1: Simply supported beam with varying transverse force.

Here, $\bar{l}$ and t_{end} refer to the length of the beam and final time, respectively. This equation models the transverse displacement of beam $\bar{u}$ in the space-time domain subject to the external transverse force $\bar{f}$ as shown in Fig. 2.1. This work considers a uniform cross-sectioned beam with constant material properties throughout the beam. The parameters ρ and A denote the density and cross-sectional area of the beam, respectively. The parameters E and I are Young's modulus and the moment of inertia of the beam, respectively. The external force $\bar{f}$ acts nonuniformly on the body, and $\bar{u}$ is the transverse displacement of the beam, which is the only unknown in the governing PDE. In addition, u_{tt} represents the second order partial derivative of u with respect to t, and u_{xxxx} represents the fourth order partial derivative of u with respect to x. The goal of the forward problem is to compute the transverse displacement of the beam supplemented with the initial and boundary conditions. For this study, simply supported beams are considered, which rest on two supports and are free to move horizontally. Real-world applications of simply supported beams include railway tracks, and bridges, to name a few. Mathematically, the simply supported boundary condition for (2.3) is given by

$$\bar{u}(0, \bar{t}) = \bar{u}(\bar{l}, \bar{t}) = \bar{u}_{\bar{x}\bar{x}}(0, \bar{t}) = \bar{u}_{\bar{x}\bar{x}}(\bar{l}, \bar{t}) = 0$$

For the numerical experiment, the parameter values of aluminium-like material are considered in the physical equation, which are widely used for making beams. The parameter values taken for the problem are $\rho = 2 \times 10^3 \text{kg/m}^3$, $A = 5 \times 10^{-2}$ m^2, $E = 10^{10}\text{N/m}^2$, and $I = 4 \times 10^{-4}\text{m}^4$. Additionally, the beam is taken to be π^2 meters long, and the external force $\bar{f}$ is taken to be $EI(1-16\pi^2)\sin(\bar{x}/\pi)\cos(4c\bar{t}/\pi)/\bar{l}^3\text{N}$, where $c = \sqrt{\frac{EI}{\rho A}}$. Taking the final time to be $\pi^2/200$, the PDE to be solved takes the form

$$10^2 \bar{u}_{\bar{t}\bar{t}} + 4 \times 10^6 \bar{u}_{\bar{x}\bar{x}\bar{x}\bar{x}} = 4 \times 10^6(1-16\pi^2)\sin(\bar{x}/\pi)\cos(800\bar{t}/\pi)/\pi^3 \quad (2.4)$$

in the domain $\bar{x} \in [0, \pi^2]$ and $\bar{t} \in [0, \pi^2/200]$. For (2.4) to be well-posed the initial condition of the beam is taken to be $\sin(\bar{x}/l)$ with zero initial velocity, where $l = \sqrt{\bar{l}}$.

For training the neural network, 16000 random training points are generated with the distribution $N_i = 2000$, $N_b = 4000$, and $N_{int} = 10000$. The neural network consists of 4 hidden layers with 20 neurons in each hidden layer. The tanh activation function, which is one of the most commonly used activation functions in the PINN literature, as described in the review paper [23], is chosen. The loss function (2.2) consists of the initial condition, boundary condition and PDE. The PDE is regularized in the loss function with the residual parameter 0.1 [57]. The L-BFGS optimizer, which is again one of the most commonly used optimizers in the PINN literature [23] is used to minimize the loss function. As shown in Fig. 2.2, 15000 epochs are performed. However, the figure clearly illustrates that the optimizer does not converge to the solution, and a vast training loss of 10^{14} is obtained. Additionally, the graph shows that the optimizer is stuck in the local minima and hence will not converge even if the number of epochs is increased for the same neural network configuration.

In [16, 58], the problem of free vibrations in the Euler-Bernoulli single-beam equation was successfully solved by PINNs, where the coefficients of the PDE were taken to be unity. This shows that PINNs can simulate the beam equations, and the challenge lies in the multiscale coefficient values that arise when dealing with a real-life physical equation. The nonconvergence in our case is due to the high value of coefficients, which is due to the dimensional equation. Consequently, a pressing need arises to transform the dimensional form of the equation into a nondimensional form. It may be possible that for some configurations containing hundreds of hidden layers and neurons, this problem may be solved without the need to non-dimensionalizing the PDE. However, nondimensionalization aims to provide computational tractability.

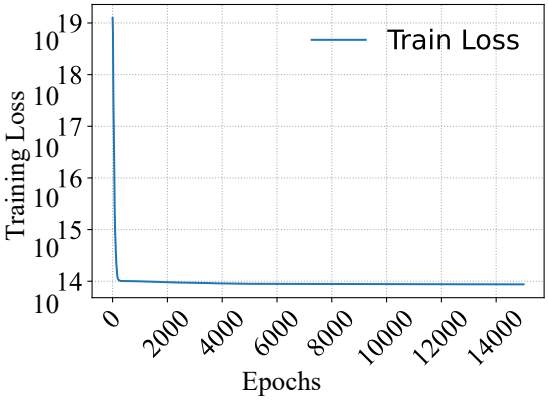


Figure 2.2: L-BFGS training loss vs. the number of epochs for the dimensional Euler-Bernoulli beam equation.

2.3. PINNS FOR NONDIMENSIONAL PDES

This section presents the proposed framework of using nondimensional equations in the PINN loss function. The method for nondimensionalizing the governing PDE is described first. Then, the algorithms for forward and inverse problems using dimensionless equations in PINNs are presented. To nondimensionalize the abstract PDE given by (2.1), the following transformations are performed

$$\bar{x} = \xi_1(x); \quad \bar{t} = \xi_2(t); \quad \bar{u} = \xi_3(u); \quad \bar{f} = \xi_4(f) \tag{2.5}$$

where, ξ_1, ξ_2, ξ_3, and ξ_4 are suitable functions that map the dimensional quantities $\bar{x}$, $\bar{t}$, $\bar{u}$, and $\bar{f}$ to the corresponding nondimensional quantities. After substituting the above transformations in (2.1) and introducing the dimensionless parameter λ, one obtains

$$\mathcal{K}(x,t) := \mathcal{D}[u](x,t;\lambda) - f(x,t) \quad \forall (x,t) \in \Omega \times T \subset \mathbb{R}^d \times \mathbb{R} \tag{2.6}$$

The proposed framework uses dimensionless equations to simplify and stabilize the problem computationally. By nondimensionalizing the variables and parameters, they are kept within a specific range, resulting in improved performance and generalization of the neural network. Furthermore, dimensionless equations generate more interpretable solutions by eliminating the units of measure, making it easier to understand the underlying physical phenomena and to compare results across different physical systems in the form of ratios and parameters. Hence, using dimensionless equations in PINNs can enhance the computational stability, generalization, and interpretability of the neural network.

2.3.1. PINN FRAMEWORK FOR FORWARD PROBLEMS

$\mathcal{K}$, the nondimensional PDE corresponding to the dimensional PDE $\bar{\mathcal{K}}$, is now used in the loss function $\mathcal{L}$ defined as follows:

$$\mathcal{L}(\theta) = \underset{\theta}{\text{Min}}\left(\frac{1}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} ||\mathcal{K}(x_n, t_n)||^2\right) \tag{2.7}$$

A schematic representation of the proposed PINN-based framework is

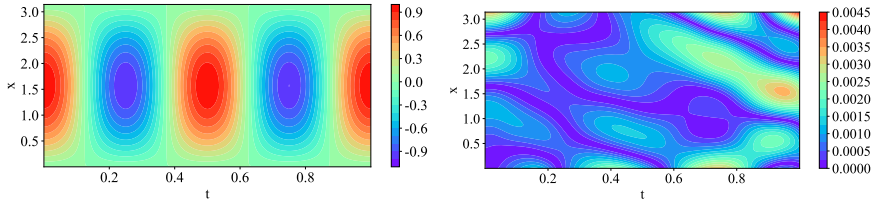


Figure 2.3: Nondimensional Euler-Bernoulli beam equation Color bar represents **Left:** Predicted solution (u^*); **Right:** Absolute error in prediction ($|u - u^*|$)

illustrated in Fig 2.4.

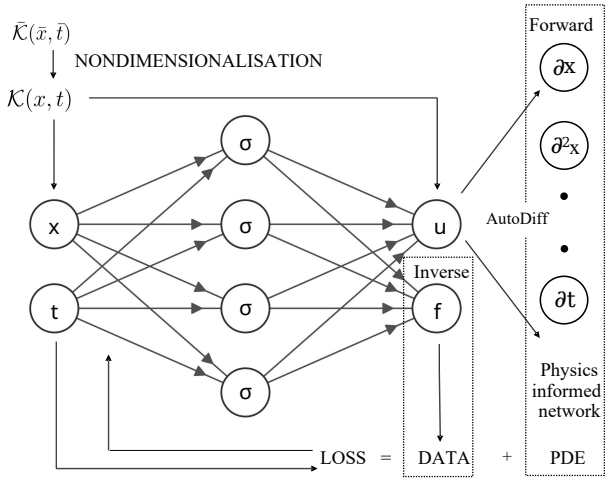


Figure 2.4: PINN framework for beam systems: For forward problems, the loss function comprises the nondimensional PDEs and the boundary and initial conditions. For inverse problems, the nondimensional PDEs are supplemented with extra data and potential initial/boundary conditions.

2.3.2. NONDIMENSIONAL EULER-BERNOULLI BEAM EQUATION

We now test the nondimensional equation in the PINN framework and evaluate the corresponding performance. To nondimensionalize (2.3), following transformations are used:

$$u = \bar{u}/l; \quad x = \bar{x}/l; \quad t = c\bar{t}/l^2; \quad f = \bar{f}l^3/(EI) \tag{2.8}$$

Upon substituting these values in (2.3), one obtains

$$u_{tt} + u_{xxxx} = f(x,t) \quad x \in [0,\pi], t \in [0,1] \tag{2.9}$$

where $f(x,t) = (1-16\pi^2)\sin(x)\cos(4\pi t)$, with initial and boundary conditions

$$u(x,0) = \sin(x), \quad u_t(x,0) = 0$$
$$u(0,t) = u(\pi,t) = u_{xx}(0,t) = u_{xx}(\pi,t) = 0$$

For the error estimation, the relative percentage error ($\mathcal{R}$) used in [57] is chosen. Here, u^* is the prediction and u is the analytical solution.

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

The same neural network architecture as the previous case is chosen to solve this resulting nondimensional PDE. A low training loss is obtained, indicating that the PINN is trained successfully. The analytical solution for this case is $u(x,t) = \sin(x)\cos(4\pi t)$, which is used to quantify the error in the approximated solution. The nondimensional displacement of the Euler-Bernoulli beam is computed within $\mathcal{R} = 5.3e-4$ percent. The nondimensional displacement prediction using PINN is shown in Fig. 2.3.(a). Fig. 2.3.(b) shows the absolute error between the exact and predicted solutions.

The contour plot for the approximate solution shows the dynamics of a simply supported beam under a force, where the x-axis represents the time, the y-axis represents the position along the length of the beam, and the colors represent the displacement of the beam. In Fig. 2.3.(b) the red regions indicate high displacement, while the blue regions indicate low displacement. There is a strong displacement at the position of the beam when a substantial force is applied, which is consistent with the known physics of this system. The network accurately captures the displacement behavior of the beam, which is evident by the smooth and continuous transition of colors across the plot.

The contour plot for the error in Fig. 2.3.(b) shows the difference between the approximate solution obtained from the network and the true solution. The x-axis represents the time, the y-axis represents the position along the length of the beam, and the colors represent the error.

The red regions indicate high error, while the blue regions indicate low error. The areas where the training point concentration is low account for more error, and areas where the concentration of training points is more have relatively low error. One approach to reduce the error is to have more training points in the regions of high error. However, the overall error is low, which indicates that the network accurately captures the displacement behavior of the beam.

From Fig. 2.3.(b), the PINNs are found to solve the dimensionless Euler-Bernoulli beam equation accurately and hence, for all further experiments, nondimensional PDEs are simulated using PINNs. Additionally, the nondimensional displacement is henceforth referred to as displacement for conciseness. The presented methodology predicts the dimensionless quantities and hence all the plots of results and their associated error plots are dimensionless. Consequently no units are mentioned in the plots of the presented results. Next, we describe the inverse problem-solving strategy using nondimensional equations.

2.3.3. PINN FRAMEWORK FOR INVERSE PROBLEMS

Algorithm 1 Inverse PINN algorithm

Goal: To predict the unknown parameter $\bar{\lambda}$ or function $\bar{f}(\bar{x}, \bar{t})$.

Step 1: Nondimensionalize the governing PDE to approximate the dimensionless parameter λ or function $f(x, t)$.

Step 2: Choose the training set from the space-time domain $\Omega \times T$, and augment with $(x_{\text{data}}, t_{\text{data}})$ at which additional data (u_{data}) are provided.

Step 3: Construct a feedforward deep neural network with inputs (x, t) and outputs u, λ or $f(x, t)$.

Step 4: Minimize the loss function (2.11) with a suitable optimization algorithm, and find the optimal parameters.

Step 5: Use the optimal parameters to approximate the parameter λ^* or the function $f^*(x, t)$.

The abstract dimensionless PDE described by (2.6) is well-posed, and the forward problem can be solved uniquely. However, in the case of an inverse problem, the problem is ill-posed and either the initial/boundary conditions or the parameters/forces are unknown. Hence the generic abstract PDE can be re-written as

$$\mathcal{K}'(x, t) := \mathcal{D}[u](x, t; \lambda) - f(x, t) \quad \forall (x, t) \in \Omega \times T \subset \mathbb{R}^d \times \mathbb{R} \tag{2.10}$$

The algorithm for the PINN framework is presented to solve inverse problems.

The aim of the inverse problem is to predict the unknown parameter λ or the force function $f(x,t)$, when data are provided for the observable u in some part of the training domain. In this chapter, u_{data} denotes the available data for the inverse problem at N_{data} points. The prediction of the unknown parameter requires additional information in the loss function as shown in Fig 2.4. It is essential for the Jacobian matrix utilized in the inverse operation study employing neural networks to exhibit a nonzero determinant, to be invertible, and to possess a reasonable ratio between its largest and smallest eigenvalues to guarantee a unique solution and ensure computational stability. The algorithm for the inverse problem is the same as for the forward problem with a minor modification in the loss function. In addition to the output u, the PINNs now predict the unknown parameter, force, initial or boundary conditions of the physical problems by leveraging the known data. The loss function for the inverse problem can be defined as

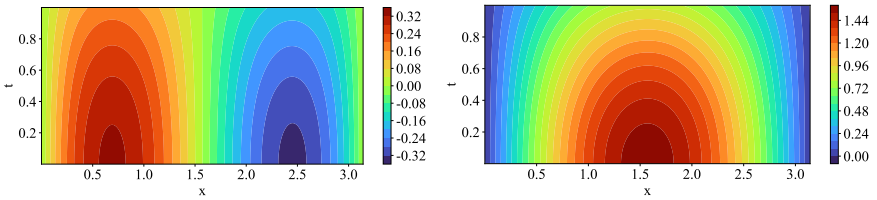


Figure 2.5: Timoshenko single beam; Color bar represents **Left:** Cross-sectional rotation (θ^*); **Right:** Transverse displacement (w^*).

$$\mathcal{L}'(\theta) = \underset{\theta}{\text{Min}}\Big(\frac{1}{N_{\text{train}}}\sum_{n=1}^{N_{\text{train}}} ||\mathcal{K}(x_n,t_n)||^2 + \frac{1}{N_{\text{data}}}\sum_{n=1}^{N_{\text{data}}} ||u_{\text{data}}(x_n,t_n) - u_{\text{pred}}(x_n,t_n)||^2\Big) \quad (2.11)$$

Here, u_{pred} denotes the prediction of u by the neural network section implementing the PINN algorithm for forward and inverse problems of dimensionless beam equations.

2.4. NUMERICAL EXPERIMENTS AND DISCUSSION

In the following subsections, five numerical experiments are presented. The experiments are conducted in a progressive manner, beginning with

simple models such as a single beam system and then progressing to more complex ones such as a double beam connected to a Winkler foundation. To verify the proposed method, we first investigate forward and inverse problems for a single beam, which serves as the proof of the concept. Then, we apply the method to more intricate cases of double-beam systems to simulate forward and inverse problems.

2.4.1. TIMOSHENKO BEAM FORWARD PROBLEM

The Euler-Bernoulli theory of beams is widely used in the literature and has been successfully applied in structures such as the Eiffel Tower and Ferris wheels. However, it does not consider the effects of transverse shear deformations, which are often significant in the vertical displacements of short and thick beams [59]. Timoshenko beam theory provides a mathematical framework for analyzing thick-beam bending [59]. According to Timoshenko theory, upon the action of an external force, the beam undergoes some cross-sectional rotation in addition to transverse displacement. Mathematically, the dynamics are modeled by a coupled system of PDEs with two variables: transverse displacement and cross-sectional rotation. The model is given by

$$\begin{aligned} \rho I \bar{\theta}_{\bar{t}\bar{t}} - EI\bar{\theta}_{\bar{x}\bar{x}} - kAG(\bar{w}_{\bar{x}} - \bar{\theta}) &= 0 \\ \rho A \bar{w}_{\bar{t}\bar{t}} - kAG(\bar{w}_{\bar{x}\bar{x}} - \bar{\theta}_{\bar{x}}) &= \bar{g}(\bar{x}, \bar{t}) \end{aligned} \tag{2.12}$$

where ρ, A, E and I have the usual meaning as in the case of the Euler-Bernoulli beam; k is called the Timoshenko shear coefficient; G is the shear modulus; and $\bar{g}(\bar{x}, \bar{t})$ is the external force acting on the beam. The transverse displacement is $\bar{w}(\bar{x}, \bar{t})$ and $\bar{\theta}(\bar{x}, \bar{t})$ is the cross-sectional rotation of the beam at position $\bar{x}$ and time $\bar{t}$. After nondimensionalizing (2.12) and taking the resulting parameters [60] to be unity, the nondimensional equation can be written as follows:

$$\begin{aligned} \theta_{tt} - \theta_{xx} + (\theta - w_x) &= 0 \\ w_{tt} + (\theta - w_x)_x &= g(x, t) \end{aligned} \tag{2.13}$$

We consider the external force [61] to be $g(x,t) = \cos(t) - \frac{\pi}{2}\sin(x)\cos(t)$ and the computational domain to be $x \in [0, \pi]$ and $t \in [0, 1]$. To make (2.13) well-posed, the initial and boundary conditions are supplemented as:

$$\theta(x,0) = \frac{\pi}{2}\cos(x) + \left(x - \frac{\pi}{2}\right), \quad \theta_t(x,0) = 0$$

$$w(x,0) = \frac{\pi}{2}\sin(x), \quad w_t(x,0) = 0$$

$$\theta(0,t) = \theta(\pi,t) = w(0,t) = w(\pi,t) = 0$$

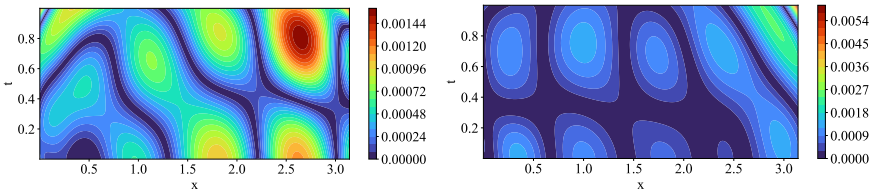


Figure 2.6: Timoshenko single beam absolute error in predictions **Left:** $|\theta - \theta^*|$; **Right:** Absolute error $|w - w^*|$.

To estimate the error in the approximated solutions, the analytical solution for the considered problem is used, which is

$$\theta(x,t) = \left(\frac{\pi}{2}\cos(x) + \left(x - \frac{\pi}{2}\right)\right)\cos(t)$$

$$w(x,t) = \frac{\pi}{2}\sin(x)\cos(t)$$

When analytical solutions are not available, there are various ways to validate the PINN solution. One approach is to compare the solutions with those obtained using numerical methods such as finite difference, finite element, finite volume or spectral methods. This can be done by comparing the predicted solutions from the PINNs with the solutions from the numerical simulation for the same physical equation. Another approach is to compare the solutions obtained through PINNs with experimental data. One can compare the predicted solutions from the PINNs with values experimentally measured over space and time. Finally, one can validate the solutions obtained through PINNs by checking if they satisfy the known physical constraints of the system. In summary, one can use available experimental data, numerical methods or physical constraints to evaluate the accuracy of the solution obtained using PINNs.

The difficulty of solving a system of PDEs is greater than that solving a single PDE, but the neural network structure used for the Euler-Bernoulli equation is successful in approximating solutions for Timoshenko beams. In particular, the transverse displacement of the beam is computed within $\mathcal{R} = 3.3e-4$ percent, and the cross-sectional rotation is approximated within $\mathcal{R} = 2.8e-3$ percent. Approximated solutions and absolute errors in predicting the transverse displacement and cross-sectional rotation are presented in Figs. 2.5 and 2.6. Fig. 2.5 demonstrates that when a sinusoidal force is applied to a Timoshenko beam, the beam bends more than it rotates. As indicated by the scale in the figures, the maximum deflection is 1.44 and the maximum rotation is 0.32. Additionally, the low error in predictions demonstrates that even with the increase in the PDE complexity, the PINN successfully solves the Timoshenko PDE with comparable results to the Euler-Bernoulli equation.

We compare the results obtained from our method with three other methods. The first method we consider is the widely used numerical technique called the finite difference method (FDM). The other two methods are neural network-based approaches, namely physics-guided neural networks (PGNN) [28, 62–65] and gradient-enhanced physics-informed neural networks (gPINN) [66]. First, for FDM we employ a central difference scheme to approximate space derivatives and a leapfrog scheme to approximate time derivatives. This approach allows us to solve problems with second-order accuracy in space and time. The results for the Timoshenko beam show that PINNs can achieve a higher level of accuracy than the FDM even with a smaller number of training points. Specifically, $30,000$ points are used in the FDM scheme while only $16,000$ points were used for training with PINNs and Table 2.1 indicates that PINNs perform better than FDM.

Second, the performance of PINN is compared to a neural network-based approach PGNN, which leverages physical knowledge embedded in the available data, for instance, the relationship between beam acceleration and displacement for the Timoshenko beam problem. Accelerometers can be employed at discrete locations along the beam to obtain acceleration data. Acceleration data at five equidistant points along the beam are used, with 2000 data points at each location. This dataset is augmented with the boundary and initial conditions of displacement to match the training data size of PINN. PGNN is a deep neural network-based architecture with inputs: position (x), time (t), and acceleration. Displacement (w) is taken as the output of this neural network. Training PGNN with identical hyperparameters to those used in PINN, PGNN predicts the displacement (w) with an error of approximately 0.002739%, as shown in Table 2.1.

Furthermore, utilizing the displacement values (w), auto differentiation and (2.13), we derived θ_x. Subsequently, a second neural network was constructed to predict θ, where θ_x is used as the input. Boundary and initial conditions for cross sectional rotation (θ) are also used to guide the PGNN towards the optimal solution. After training the PGNN, cross-sectional rotation is predicted with approximately 3.486727% error. It can be inferred from Table 2.1 that both displacement and rotation predictions exhibited higher errors than PINN. This discrepancy can be attributed to the restricted availability of acceleration data at only discrete spatial locations within the interior domain rather than a random distribution across the entire domain. Furthermore, the second neural network, employed for rotation prediction, demonstrated inferior performance potentially due to error propagation.

Third, we perform another comparison with a neural network-based method to simulate PDEs, gradient-enhanced PINN (gPINN) [66], which differs from PINN in terms of the loss function. The acronym "gPINN" proposed in [66] is used in this work instead of "GPINN" as it is

used for another method [67]. In addition to the loss function of PINN, gPINN leverages gradient information of the PDE residual and embed the gradient into the loss function. For the Timoshenko beam problem, derivatives of the system of PDE (2.13) with respect to space (x) and time (t) are supplemented in the loss function. Table 2.1 shows that gPINN exhibits higher relative error percentages in learning displacement and cross-sectional rotation than PINN. The high-order derivatives of the physical equations in the loss function of gPINN make it challenging for autodifferentiation [68] and backpropagation of the loss function, resulting in poor predictions of deflection and rotation for the Timoshenko beam. Table 2.1 demonstrates that PINN outperforms FDM, PGNN, and gPINN in accurately predicting displacement and cross-sectional rotation for the Timoshenko beam, emphasizing its superior performance compared to the three alternative methods.

Table 2.1: Timoshenko beam: $\mathcal{R}$ at $t = 1$

w, θ	PINN	FDM	PGNN	gPINN
w (%)	3.3e-4	0.005615	0.002739	0.249849
θ (%)	2.8e-3	0.004733	3.486727	5.498449

2.4.2. TIMOSHENKO BEAM INVERSE PROBLEM

This section addresses the inverse problem for the Timoshenko beam, with the aim to determine the material properties of a beam leveraging the PDE and displacement and rotation data of the beam. In structural engineering, the inverse problem of a Timoshenko beam PDE is significant for determining the structural behavior and for health monitoring of beam systems. This helps engineers infer the internal material properties and unknown forces from observed responses such as displacement and rotation measurements. The PINN solves this problem by combining the knowledge of physics and deep learning. The PINN uses a neural network to learn the mapping between the unknown parameters of the PDE and observed data while incorporating the constraints of physics in the form of PDEs. This parameter identification aids in providing crucial information for structural diagnosis and repair and helps engineers ensure the safety and stability of structures. The Timoshenko model for parameter estimation is presented as follows.

$$\begin{aligned} \alpha\theta_{tt} - \theta_{xx} + (\theta - w_x) &= 0 \\ w_{tt} + (\theta - w_x)_x &= g(x,t) \end{aligned} \tag{2.14}$$

In the context of the inverse problem of the Timoshenko beam, the PINN is trained on the observed deflections and rotations of the

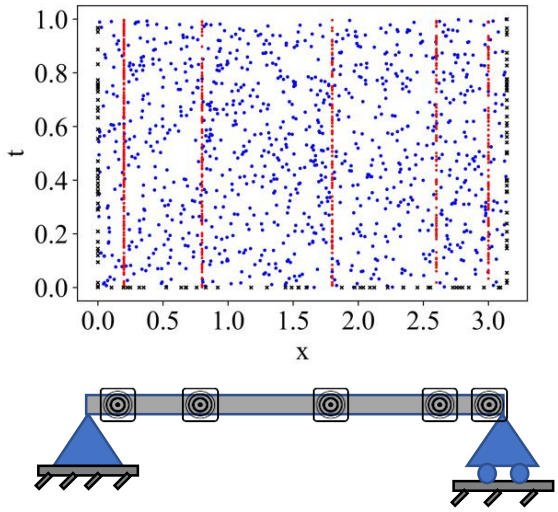


Figure 2.7: Data to learn the parameters for the Timoshenko single-beam: **Blue dots** Collocation points. **Red dots** Additional data points of rotations (θ) and displacement (u). **Black dots** Initial and boundary points.

beam, and the material properties are treated as the unknowns to be estimated. In this case, the force g(x, t) applied to the beam is considered to be known, and the only unknown in the model is α. This makes the problem ill-posed, requiring additional data at a priori to predict the unknown parameter. For $\alpha = 1$, the transverse displacement and cross-sectional rotation data obtained from the forward problem is supplied to approximate the parameter value. This data is not error-free and comes with 10^{-3} percent error for transverse displacement and with 10^{-4} percent error for cross-sectional rotation. As shown in Fig. 2.7, the additional data is supplied on 5000 points (red dots) at five positions on the beam ($x = 0.2, 0.8, 1.8, 2.6, 3$). In practice, this data can be collected using sensors installed at the corresponding locations on the beam as shown in Fig. 2.7.

To solve the inverse problem, the neural network consist of 1600 random training points with the distribution $N_{\text{i}} = 200$, $N_{\text{b}} = 400$, and $N_{\text{int}} = 1000$. To regularize the PDE term in the loss function, a regularization parameter of 1 was chosen [25]. Using the L-BFGS optimizer 5000 iterations are performed and the other parameters are kept the same as in the forward Timoshenko problem. At $t = 0.5$, the unknown parameter $\alpha = 1.0136$ is learned.

We perform a comparison between the PINN and DNNs, as using

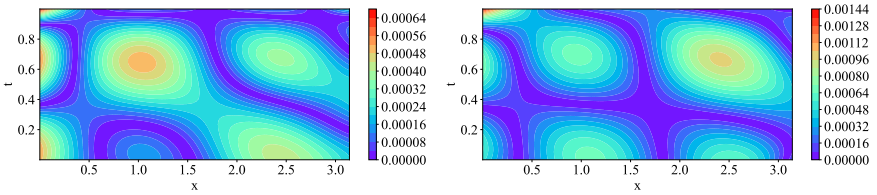


Figure 2.8: Euler-Bernoulli double-beam: color bar represents absolute error in predictions **Left:** $|w_1 - w_1^*|$; **Right:** $|w_2 - w_2^*|$.

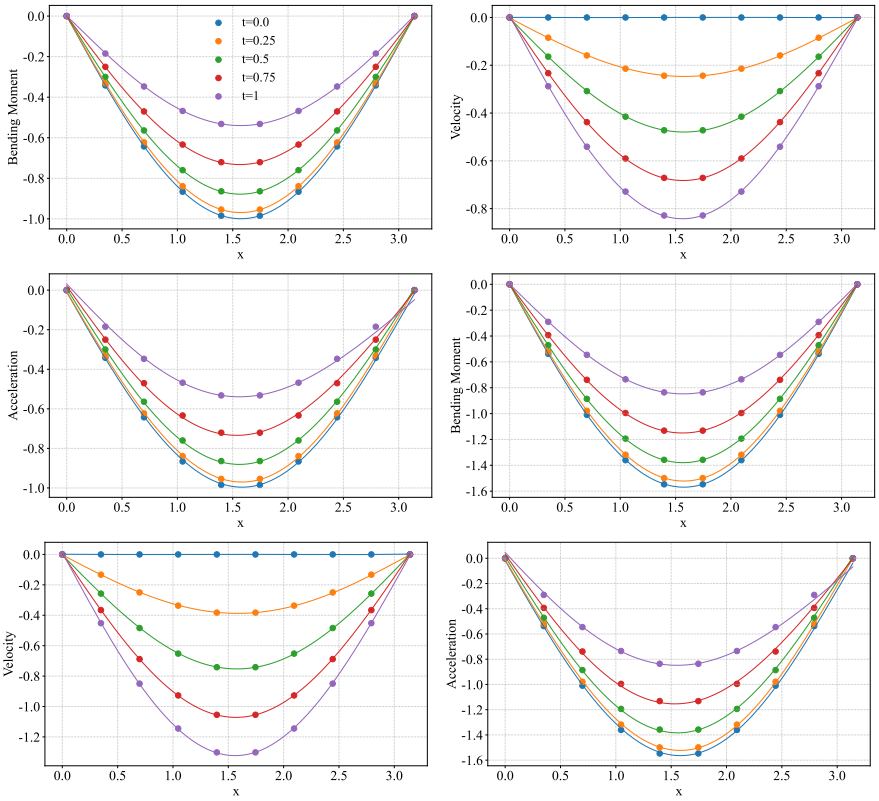


Figure 2.9: Derived quantities for the Euler-Bernoulli double beam. Scattered points represent the exact solution and the continuous line refers to the derived solution. **Top:** First beam **Left** Bending moment; **Mid** Velocity; **Right** Acceleration. **Bottom:** Second beam **Left** Bending moment; **Mid** Velocity; **Right** Acceleration.

a numerical iterative method for inverse problems is computationally

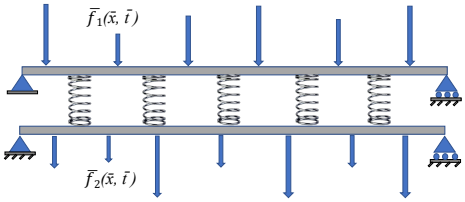


Figure 2.10: Double beam system connected by a Winkler foundation.

expensive. From PINNs, at $t = 0.5$, the unknown parameter $\alpha = 1.0136$ is learned. We utilize DNNs to identify the parameters of a Timoshenko single beam. We use the same architecture for DNN as used by the PINN. The predicted value of alpha is 0.6124 using DNN. PINN is more accurate than DNNs for the inverse problem of beam systems.

However, there are several issues that one may need to take care of while solving inverse problems through the presented framework. First, to avoid overfitting, the minimum training data points required to solve the problem should be determined empirically by gradually increasing the number of training points until the model performance is satisfactory. Second, for some physical problems, noisy data may lead to nonconvergence of the optimization algorithm. Hence, suitable filtering or preprocessing of data may be required before using the PINN framework. Finally, for every run of the neural network, one may learn a different parameter or function value; due to the convergence of the optimizers at different local minima, it may be useful to find the statistics of the inverse problem solution through multiple runs.

Experimental results for single beam equations illustrate that PINNs can efficiently solve forward and inverse problems for single beams. In this study, we investigate the ability of PINNs to handle more complex systems, specifically double-beam systems connected by a Winkler foundation, as depicted in Fig. 2.10.

2.4.3. EULER-BERNOULLI DOUBLE-BEAM FORWARD PROBLEM

In this section, and for all further experiments, forced transverse vibrations of two parallel beams are studied. Structurally, two parallel beams of equal lengths joined by a Winkler massless foundation are considered. Both beams are considered slender and have homogeneous material properties. The transverse displacement of both beams is governed by the following system of PDEs [41]:

$$\begin{aligned} m_1 \bar{w}_{1_{\bar{t}\bar{t}}} + K_1 \bar{w}_{1_{\bar{x}\bar{x}\bar{x}\bar{x}}} + k(\bar{w}_1 - \bar{w}_2) &= \bar{f}_1(\bar{x}, \bar{t}) \\ m_2 \bar{w}_{2_{\bar{t}\bar{t}}} + K_2 \bar{w}_{2_{\bar{x}\bar{x}\bar{x}\bar{x}}} + k(\bar{w}_2 - \bar{w}_1) &= \bar{f}_2(\bar{x}, \bar{t}) \end{aligned} \tag{2.15}$$

Here, $\bar{w}_1$ and $\bar{w}_2$ are the beam displacements for the first and the second beams respectively. The distributed continuous forces acting transversely on the beams are $\bar{f}_1$ and $\bar{f}_2$ as shown in Fig. 2.10. The product of the density and the cross-sectional area of the beams is given by $m_1 = \rho_1 A_1$ for the first beam and $m_2 = \rho_2 A_2$ for the second beam. The parameters K_1 and K_2 denote the flexural rigidity of the beams and are given by $K_1 = E_1 I_1$ and $K_2 = E_2 I_2$. The stiffness modulus of the Winkler elastic layer connecting both beams is given by k. For simplicity, we consider $m_1 = m_2$, and $K_1 = K_2$, and nondimensionalize (2.15). After taking all the resulting parameters to be unity, the nondimensional equation has the same form as (2.15) with unit coefficients. The initial conditions are

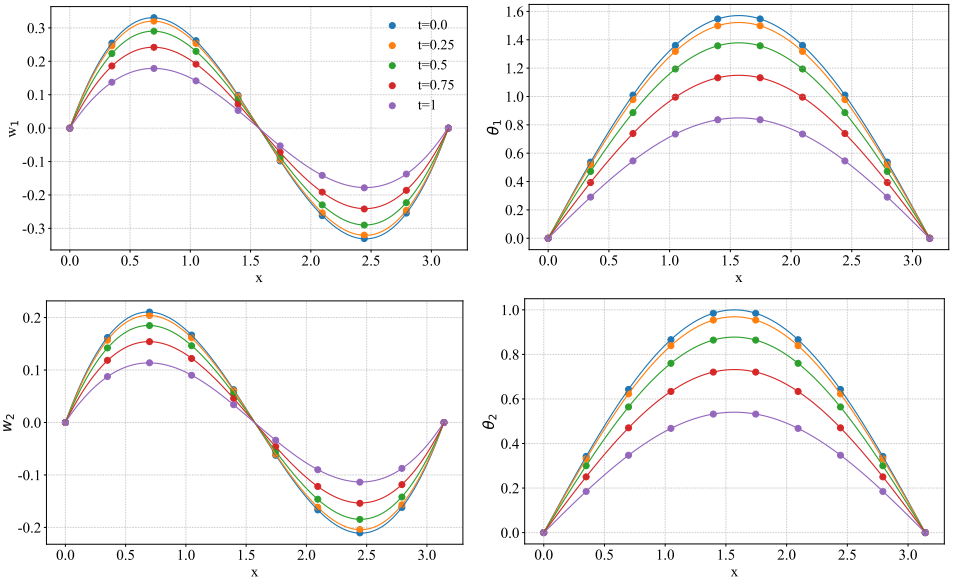


Figure 2.11: Timoshenko double beam. Scattered points represent the exact solution, and the continuous line refers to the predicted solution. **Top:** First beam **Left** Displacement (w_1); **Right** Rotation (θ_1). **Bottom:** Second beam **Left** Displacement (w_2); **Right** Rotation (θ_2)

.

$$w_1(x,0) = \sin(x), \quad w_{1_t}(x,0) = 0$$
$$w_2(x,0) = \frac{\pi}{2}\sin(x), \quad w_{2_t}(x,0) = 0$$

All four ends of the beams are assumed to be simply supported, expressed as

$$w_1(0,t) = w_1(\pi,t) = w_{1_{xx}}(0,t) = w_{1_{xx}}(\pi,t) = 0$$
$$w_2(0,t) = w_2(\pi,t) = w_{2_{xx}}(0,t) = w_{2_{xx}}(\pi,t) = 0$$

The external acting force is

$$f_1(x,t) = \left(1 - \frac{\pi}{2}\right)\sin(x)\cos(t)$$
$$f_2(x,t) = \left(\frac{\pi}{2} - 1\right)\sin(x)\cos(t)$$

For the considered problem, the analytical solution is given by

$$w_1(x,t) = \sin(x)\cos(t), \quad w_2(x,t) = \frac{\pi}{2}\sin(x)\cos(t)$$

In addition to computing the beam displacements, derived quantities such as velocity, acceleration, and bending moment are also computed for this problem. These derived quantities also help in the prognosis and diagnostics of the system. For instance, the bending moment estimates the bending effect when an external force is applied to a structural element. Estimating the bending moment can be used to quantify the bending upon the action of applied forces. The beam is the most common structural member vulnerable to bending moments because it can bend at any point along its length when subjected to an external force.

Table 2.2: Euler-Bernoulli double-beam: $\mathcal{R}$ at $t = 1$

	First beam	Second beam
Displacement (%)	1.9348×10^{-5}	4.3253×10^{-5}
Bending Moment (%)	9.6112×10^{-4}	6.5506×10^{-4}
Velocity (%)	1.9043×10^{-3}	2.0161×10^{-3}
Acceleration (%)	1.9011×10^{-2}	1.4442×10^{-2}

For simulating Euler-Bernoulli double beams, the same neural network architecture as for the single Euler-Bernoulli beam is considered. The only change is in the residual parameter, which is 1 for this case. The results are illustrated in Fig. 2.8, Fig. 2.9 and Table 2.2. The absolute difference between the PINN predicted solution and the exact solution for the first beam is approximately 10^{-4}, and for the second beam, it is approximately 10^{-3}, as shown in Fig. 2.8. The bending moment, velocity and acceleration are computed using the autodifferentiation and backpropagation features of the neural network. Table 2.2 describes the efficiency in the computation of these quantities at $t = 1$ for both beams. The relative percent error in computing the transverse displacement

of the beams on the order of 10^{-5}, and for acceleration, this error is on the order of 10^{-2}, which is very low and shows the potential of physics-informed learning. Fig. 2.9 illustrates the computed velocity, bending moment, and acceleration of both beams.

2.4.4. TIMOSHENKO DOUBLE-BEAM FORWARD PROBLEM

The double-beam system modeled by Euler-Bernoulli theory can also be modelled using Timoshenko theory under the same assumptions as described for the single Timoshenko equations [40]. In addition to providing the transverse displacement of the beams, Timoshenko theory also provides the cross-sectional rotation of both beams through the system of PDEs [40] given by

$$
\begin{aligned}
kA_1G(\bar{\theta}_{1_{\bar{x}}} - \bar{w}_{1_{\bar{x}\bar{x}}}) + \rho A_1 \bar{w}_{1_{\bar{t}\bar{t}}} + K(\bar{w}_1 - \bar{w}_2) &= \bar{f}_1(\bar{x}, \bar{t}) \\
EI_2\bar{\theta}_{2_{\bar{x}\bar{x}}} + GA_2k(\bar{w}_{2_{\bar{x}}} - \bar{\theta}_2) - \rho I_2\bar{\theta}_{2_{\bar{t}\bar{t}}} &= 0 \\
kA_2G(\bar{\theta}_{2_{\bar{x}}} - \bar{w}_{2_{\bar{x}\bar{x}}}) + \rho A_2 \bar{w}_{2_{\bar{t}\bar{t}}} + K(\bar{w}_2 - \bar{w}_1) &= \bar{f}_2(\bar{x}, \bar{t}) \\
EI_1\bar{\theta}_{1_{\bar{x}\bar{x}}} + GA_1k(\bar{w}_{1_{\bar{x}}} - \bar{\theta}_1) - \rho I_1\bar{\theta}_{1_{\bar{t}\bar{t}}} &= 0
\end{aligned}
\tag{2.16}
$$

where $\bar{w}_i(\bar{x}, \bar{t})$ and $\bar{\theta}_i(\bar{x}, \bar{t})$, $i = 1, 2$ denote the transverse displacement and cross-sectional rotation of the beams respectively. K is the stiffness modulus of the Winkler elastic layer. G is the shear modulus and k is the Timoshenko shear coefficient. The rest of the parameters have the usual meanings as described earlier. For simplicity, we consider $A_1 = A_2$, and $I_1 = I_2$ and nondimensionalize (2.16). With some additional assumptions, the non-dimensional equation has the same form as (2.16) with unit coefficients. For the numerical experiment the initial state of the double beam system is taken to be

$$
\begin{aligned}
\theta_1(x, 0) &= \left(\frac{\pi}{2}\cos(x) + \left(x - \frac{\pi}{2}\right)\right), \quad \theta_{1_t}(x, 0) = 0 \\
w_1(x, 0) &= \frac{\pi}{2}\sin(x), \quad w_{1_t}(x, 0) = 0 \\
\theta_2(x, 0) &= \frac{2}{\pi}\left(\frac{\pi}{2}\cos(x) + \left(x - \frac{\pi}{2}\right)\right), \quad \theta_{2_t}(x, 0) = 0 \\
w_2(x, 0) &= \sin(x), \quad w_{2_t}(x, 0) = 0
\end{aligned}
$$

Simply supported boundary conditions are provided to make the problem wellposed

$$
\begin{aligned}
\theta_1(0, t) = \theta_1(\pi, t) = w_1(0, t) = w_1(\pi, t) = 0 \\
\theta_2(0, t) = \theta_2(\pi, t) = w_2(0, t) = w_2(\pi, t) = 0
\end{aligned}
$$

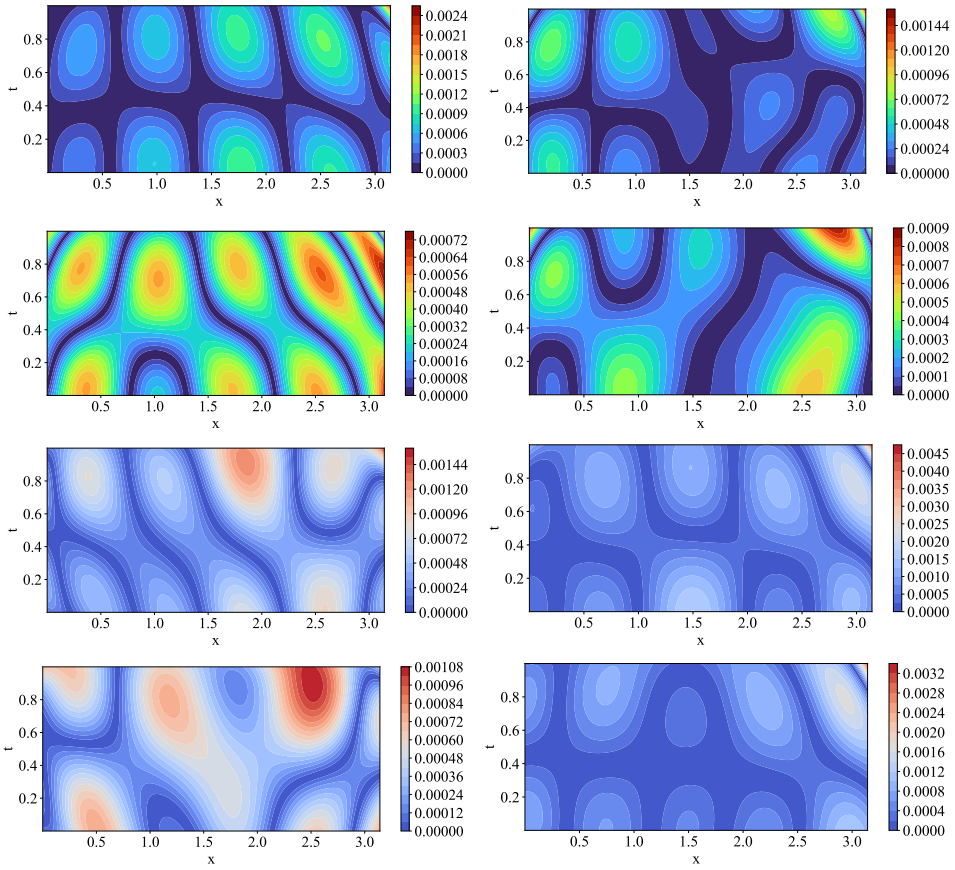


Figure 2.12: Timoshenko double beam absolute errors in prediction; θ_1 and w_1 are the rotation and displacement of the first beam, θ_2 and w_2 are the rotation and displacement of the second beam **First row** 16000 training points **(Left)** $|\theta_1 - \theta_1^*|$; **(Right)** $|w_1 - w_1^*|$; **(Second row, Left)** $|\theta_2 - \theta_2^*|$; **(Right)** $|w_2 - w_2^*|$. **Third row** 1600 training points **(Left)** $|\theta_1 - \theta_1^*|$; **(Right)** $|w_1 - w_1^*|$; **(Fourth row, Left)** $|\theta_2 - \theta_2^*|$; **(Right)** $|w_2 - w_2^*|$.

Here, $f_1(x,t)$, $f_2(x,t)$ and the analytic solutions are as follows

$$f_1(x,t) = \cos(t)(1-\sin(x))$$
$$f_2(x,t) = \frac{2}{\pi}\cos(t) - \frac{\pi}{2}\sin(x)\cos(t)$$

$$\theta_1(x,t) = \left(\frac{\pi}{2}\cos(x) + \left(x - \frac{\pi}{2}\right)\right)\cos(t)$$
$$\theta_2(x,t) = \frac{2}{\pi}\left(\frac{\pi}{2}\cos(x) + \left(x - \frac{\pi}{2}\right)\right)\cos(t)$$
$$w_1(x,t) = \frac{\pi}{2}\sin(x)\cos(t), \quad w_2(x,t) = \sin(x)\cos(t)$$

Table 2.3: Timoshenko double-beam: hyperparameters

No. of points	N_i	N_b	N_{int}	Layers	Neurons	Epochs
16000	2000	2000	10000	4	20	15K
1600	200	200	1000	4	20	15K

Table 2.4: Timoshenko double-beam: $\mathcal{R}$ at $t = 1$

	16000 points	1600 points
θ_1 (%)	1.6038×10^{-3}	2.6211×10^{-3}
w_1(%)	3.9302×10^{-5}	2.503×10^{-4}
θ_2 (%)	1.0826×10^{-3}	4.9405×10^{-3}
w_2(%)	7.8614×10^{-5}	3.4904×10^{-4}

Two experiments are performed, varying the number of training points, as shown in Table 2.3. Table 2.4 shows the relative percent error in approximating the transverse displacement and cross-sectional rotations for both beams. For cross-sectional rotations θ_1 and θ_2, the magnitude of the percent error remains the same even for fewer training points.

Using a large number of training points can increase the training time and may not be feasible for problems with many parameters. In these cases, using fewer training points can lead to less accurate solutions, but they can be obtained relatively faster. This approach allows engineers to make informed decisions about the parameters, and once optimal parameters have been identified, forward solutions can be recalculated with higher accuracy by using more training points. This is referred to as training with fewer points for the forward problem.

The absolute difference between the predicted and exact solutions of θ_1, w_1, θ_2 and w_2, even for 1600 training points is very small as shown in Fig. 2.11 and Fig. 2.12. Fig. 2.11 presents the PINNs prediction for a double Timoshenko beam. The scattered points refer to the exact solution, and the continuous line represents the predicted solution. The force is applied uniformly in both beams; however, the deflection and rotation of the first beam are greater than those of the second beam.

The results in Fig. 2.12 indicate that, for the second beam, a larger number of training points (16000) results in a more accurate prediction of deflection and rotation than a smaller number of training points (1600). Conversely, for the first beam, a smaller number of training points (1600) results in a more accurate prediction of the quantity of interest than a larger number of training points (16000). In any case, the difference in absolute error is relatively small, demonstrating that even with fewer training points, PINNs can still produce accurate predictions.

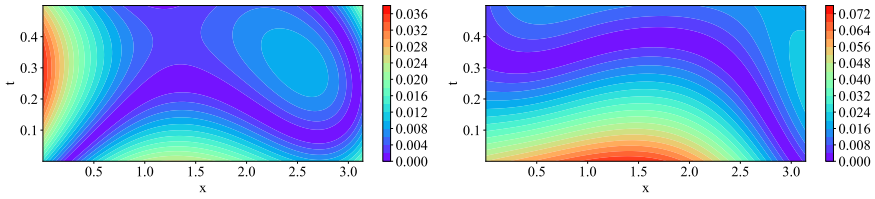


Figure 2.13: Timoshenko double-beam inverse problem: absolute error in the prediction of force when the additional data of rotation and deflections provided at five locations has **left:** no noise **right:** 20 percent Gaussian noise.

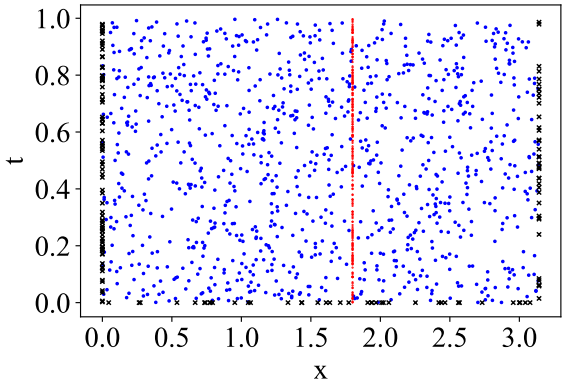


Figure 2.14: Data to learn material properties for the Timoshenko double beam: **Blue dots** Collocation points. **Red dots** Additional data points of displacement and rotation for the double beam at one location. **Black dots** Initial and boundary points.

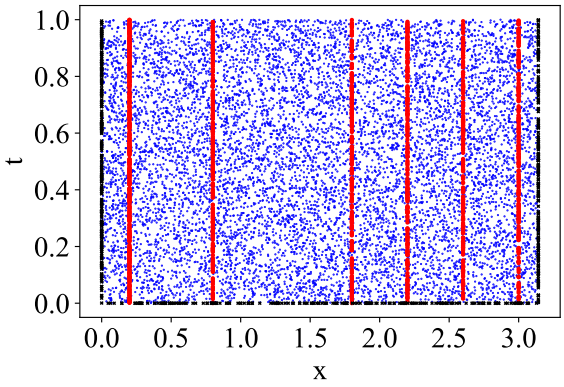


Figure 2.15: Data to learn force for the Timoshenko double beam: **Blue dots** Collocation points. **Red dots** Additional data points of displacement and rotation for the double beam at six different locations. **Black dots** Initial and boundary points.

2.4.5. TIMOSHENKO DOUBLE-BEAM INVERSE PROBLEM

The applied force on structural systems is critical for structural design and condition assessment. In design, control, and diagnosis, accurate estimation of dynamic forces acting on a structure is essential. These details can be used to evaluate the structural condition. For example, understanding the impact of heavy vehicles on bridge structures can aid in detecting early damage to them. Indirect force determination is of special interest when the applied forces cannot be measured directly, while the responses can be measured easily.

For the inverse problem, three distinct experiments are performed on (2.16). First, the unknown parameter is learned from the Timoshenko double-beam system. We consider the unknown parameter to be ρA_1 from (2.16). For the value of $\rho A_1 = 1$, the data for transverse displacement and cross-sectional rotation are provided at some points in the computational domain. Second, the unknown applied function on the first beam is learned by providing noise-free simulated displacement and cross-sectional rotation data. For this case, all other parameters, initial and boundary conditions are considered to be known, and only the function $f_1(x,t)$ is unknown. Third, the same force function is predicted by providing noisy displacement and cross-sectional rotation data. The data generated for learning the function in the second case are corrupted with noise to be used in the third case. The exact solution for the function to be learned in the second and third cases is $\cos(t)(1-\sin(x))$.

The inverse problem in engineering refers to the process of estimating unknown parameters or functions from a set of measured data. In

PINNs, the inverse problem is usually solved by training a neural network to fit the measured data and the known physical laws. However, the measured data can be affected by various sources of noise, which can make estimation of the quantity of interest more challenging. The noise can make the measured data unreliable, and the neural network may not be able to accurately estimate the unknown parameters or functions. In such a scenario, the optimizer of the neural network does not necessarily converge to local minima.

The same neural network architecture is used as in the forward double-beam Timoshenko problem, with residual parameter 1 to regularize the physical equation in the loss function. Here, 2500 epochs are performed using the L-BFGS optimizer to train the neural network. For learning the parameter, 5000 data points are provided at $x = 1.8$, as shown in Fig. 2.14. The exact value of the unknown parameter is $\rho A_1 = 1$ in (2.16), and the predicted value of the parameter using the PINN framework is 1.0208, which is close to the desired value. Even for a system of four PDEs, by only providing data at one particular beam location, the unknown parameter is learned successfully using PINNs. This shows that PINNs can handle large complex systems of PDEs efficiently.

Table 2.5: Timoshenko double-beam inverse problem: noise vs. $\mathcal{R}$

Noise percent (%)	Relative error percent (%)
0	4.3271×10^{-2}
10	4.8688×10^{-2}
20	1.1123×10^{-1}

The function $f_1(x, t)$, the applied force on the first beam is predicted in the second experiment. As illustrated in Fig. 2.15, the data for transverse displacement and cross-sectional rotation are provided at 6 different locations with 5000 data points at each location.

For the third experiment, the data provided for learning the unknown function $f_1(x, t)$ are provided with 10% and 20% Gaussian noise and the corresponding performance in learning the function is shown in Table 2.5. Even with 10% and 20% noise, the relative error percent between analytic and predicted force is lower, as seen in TABLE 2.5. Fig. 2.13 shows the force prediction along the beam when rotation and deflection observations are available at five points. The results demonstrate that the PINN is more precise in its predictions when the data are free from noise compared to when they are noisy. Despite the presence of noise in the data, the absolute error remains within the magnitude of 10^{-2}, which is comparable to the error observed when data are not noisy. To be more precise, Fig. 2.13 shows the absolute difference error of the

PINN predicted and exact force at $t = 0.5$ with 0 percent and 20 percent noise. Even with 20 percent noise, the unknown force is learned with less than 1% error over the entire space-time domain, demonstrating that PINN is a very accurate and robust approach.

The minimum number of data points required to estimate the model parameters depends on several factors, such as the complexity of the physics, the number of physical parameters in the model, and the quality of the data. More data points and more complex physics require more neural network capacity, resulting in a larger neural network with more hyperparameters. In practice, more data points lead to overfitting. The minimum training data points required for a PINN framework are determined empirically by gradually increasing the number of training points until the performance of the model is satisfactory.

Finally, a sensitivity analysis is carried out to examine the influence of input variables, specifically the displacement and rotation, on the output variable, which is the force. The analysis involves adding 20% Gaussian noise to the displacement data while no noise is added to the rotation data. The resulting mean accuracy of the force is 0.14313413. In contrast, when 20% noise is introduced to the rotation data with the displacement data remaining unaltered, the mean accuracy of the force is 0.204627. The results of this analysis show that the force is more sensitive to rotations than the displacement data.

2.5. CONCLUSIONS

The design and maintenance of complex structural systems are challenging due to the multiscale interaction of their components. It is desirable to predict the behavior of these complex systems by solving the governing model of interest. Recently, PINNs have emerged as a viable method for simulating PDEs. In this work, we propose using the PINN algorithm with the nondimensionalization step aiding in the learning procedure for complex beam systems. The PINN framework successfully solves the forward and inverse problems for nondimensional single and double-beam systems. Based on the numerical experiments, the following conclusions are drawn.

First, the relative percent error in computing the beam displacement does not increase with increasing model complexity when solving the forward problem. In fact, for both Euler-Bernoulli and Timoshenko theory, the error decreases by an order of magnitude for double-beam systems compared to single-beam systems. In addition, the error in computing the bending rotation is comparable for single and double Timoshenko beam systems. This nonincrease in error as the model complexity increases suggests that the PINN framework is appropriate for simulating large-scale systems with multiple connected components.

Second, it is demonstrated that PINNs precisely discover the unknown

force function and model parameters through their inverse problem-solving capability. The proposed algorithm successfully learns the model parameter with less than 3% error for the single Timoshenko beam. In addition, for the double beam Timoshenko system, the unknown function is approximated on the whole space-time domain with less than 0.05% error, demonstrating the effectiveness of the algorithm for solving inverse problems.

Third, physical quantities such as velocity, acceleration, and bending moment characterize the behavior of the system. Even though the derived quantities are not directly trained in the neural network, they are approximated with less than $2e-2\%$ error for the Euler-Bernoulli double-beam system.

Fourth, the ability of the algorithm to use fewer training points in forward problems and to accommodate noisy data in inverse problems is exploited. The obtained results show that even with 1600 training points, the double Timoshenko beam displacement is predicted on the entire space-time domain with less than $5e-3\%$ error. In the case of the inverse problem, the force function is discovered with less than 0.2% error even when the data used in the learning procedure contains 20% Gaussian noise. These findings imply that the algorithm is accurate and robust under the tested noise levels.

To summarize, PINNs enable the simulation of complex structural systems with multiple interacting components efficiently, accurately, and robustly. In the future, this approach could be extended to estimate displacements for various input forces and mechanical vibration modes and incorporate robust methods to account for stochasticities. Additionally, future works on PINNs could be focused on reducing the computational cost and developing methodologies to augment their generalizability, thereby expanding the applicability of PINNs beyond the training domain.

REFERENCES

[1] K. A. Kobbacy, D. P. Murthy, *et al. Complex system maintenance handbook*. Springer, 2008. doi: 10.1007/978-1-84800-011-7.

[2] M. Chatzis and G. Deodatis. “Modeling of very large interacting multiple-beam systems with application to suspension bridge cables”. In: *Journal of Structural Engineering* 139.9 (2013), pp. 1541–1554. doi: 10.1061/(ASCE)ST.1943-541X.0000740.

[3] Z. Liu, Y. Song, Y. Han, H. Wang, J. Zhang, and Z. Han. “Advances of research on high-speed railway catenary”. In: *Journal of Modern Transportation* 26.1 (2018), pp. 1–23. doi: 10.1007/s40534-017-0148-4.

[4] C. Foias, O. Manley, R. Rosa, and R. Temam. *Navier-Stokes equations and turbulence*. Vol. 83. Cambridge University Press, 2001. doi: 10.1017/CBO9780511546754.

[5] D. Lucor, A. Agrawal, and A. Sergent. “Simple computational strategies for more effective physics-informed neural networks modeling of turbulent natural convection”. In: *Journal of Computational Physics* 456 (2022), p. 111022. doi: 10.1016/j.jcp.2022.111022.

[6] V. Stojanović and P. Kozić. *Vibrations and stability of complex beam systems*. Springer, 2015. doi: 10.1007/978-3-319-13767-4.

[7] J. Mercieca and V. Kadirkamanathan. “Estimation and identification of spatio-temporal models with applications in engineering, healthcare and social science”. In: *Annual Reviews in Control* 42 (2016), pp. 285–298. doi: 10.1016/j.arcontrol.2016.09.011.

[8] H.-X. Li and C. Qi. “Modeling of distributed parameter systems for applications—A synthesized review from time–space separation”. In: *Journal of Process Control* 20.8 (2010), pp. 891–901. doi: 10.1016/j.jprocont.2010.06.016.

[9] T. Kapoor, H. Wang, A. Núñez, and R. Dollevoet. “Predicting traction return current in electric railway systems through physics-informed neural networks”. In: *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2022, pp. 1460–1468. doi: 10.1109/SSCI51031.2022.10022290.

[10] A. Chandra, M. Curti, K. Tiels, E. A. Lomonova, and D. M. Tartakovsky. “Physics-informed neural networks for modelling anisotropic and bi-anisotropic electromagnetic constitutive laws through indirect data”. In: *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2022, pp. 1451–1459. doi: 10.1109/SSCI51031.2022.10022292.

[11] L. Yuan, Y.-Q. Ni, X.-Y. Deng, and S. Hao. “A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations”. In: *Journal of Computational Physics* 462 (2022), p. 111260. doi: 10.1016/j.jcp.2022.111260.

[12] A. Fallah and M. M. Aghdam. “Physics-informed neural network for bending and free vibration analysis of three-dimensional functionally graded porous beam resting on elastic foundation”. In: *Engineering with Computers* (2023), pp. 1–18. doi: 10.1007/s00366-023-01799-7.

[13] T. Kapoor, H. Wang, A. Nunez, and R. Dollevoet. “Physics-informed neural networks for solving forward and inverse problems in complex beam systems”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023). doi: 10.1109/TNNLS.2023.3310585.

[14] M. G. Kapteyn, J. V. Pretorius, and K. E. Willcox. “A probabilistic graphical model foundation for enabling predictive digital twins at scale”. In: *Nature Computational Science* 1.5 (2021), pp. 337–347. doi: 10.1038/s43588-021-00069-0.

[15] H. Wang, A. Núñez, Z. Liu, D. Zhang, and R. Dollevoet. “A Bayesian network approach for condition monitoring of high-speed railway catenaries”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.10 (2019), pp. 4037–4051. doi: 10.1109/TITS.2019.2934346.

[16] F.-G. Yuan, S. A. Zargar, Q. Chen, and S. Wang. “Machine learning for structural health monitoring: challenges and opportunities”. In: *Sensors and smart structures technologies for civil, mechanical, and aerospace systems 2020*. Vol. 11379. International Society for Optics and Photonics. 2020, p. 1137903. doi: 10.1117/12.2561610.

[17] J. Borggaard and J. Burns. “A PDE sensitivity equation method for optimal aerodynamic design”. In: *Journal of Computational Physics* 136.2 (1997), pp. 366–384. doi: 10.1006/jcph.1997.5743.

[18] Z. Lai, C. Mylonas, S. Nagarajaiah, and E. Chatzi. “Structural identification with physics-informed neural ordinary differential equations”. In: *Journal of Sound and Vibration* 508 (2021), p. 116196. doi: 10.1016/j.jsv.2021.116196.

[19] M. Raissi and G. E. Karniadakis. “Hidden physics models: Machine learning of nonlinear partial differential equations”. In: *Journal of Computational Physics* 357 (2018), pp. 125–141. doi: 10.1016/j.jcp.2017.11.039.

[20] N. Abhyankar, E. Hall, and S. Hanagud. “Chaotic vibrations of beams: numerical solution of partial differential equations”. In: (1993). doi: 10.1115/1.2900741.

[21] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440. doi: 10.1038/s42254-021-00314-5.

[22] Fang, Zhiwei. “A high-efficient hybrid physics-informed neural networks based on convolutional neural network”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021). doi: 10.1109/TNNLS.2021.3070878.

[23] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. “Scientific machine learning through physics–informed neural networks: where we are and what’s next”. In: *Journal of Scientific Computing* 92.3 (2022), p. 88. doi: 10.1007/s10915-022-01939-z.

[24] J. Blechschmidt and O. G. Ernst. “Three ways to solve partial differential equations with neural networks—A review”. In: *GAMM-Mitteilungen* 44.2 (2021), e202100006. doi: 10.1002/gamm.202100006.

[25] M. Raissi, P. Perdikaris, and G. E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational physics* 378 (2019), pp. 686–707. doi: 10.1016/j.jcp.2018.10.045.

[26] Meng, Yuxin and Rigall, Eric and Chen, Xueen and Gao, Feng and Dong, Junyu and Chen, Sheng. “Physics-Guided Generative Adversarial Networks for Sea Subsurface Temperature Prediction”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021). doi: 10.1109/TNNLS.2021.3123968.

[27] Xu, Wengang and Zhou, Zheng and Li, Tianfu and Sun, Chuang and Chen, Xuefeng and Yan, Ruqiang. “Physics-Constraint Variational Neural Network for Wear State Assessment of External Gear Pump”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022). doi: 10.1109/TNNLS.2022.3213009.

[28] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar. “Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data”. In: *IEEE Transactions on Knowledge and Data Engineering* 29.10 (2017), pp. 2318–2331. doi: 10.1109/TKDE.2017.2720168.

[29] M. H. Hassoun. *Fundamentals of artificial neural networks*. IEEE, 1996. doi: 10.1109/JPROC.1996.503146.

[30] Oszkinat, Clemens and Luczak, Susan E and Rosen, IG. “Uncertainty quantification in estimating blood alcohol concentration from transdermal alcohol level with physics-informed neural networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022). doi: 10.1109/TNNLS.2022.3140726.

[31] Hua, Jiaqi and Li, Yingguang and Liu, Changqing and Wan, Peng and Liu, Xu. “Physics-Informed Neural Networks With Weighted Losses by Uncertainty Evaluation for Accurate and Stable Prediction of Manufacturing Systems”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023). doi: 10.1109/TNNLS.2023.3247163.

[32] S. Wang, X. Yu, and P. Perdikaris. “When and why PINNs fail to train: A neural tangent kernel perspective”. In: *Journal of Computational Physics* 449 (2022), p. 110768. doi: 10.1016/j.jcp.2021.110768.

[33] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. “Characterizing possible failure modes in physics-informed neural networks”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[34] G. Kissas, Y. Yang, E. Hwuang, W. R. Witschey, J. A. Detre, and P. Perdikaris. “Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 358 (2020), p. 112623. doi: 10.1016/j.cma.2019.112623.

[35] Feng, Jiali and Liu, Zhijie and He, Xiuyu and Li, Qing and He, Wei. “Vibration Suppression of a High-Rise Building With Adaptive Iterative Learning Control”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–12. doi: 10.1109/TNNLS.2021.3120838.

[36] Joghataie, Abdolreza and Torghabehi, Omid Oliyan. “Simulating dynamic plastic continuous neural networks by finite elements”. In: *IEEE Transactions on Neural Networks and Learning Systems* 25.8 (2013), 1583–1587. doi: 10.1109/TNNLS.2013.2294315.

[37] Liu, Yu and Wu, Xiaoqi and Yao, Xiangqian and Zhao, Jingyi. “Backstepping Technology-Based Adaptive Boundary ILC for an Input-Output-Constrained Flexible Beam”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–9. doi: 10.1109/TNNLS.2022.3157950.

[38] He, Wei and Meng, Tingting and Huang, Deqing and Li, Xuefang. “Adaptive Boundary Iterative Learning Control for an Euler–Bernoulli Beam System With Input Constraint”. In: *IEEE Transactions on Neural Networks and Learning Systems* 29.5 (2018), 1539–1549. doi: 10.1109/TNNLS.2017.2673865.

[39] Z. Oniszczuk. “Free transverse vibrations of elastically connected simply supported double-beam complex system”. In: *Journal of Sound and Vibration* 232.2 (2000), pp. 387–403. doi: 10.1006/jsvi.1999.2744.

[40] V. Stojanović and P. Kozić. “Forced transverse vibration of Rayleigh and Timoshenko double-beam system with effect of compressive axial load”. In: *International Journal of Mechanical Sciences* 60.1 (2012), pp. 59–71. doi: 10.1016/j.ijmecsci.2012.04.009.

[41] Z. Oniszczuk. “Forced transverse vibrations of an elastically connected complex simply supported double-beam system”. In: *Journal of Sound and Vibration* 264.2 (2003), pp. 273–286. doi: 10.1016/S0022-460X(02)01166-5.

[42] M. Abu-Hilal. “Dynamic response of a double Euler–Bernoulli beam due to a moving constant load”. In: *Journal of Sound and Vibration* 297.3-5 (2006), pp. 477–491. doi: 10.1016/j.jsv.2006.03.050.

[43] X. Zhao, B. Chen, Y. Li, W. Zhu, F. Nkiegaing, and Y. Shao. “Forced vibration analysis of Timoshenko double-beam system under compressive axial load by means of Green’s functions”. In: *Journal of Sound and Vibration* 464 (2020), p. 115001. doi: 10.1016/j.jsv.2019.115001.

[44] J. Li and H. Hua. “Spectral finite element analysis of elastically connected double-beam systems”. In: *Finite Elements in Analysis and Design* 43.15 (2007), pp. 1155–1168. doi: 10.1016/j.finel.2007.08.007.

[45] A. Palmeri and S. Adhikari. “A Galerkin-type state-space approach for transverse vibrations of slender double-beam systems with viscoelastic inner layer”. In: *Journal of Sound and Vibration* 330.26 (2011), pp. 6372–6386. doi: 10.1016/j.jsv.2011.07.037.

[46] Z. Ying and Y. Ni. “A response-adjustable sandwich beam with harmonic distribution parameters under stochastic excitations”. In: *International Journal of Structural Stability and Dynamics* 17.07 (2017), p. 1750075. doi: 10.1142/S0219455417500754.

[47] Z. Ying, Y. Ni, and R. Huan. “Vibration response characteristics of quasi-periodic sandwich beam with magnetorheological visco-elastomer core under random support excitations”. In: *Journal of Vibration and Acoustics* 140.5 (2018). doi: 10.1115/1.4039726.

[48] T. Murmu and S. Adhikari. “Nonlocal transverse vibration of double-nanobeam-systems”. In: *Journal of Applied Physics* 108.8 (2010), p. 083514. doi: 10.1063/1.3496627.

[49] B. Chen, B. Lin, X. Zhao, W. Zhu, Y. Yang, and Y. Li. “Closed-form solutions for forced vibrations of a cracked double-beam system interconnected by a viscoelastic layer resting on Winkler–Pasternak elastic foundation”. In: *Thin-Walled Structures* 163 (2021), p. 107688. doi: 10.1016/j.tws.2021.107688.

[50] S. Liu and B. Yang. “A closed-form analytical solution method for vibration analysis of elastically connected double-beam systems”. In: *Composite Structures* 212 (2019), pp. 598–608. doi: 10.1016/j.compstruct.2019.01.038.

[51] X. Zhao and P. Chang. “Free and forced vibration of double beam with arbitrary end conditions connected with a viscoelastic layer and discrete points”. In: *International Journal of Mechanical Sciences* 209 (2021), p. 106707. doi: 10.1016/j.ijmecsci.2021.106707.

[52] Y. Li, F. Xiong, L. Xie, and L. Sun. “State-space approach for transverse vibration of double-beam systems”. In: *International Journal of Mechanical Sciences* 189 (2021), p. 105974. doi: 10.1016/j.ijmecsci.2020.105974.

[53] O. Z. S. Ong, M. H. Ghayesh, D. Losic, and M. Amabili. “Coupled dynamics of double beams reinforced with bidirectional functionally graded carbon nanotubes”. In: *Engineering Analysis with Boundary Elements* 143 (2022), pp. 263–282. doi: 10.1016/j.enganabound.2022.06.023.

[54] Radev, Stefan T. and Mertens, Ulf K. and Voss, Andreas and Ardizzone, Lynton and Köthe, Ullrich. “BayesFlow: Learning Complex Stochastic Models With Invertible Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.4 (2022), 1452–1466. doi: 10.1109/TNNLS.2020.3042395.

[55] Ning, Hanwen and Qing, Guangyan and Jing, Xingjian. “Identification of Nonlinear Spatiotemporal Dynamical Systems With Nonuniform Observations Using Reproducing-Kernel-Based Integral Least Square Regulation”. In: *IEEE Transactions on Neural Networks and Learning Systems* 27.11 (2016), 2399–2412. doi: 10.1109/TNNLS.2015.2473686.

[56] Jin, Pengzhan and Zhang, Zhen and Kevrekidis, Ioannis G. and Karniadakis, George Em. “Learning Poisson Systems and Trajectories of Autonomous Systems via Poisson Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022), pp. 1–13. doi: 10.1109/TNNLS.2022.3148734.

[57] S. Mishra and R. Molinaro. “Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs”. In: *IMA Journal of Numerical Analysis* 42.2 (2022), pp. 981–1022. doi: 10.1093/imanum/drab032.

[58] M. Bazmara, M. Silani, M. Mianroodi, *et al.* “Physics-informed neural networks for nonlinear bending of 3D functionally graded beam”. In: *Structures*. Vol. 49. Elsevier. 2023, pp. 152–162. doi: 10.1016/j.istruc.2023.01.115.

[59] U. Goerguelue. “Beam theories the difference between Euler-Bernoulli and Timoschenko”. In: *Lecture Handouts* (2009).

[60] P. Akella, E. G. Hemingway, and O. M. O’Reilly. “A Visualization Tool for the Vibration of Euler-Bernoulli and Timoshenko Beams”. In: ().

[61] B. Semper. “Semi-discrete and fully discrete Galerkin methods for the vibrating Timoshenko beam”. In: *Computer Methods in Applied Mechanics and Engineering* 117.3-4 (1994), pp. 353–360. doi: 10.1016/0045-7825(94)90122-8.

[62] Y. Yu, H. Yao, and Y. Liu. “Structural dynamics simulation using a novel physics-guided machine learning method”. In: *Engineering Applications of Artificial Intelligence* 96 (2020), p. 103947. doi: 10.1016/j.engappai.2020.103947.

[63] Z. Zhang and C. Sun. “Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating”. In: *Structural Health Monitoring* 20.4 (2021), pp. 1675–1688. doi: 10.1177/1475921720927488.

[64] A. Karpatne, W. Watkins, J. Read, and V. Kumar. “Physics-guided neural networks (PGNN): An application in lake temperature modeling”. In: *arXiv preprint arXiv:1710.11431* 2 (2017). doi: 10.48550/arXiv.1710.11431.

[65] A. Khandelwal, S. Xu, X. Li, X. Jia, M. Stienbach, C. Duffy, J. Nieber, and V. Kumar. “Physics guided machine learning methods for hydrology”. In: *arXiv preprint arXiv:2012.02854* (2020). doi: 10.48550/arXiv.2012.02854.

[66] J. Yu, L. Lu, X. Meng, and G. E. Karniadakis. “Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems”. In: *Computer Methods in Applied Mechanics and Engineering* 393 (2022), p. 114823. doi: 10.1016/j.cma.2022.114823.

[67] Y. Miao and H. Li. “GPINN: Physics-informed Neural Network with Graph Embedding”. In: *arXiv preprint arXiv:2306.09792* (2023). doi: 10.48550/arXiv.2306.09792.

[68] J. Bettencourt, M. J. Johnson, and D. Duvenaud. “Taylor-mode automatic differentiation for higher-order derivatives in JAX”. In: *Program Transformations for ML Workshop at NeurIPS 2019*. 2019.

3

CAUSAL PINNS FOR BEAM SIMULATIONS

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

Apart from minor updates, this chapter has been published as: Kapoor, T., Wang, H., Núñez, A., & Dollevoet, R. (2024). Transfer learning for improved generalizability in causal physics-informed neural networks for beam simulations. Engineering Applications of Artificial Intelligence, 133, 108085.

3.1. INTRODUCTION

Beams on elastic foundations (shown in Fig. 3.1) are a fundamental and indispensable structural component in civil engineering, providing critical support and stability to different and diverse structures [1–5]. Due to their characteristic to distribute loads, mitigate deformations, and enhance structural stability, these beams are extensively utilized in various structures, such as railway tracks [1], pile foundations embedded in soils [6], and longitudinal fibers in a composite elastomer [3], among others. Understanding their dynamics is essential for ensuring the structural integrity of these systems, developing effective maintenance strategies, optimizing machine performance, refining design methodologies, and enabling precise control mechanisms. These issues highlight the need for advanced methodologies to simulate and predict the underlying dynamics of beams on elastic foundations, facilitating safer, more efficient, and reliable structures and systems.

However, accurately predicting the dynamics of beams on elastic foundations through experiments and measurements could be infeasible [7]. Conducting many experiments with varying materials, conditions, and prototypes becomes impractical and prohibitively costly. In practice, finite element-based software provides a viable alternative for simulating such scenarios [8]. However, these software solutions are restricted in generalization. For instance, even a slight change in the problem domain requires performing the entire new simulation from scratch, including mesh creation and adjustments [9]. This non-generalization becomes particularly problematic when different aspects of the system need to be investigated separately or when multiple design iterations are required. The number of simulations necessary for tackling a design problem can quickly escalate into thousands, making the task laborious and time-consuming.

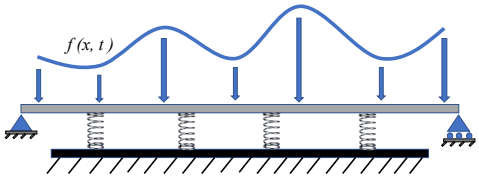


Figure 3.1: Simply supported beam on an elastic foundation under varying transverse force

Recently, deep learning and neural networks, in particular, have been used extensively as surrogates to model the underlying physical phenomenon [10, 11] including applications to shape optimization, resulting in cost-efficient shapes [12–14]. However, even state-of-the-art supervised machine-learning approaches encounter similar challenges as traditional experimental methods, requiring substantial

input-output data at various fidelities to learn the underlying dynamics effectively. This large data requirement poses a significant hurdle, as obtaining such vast data can be arduous and resource-intensive [9].

One potential approach to mitigate the need for an enormous amount of data is to incorporate the underlying physics into the learning procedure, thereby guiding the neural network based on physics principles as presented by [15–19], among others. One popular class of methods that adopts this approach is physics-informed neural networks (PINNs) proposed by [15]. PINNs are a form of semi-supervised learning where the boundary and initial conditions serve as input-output pairs while the solution is regularized by the governing partial differential equations (PDEs). However, several challenges for PINNs have emerged, including spectral bias [20], shock learning [21], generalization with even slight changes in physical parameters and computational domain [22–24], and difficulties dealing with large coefficients [25–27].

Another such open problem for vanilla PINNs is handling extensive space-time domains, as discussed in [28–31], among others. This challenge can be attributed to the training process, as vanilla PINNs tend to prioritize training at a higher time level due to implicit gradient bias [20], leading to violation in temporal causality and inaccurate solutions, particularly for problems highly dependent on initial conditions.

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

Our work proposes to train PINN while respecting causality [20] in the context of structural engineering, referred to as causal PINN hereafter. In particular, our principal aim is to resolve the training challenge and achieve precise predictions of beam dynamics in a large space-time domain. This challenge is overcome by proposing a modification in the training approach of PINNs, enforcing training at lower time levels before progressing to subsequent ones. Consequently, a weighted loss function is utilized, incorporating a causality parameter to preserve the physical causality inherent in beam dynamics. The causal PINN approach, validated through numerical experiments, demonstrates enhanced accuracy in prediction.

However, as we present in this work, even after employing causal PINN, the models lack generalizability to different initial conditions and computational domains, requiring each new problem to be solved from scratch. This limitation reverts the problem to the need for extensive

simulations for each problem. To mitigate this issue, we propose employing transfer learning (TL) [32] in conjunction with causal PINN. The idea of transfer learning is to utilize the knowledge acquired from solving one problem in the form of trained model parameters to be utilized in a similar or related problem, accelerating the training process.

We examine the application of PINNs on well-known Euler-Bernoulli and Timoshenko beam models on elastic foundations, specifically the Winkler foundation [1, 33]. Through the numerical examples in this chapter, we show that vanilla PINNs face challenges in approximating solutions for PDEs, for which even analytical solutions are available. In practice, only a handful of PDEs have analytical solutions that serve as prototypes for proof of concept to validate a proposed method. The inefficiency of vanilla PINNs in resolving solutions for such PDEs signifies its limited applicability in the real world, which we tackle in this chapter by enforcing a causal training framework.

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

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

The rest of the chapter is structured as follows: Section 3.2 presents related works to this chapter. Section 3.3 provides a detailed discussion of vanilla and causal PINN. Section 3.4 introduces the proposed framework of fusing transfer learning with causal PINN to train different models. Section 3.5 presents the numerical experiments results, showcasing the effectiveness of our methodology in addressing challenging beam problems where the vanilla and advanced PINN-based methods fail. Finally, the main findings are summarized, and conclusions drawn from this study are presented in Section 3.6.

3.2. RELATED WORKS

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

Applying transfer learning within PINNs has garnered significant attention [34]. Notably, [35] predicted laser deposition temperature fields accurately without labeled data, using physical losses and transfer learning. In another work, [36] utilized transfer learning for accurate temperature field inversion with limited observations, employing a PINN and optimal position selection. [37] developed a multi-objective loss function and transfer learning for accurate elastoplastic solid mechanics solutions through PINN. In a different study, [38] proposed a transfer learning-based PINN framework for efficient stress-strain constitutive modeling. While our research aligns with the fundamental principle of leveraging transfer learning, a distinguishing aspect lies in our consideration of causality during the training of the models.

In the literature, research has been conducted to enforce causality in the PINN framework without incorporating transfer learning [39]. In another work, [40] proposed a causal framework incorporating transfer learning to simulate time-dependent PDEs. Although our work shares similar ideas of incorporating causality and utilizing transfer learning within the PINN framework, we employ transfer learning to train distinct models under diverse conditions. Conversely, [40] employs transfer learning within a particular problem by segmenting the domain into multiple subdomains and leveraging insights from one subdomain to another, employing the concepts of domain decomposition and PINNs [30].

Recently, beam simulations have concentrated on physics-informed methodologies, largely omitting the considerations of causality and transfer learning. Noteworthy works include [41], which utilized PINNs for estimating nonlinear bending behavior within a confined domain. Similarly, [42, 43] delved into applying PINNs for the system of beam models and moving load problems, albeit within the limited domain confines. [44] introduced a spatio-temporal PINN tailored for analyzing the dynamics of cantilever beams. In [45], a self-adaptive PINN framework capable of accommodating varying load conditions is presented. Additionally, [46] sought to enhance predictions by incorporating supplementary data, all still constrained within the confined domain bounded by the capabilities of PINNs. This work aims to enhance the potential of physics-informed methodologies for simulating beam dynamics.

3.3. VANILLA AND CAUSAL PINN

This section is structured into two subsections. First, we provide an overview of the architecture of the vanilla PINN [15]. Second, a modification in the PINN loss function leading to the incorporation of causality in the PINN loss function, as proposed by [20], is presented.

3.3.1. VANILLA PINN

Recently, PINNs have been widely used for solving PDEs across diverse domains, including but not limited to works by [47–49]. PINNs are based on deep neural network (DNN) architecture, and the idea of PINN is to incorporate physical knowledge in the loss function of DNN. The loss function consists of two terms - a data term and a physics term. The data term ensures that the neural network fits the provided data points, while the physics term enforces the PDE constraints. Here, the data term refers to the value of the quantity of interest at initial and boundary points. Minimizing the data term amounts to measuring the discrepancy between the predicted solution of the PINN and the measured data points. The physics term incorporates the PDE constraints into the loss function, evaluating the differential operator of the PDE using automatic differentiation [50]. The resulting equation is then included as a penalty term in the loss function. To elucidate these terms, we consider an abstract PDE as,

$$D(u(x,t,k)) = f(x,t), \quad (x,t) \in \mathcal{D} \times \mathcal{T} \tag{3.1}$$

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

$$\begin{aligned} u(x,0) &= g(x), \quad (x,0) \in \mathcal{D} \times \Gamma \\ u(x_b,t) &= \bar{g}(x_b,t), \quad (x_b,t) \in \Omega \times \mathcal{T} \end{aligned} \tag{3.2}$$

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

$$L(\mu) = \lambda_1 L_{\text{PDE}}(\mu) + \lambda_2 L_{\text{IC}}(\mu) + \lambda_3 L_{\text{BC}}(\mu) \tag{3.3}$$

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

$$L_{\text{PDE}}(\mu) = \frac{1}{N_{\text{int}}} \sum_{n=1}^{N_{\text{int}}} ||D(u^*(x^{(n)}, t^{(n)}, k)) - f(x^{(n)}, t^{(n)})||^p \tag{3.4}$$

The loss terms for initial and boundary conditions in (3.3) are defined as follows,

$$L_{\text{IC}}(\mu) = \frac{1}{N_{\text{i}}} \sum_{n=1}^{N_{\text{i}}} ||u^*(x^{(n)}, 0) - g(x^{(n)})||^p$$
$$L_{\text{BC}}(\mu) = \frac{1}{N_{\text{b}}} \sum_{n=1}^{N_{\text{b}}} ||u^*(x_{\text{b}}^{(n)}, t^{(n)}) - \bar{g}(x_{\text{b}}^{(n)}, t^{(n)})||^p \tag{3.5}$$

here, N is the total number of training points, which is the sum of interior training points (N_{int}), initial training points (N_{i}), and boundary training points N_{b}. The approximation of u by the neural network is denoted by u^*. Training with L^2-norm amounts to $p = 2$. The primary objective is minimizing (3.3) and obtaining optimal parameters (μ). These optimized parameters are then utilized for predicting the PDE solution $u(x, t), \forall (x, t) \in \mathcal{D} \times \mathcal{T}$.

3.3.2. CAUSAL PINN

This subsection presents causal PINN, modifying the PINN loss function [20]. The notion of causal PINNs is inspired by traditional numerical methods for solving differential equations that prioritize resolving the solution at lower times before approximating the solution at higher times. The modification in the loss function pertains to the PDE term $L_{\text{PDE}}(\mu)$, while the initial $L_{\text{IC}}(\mu)$ and boundary $L_{\text{BC}}(\mu)$ loss terms remain unchanged. The causal PDE loss term $L_{\text{PDE}}(\mu)$ is defined as

$$L_{\text{PDE}}(\mu) = \sum_{i=1}^{N_{\text{t}}} w_i L_{\text{PDE}}(t_i, \mu)$$
$$w_1 = 1, \quad w_i = e^{-\epsilon \sum_{k=1}^{i-1} L_{\text{PDE}}(t_k, \mu)}, \quad i = 2, 3, \ldots N_{\text{t}} \tag{3.6}$$

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

$$\frac{1}{N_t} \left[w_1 L_{\text{PDE}}(t_1, \mu) + \sum_{i=2}^{N_t} e^{-\epsilon \sum_{k=1}^{i-1} L_{\text{PDE}}(t_k, \mu)} L_{\text{PDE}}(t_i, \mu) \right] \tag{3.7}$$

From the above loss function, it is evident that for $L_{\text{PDE}}(\mu)$ to be minimized, the weights $w_1, \ldots, w_{N_t}$ should be large. However, the

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

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

3.4. TRANSFER LEARNING FOR CAUSAL PINN

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

The proposed approach incorporates transfer learning for different scenarios for the same physical beam equation. Fig. 3.2 visually demonstrates the steps: initially, the parent model is trained using causal PINN for a significant number of epochs (n_1). Subsequently, the trained parameters are utilized as an initialization for the training of other problems of the physical equation with different initial conditions or for an extended domain, which is trained for a reduced number of epochs (n_2), where $n_2 << n_1$, reducing the computational cost of training the model again from the start. The step-by-step illustration is provided in Fig. 3.2.

In Fig. 3.2, the top horizontal block illustrates the training of causal PINN for the parent model, specifically the primary beam model, either the Euler-Bernoulli or the Timoshenko beam model. Xavier initialization [51] is utilized to train the parent beam model to address the vanishing or exploding gradient problem in neural networks. Proper weight initialization is crucial for stability and convergence. The parent model captures common features applicable to different subcases. The model parameters, generated using Xavier initialization for the initial neural network, undergo a training process adhering to the causal loss function. This training involves the resolution of solutions at lower times prior to approximating at higher times, as shown by the snapshots of the

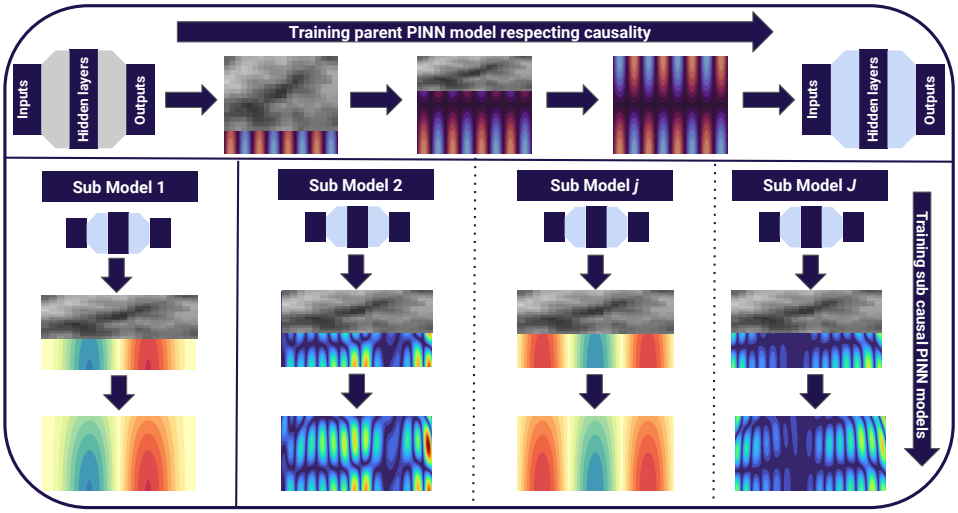


Figure 3.2: **Proposed Transfer Learning Framework in Causal PINN:** The top horizontal block outlines the training process of the causal PINN for the parent model, which is the primary beam model under consideration. The model parameters, initialized using Xavier initialization within the first neural network, undergo training while adhering to causality. The resulting parameters of the trained model serve as the initialization for J subsequent tasks shown by the bottom vertical blocks ($1 \leq j \leq J$). These tasks pertain to different initial conditions and extensions of both spatial and temporal domains. The training of these subtasks is also performed to adhere to causality.

resolved solution. As the number of epochs increases, the model prediction at higher time levels improves only when the solution at lower time levels has been resolved up to a certain accuracy.

The resulting parameters from this training serve as the initialization for subsequent j tasks presented by the bottom vertical blocks in Fig. 3.2. Training subcases involve initializing parameters from the parent model, speeding up convergence and avoiding training from scratch. Reusing parameters reduces computational costs, making the process more efficient. Knowledge transfer from the parent model improves generalization, enabling submodels to adapt effectively to variations in conditions or domains. These subcases involve diverse initial conditions and extensions of both spatial and temporal domains. Notably, the training of these subtasks is also performed by minimizing

the loss terms (3.7) and (3.5) in the loss function (3.3), ensuring a coherent and principled transfer learning framework.

Transferring good knowledge from the parent beam model would accelerate the convergence of subcases. However, dealing with highly complex subcases presents a challenge where improving or optimizing the network may not significantly enhance model accuracy. This situation is akin to the “Kolmogorov complexity" concept, which measures the length of the shortest computer program required to produce a specific output. While not considered in the current work, it is important to consider the Kolmogorov complexity of the parent task and subcases as discussed in [52, 53]. In transfer learning, Kolmogorov complexity becomes pivotal as it captures the intricacy within a dataset or the solution of the PDE in our case. A highly complex solution containing intricate patterns and possible noise can pose challenges for even well-optimized neural networks in extracting meaningful features. This complexity is particularly pertinent in transfer learning, where pre-trained parent models may face challenges in transferring knowledge effectively to a target domain characterized by high intricacy. The diminished transferability of knowledge hampers anticipated improvements in model accuracy. To address this, reducing dataset complexity might be essential. The trained model can better focus on crucial patterns by processing the solution through feature reduction and noise elimination, fostering improved generalization and accuracy in the target task [54, 55].

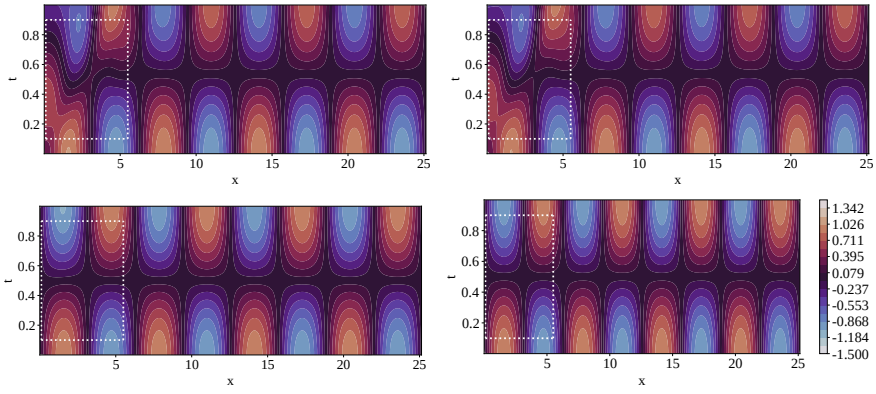


Figure 3.3: Euler-Bernoulli beam displacement on the Winkler foundation **(Top Left.)** Predicted solution using PINN **(Top Right.)** Predicted solution using SA-PINN **(Bottom Left.)** Predicted solution using causal PINN **(Bottom Right.)** Reference solution

The proposed framework addresses key structural engineering issues,

contributing to design, optimization, and control methodologies. Causal PINNs prioritize lower time levels during training, enhancing the understanding of temporal structural behaviours, especially critical for dynamic load responses and environmental changes. Incorporating transfer learning reduces the computational cost, aiding the application of the approach in real-world scenarios. Fusing temporal causality and transfer learning contributes to a larger design space exploration essential for a structural design problem. The proposed framework can be utilized to adapt the control strategies of structures based on knowledge gained from lower time levels [56, 57]. This adaptability is valuable in developing control systems that can respond dynamically to changes in the structural environment, ensuring optimal performance and safety. The proposed method could also help structural health monitoring by updating the model as new data becomes available, enabling real-time monitoring and early detection of potential issues [58, 59].

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

3.5. NUMERICAL EXPERIMENTS

This section presents the numerical experiments for simulating the dynamics of the Euler-Bernoulli and Timoshenko beam models using the proposed framework. The proposed framework is compared with five PINN-based methodologies, namely vanilla PINNs [15], Self-adaptive PINNs (SA-PINN) [60], gradient-enhanced PINN (gPINN) [61], PINNs with adaptive activation function (Adap. PINN) [62], and Wavelet PINN (Wav. PINN) [63]. In addition, leveraging transfer learning, several other experiments are performed for noisy data, different initial conditions, and extrapolation in both spatial and temporal domains for the beam models.

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

hyperparameters λ_1, λ_2 and λ_3 are taken to be 1 each. The selected evaluation metric is the L^2 relative error percentage ($\mathcal{R}$) defined as

$$\mathcal{R} = \frac{||u^* - u||_2}{||u||_2} \times 100 \tag{3.8}$$

where u^* is the approximated PDE solution by the neural network, and u refers to the ground truth. We utilize the trained parameters (μ) of the main model as initialization for training the subcase neural networks for only 1500 epochs, achieving the same level of accuracy as the main model. The configurations were selected using ensemble training through a grid-based search, and for conciseness, only the best-obtained hyperparameters are presented.

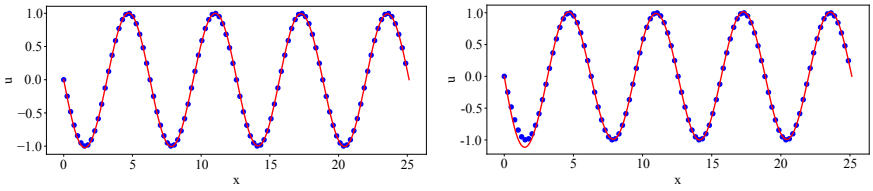


Figure 3.4: Euler-Bernoulli beam equation on the Winkler foundation with noise in the initial condition of displacement of the beam **Left:** Predicted solution at final time ($t = 1$) with 10% Gaussian noise; **Right:** Predicted solution at final time ($t = 1$) with 20% Gaussian noise. Blue dots are ground truth and red curve is prediction.

3.5.1. EULER-BERNOULLI BEAM

The Euler-Bernoulli beam model is a mathematical framework used to analyze the behavior of beams when subjected to loads. It is derived from the three-dimensional elasticity theory or through principles such as Newton's second law or the generalized Hamiltonian Principle [64]. The model assumes certain simplifications: it neglects the effects of rotary inertia and transverse shear deformations. The Euler-Bernoulli beam equation describes the behavior of a beam subjected to bending Fig. 3.1. When the beam is supported on a Winkler foundation, representing an elastic foundation, the Euler-Bernoulli beam equation is modified to account for the interaction between the beam and the foundation. This modified equation considers the stiffness of the foundation and its influence on the behavior of the beam. The mathematical model of a simply supported Euler-Bernoulli beam on a Winkler foundation is described by [33]

$$u_{tt} + u_{xxxx} + p(x,t) = f(x,t), \quad x \in [0, 8\pi], t \in [0, 1] \tag{3.9}$$

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

$$\begin{aligned} u(x,0) = \sin(x), \quad u_t(x,0) = 0 \\ u(0,t) = u(8\pi,t) = u_{xx}(0,t) = u_{xx}(8\pi,t) = 0 \end{aligned} \tag{3.10}$$

The foundation reaction, $p(x,t)$, assumes that the reaction at every location is proportional to the displacement at a particular location, and the springs are linear and independent, as described in (3.9). The reaction force of the foundation is given by $p(x,t) = ku(x,t)$, where $u(x,t)$ is vertical displacement and k is the stiffness of linear springs. The exact solution for this problem is given by $u(x,t) = \sin(x)\cos(\pi t)$.

Solving (3.9), one can determine the vertical displacement of the beam at any point along its length and other important quantities of interest, such as bending moments and beam acceleration. These quantities help engineers understand how the beam will perform structurally and ensure it meets the desired design criteria. By calculating the displacement, engineers can check whether the beam deflects within acceptable limits under the applied loads.

Table 3.1: Euler-Bernoulli beam: $\mathcal{R}$ at $t = 1$ for $k = 1$

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

We simulate (3.9) with five different methods to establish that incorporating causality provides more accuracy in the predicted solution than vanilla PINN, SA-PINN, adaptive activation PINN, wavelet PINN and gPINN for beam dynamics. The results presented in Table 3.1 indicate that vanilla PINN, SA-PINN, adaptive activation PINN, wavelet PINN and gPINN provide less accurate displacement predictions at $t = 1$ for the Euler-Bernoulli equation for stiffness $k = 1$. In contrast, causal PINN yields more accurate displacement predictions as the relative percent error is 0.03. This observation is further supported by the findings depicted in Fig. 3.3, which demonstrates that PINN and SA-PINN models are not accurate, particularly during the initial time, highlighted by the white rectangular box in Fig. 3.3. However, this challenge is effectively overcome by incorporating a causality-respecting loss function, which facilitates training the solution at lower time levels before training at higher times. Additional experiments concerning comparison of the proposed framework with a combination of PINNs and recurrent neural architectures for the Euler-Bernoulli beam equation on the Winkler

foundation are presented in Appendix A. Furthermore, Additional plots for adaptive activation PINN, gPINN and wavelet PINN are presented in Appendix B.

Moreover, we empirically analyze the correlation between weights (w_i) and the corresponding loss at that time step ($L_{\text{PDE}}(t_i, \mu)$) for causal PINNs. We visualize the evolution of the magnitude of weights and errors over epochs for six random weights to understand how their magnitudes impact model training at that time level. The goal is to observe how the error decreases as the magnitudes of the weights increase over epochs, as shown in Fig. 3.5. Six random weights ($w_1, \ldots, w_6$) are considered at time steps $0.06, 0.29, 0.30, 0.41, 0.45,$ and 0.53. From the first row of Fig. 3.5, it is evident that, as training progresses, the magnitude of weights increases and approaches a value of 1. The second and third figures in the first row of Fig. 3.5 provide zoomed-in segments of the first figure of the first row, revealing a sequential convergence pattern at each time level: initially, w_1 converges, followed by w_2, and so forth. After 6000 epochs, all weights nearly converge to 1.

The second row of Fig. 3.5 illustrates the relationship between error and epochs, demonstrating a consistent decrease in error with increasing epochs. The second and third figures in the second row of Fig. 3.5 are zoomed-in segments of first figure of the second row in Fig. 3.5, showing a sequential reduction in error corresponding to an increase in the magnitude of weight. It is evident that as the weights sequentially increase at each time level, the error also decreases sequentially. The observed pattern suggests that weights first converge at lower time levels before progressing to subsequent levels, gradually improving accuracy.

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

NOISY INITIAL CONDITIONS

This subsection presents the performance of the proposed method with noisy initial conditions. Initial conditions may not be perfectly known in real-world scenarios or contain uncertainties or noise. By learning displacements for noisy initial conditions, we can develop models that accurately represent the behavior of the system under such realistic conditions, allowing us to account for uncertainties and better understand the actual response of the system. To observe the dynamics of beam models under these conditions, we introduce Gaussian noise in the initial condition ranging from 5% to 20%. The hyperparameter selection is the same as the main model, except for the number of epochs. With transfer learning, we perform 1500 epochs instead of

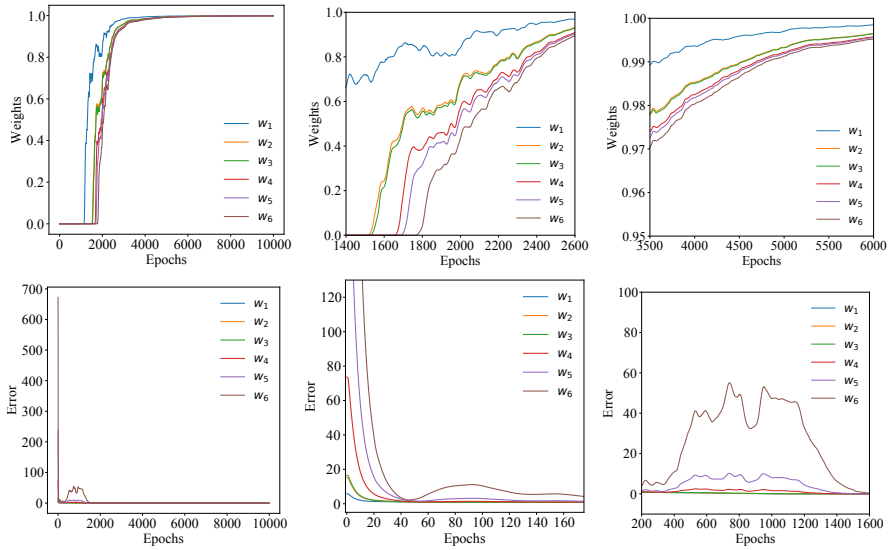


Figure 3.5: Variation of weights (w_i) and the corresponding loss at that time step ($L_{\text{PDE}}(t_i, \mu)$) over training epochs. **(Top Left.)** Magnitude of six random weights ($w_1, \ldots, w_6$) varying with epochs, where $w_1 < w_2 \ldots < w_6$. **(Top Middle and Right.)** Zoomed-in segments of **(Top Left.)**. **(Bottom Left.)** Relative percent error at six distinct time levels ($L_{\text{PDE}}(t_i, \mu)$) corresponding to weights ($w_1, \ldots, w_6$) varying with epochs. **(Bottom Middle and Right.)** Zoomed-in segments of **(Bottom Right.)**.

10000.

Table 3.2 presents the results from 5% to 20% Gaussian noise levels in the initial conditions for the displacement of the beam with and without (w/o) using transfer learning. The proposed method predicts u with less relative error percent. This prediction is significantly more accurate compared to the case without transfer learning. Also, Fig. 3.4 shows the results for 10% and 20% noise levels in the initial conditions for the displacement of the beam using transfer learning, demonstrating the computational efficiency of the proposed method.

Fig. 3.6 illustrates the comparison of relative error percentages concerning the noise percentage for both methods, one with transfer learning and the other without it. In the transfer learning scenario, it becomes apparent that an increase in the noise percentage results in a corresponding increase in the relative error percentage. When the subcases use the trained parameters for initialization, noise and error percentages exhibit a direct proportional relationship. However, in

cases where trained parameters are not utilized, no discernible pattern emerges due to the non-convergence in minimizing the loss function.

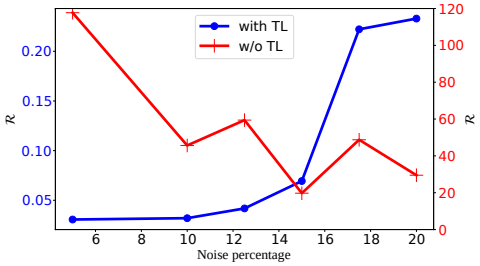


Figure 3.6: $\mathcal{R}$ vs noise percentage in the initial condition for the Euler Bernoulli beam for both approaches - with and without transfer learning.

Table 3.2: Euler Bernoulli Beam: $\mathcal{R}$ at $t = 1$ for different percentages of noise in the initial condition. "TL" refers to transfer learning, and "w/o" refers to without. Abbreviations are used consistently for the following tables.

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

DIFFERENT INITIAL DISPLACEMENTS AND VELOCITIES

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

Here, we consider different initial conditions compared to the parent model. The initial conditions for this case are $u(x, 0) = a\sin(x)$ and $u_t(x, t = 0) = a\sin(x)$. The analytical solution for the corresponding problem is $u(x, t) = a\sin(x)e^t$. We utilize the trained parameters of the Euler-Bernoulli beam model as an initialization for training this problem with different initial conditions considering $a = 1, 2$ (representing case 1 and case 2 in Table 3.3). The hyperparameters remain unchanged; the

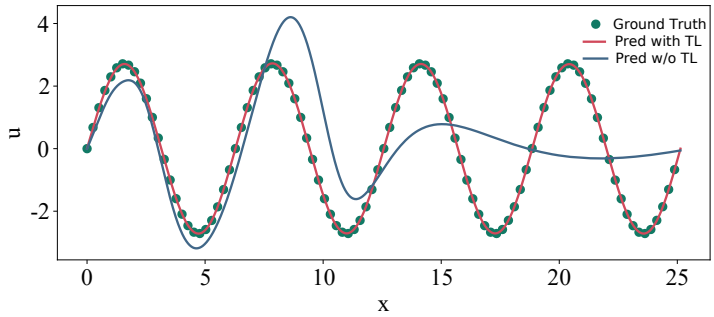


Figure 3.7: Euler-Bernoulli beam on the Winkler foundation for initial velocity for case 1: Causal PINN prediction at final time $t = 1$ with and without transfer learning

only change is the number of epochs, which is only 3000. Relative error percentages of displacement are presented in Table 3.3, which shows a large difference in relative percent errors. From Fig. 3.7, it is evident for the first case that the transfer learning approach achieves accurate predictions in fewer epochs.

Table 3.3: Euler Bernoulli Beam: $\mathcal{R}$ at $t = 1$ for different velocities

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

3.5.2. TIMOSHENKO BEAM

The Timoshenko beam theory considers the shear deformation and rotational effects neglected in the Euler-Bernoulli beam equation [64]. Hence, in addition to the quantity vertical displacement (u), Timoshenko theory considers the cross-sectional rotation (θ) as another unknown variable. The mathematical model for a beam resting on a Winkler foundation and subjected to an external load based on the Timoshenko beam theory is given as follows [33]

$$\begin{aligned} \theta_{tt} - \theta_{xx} + (\theta - u_x) &= 0; \\ u_{tt} + (\theta - u_x)_x + ku &= h(x,t) \end{aligned} \tag{3.11}$$

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

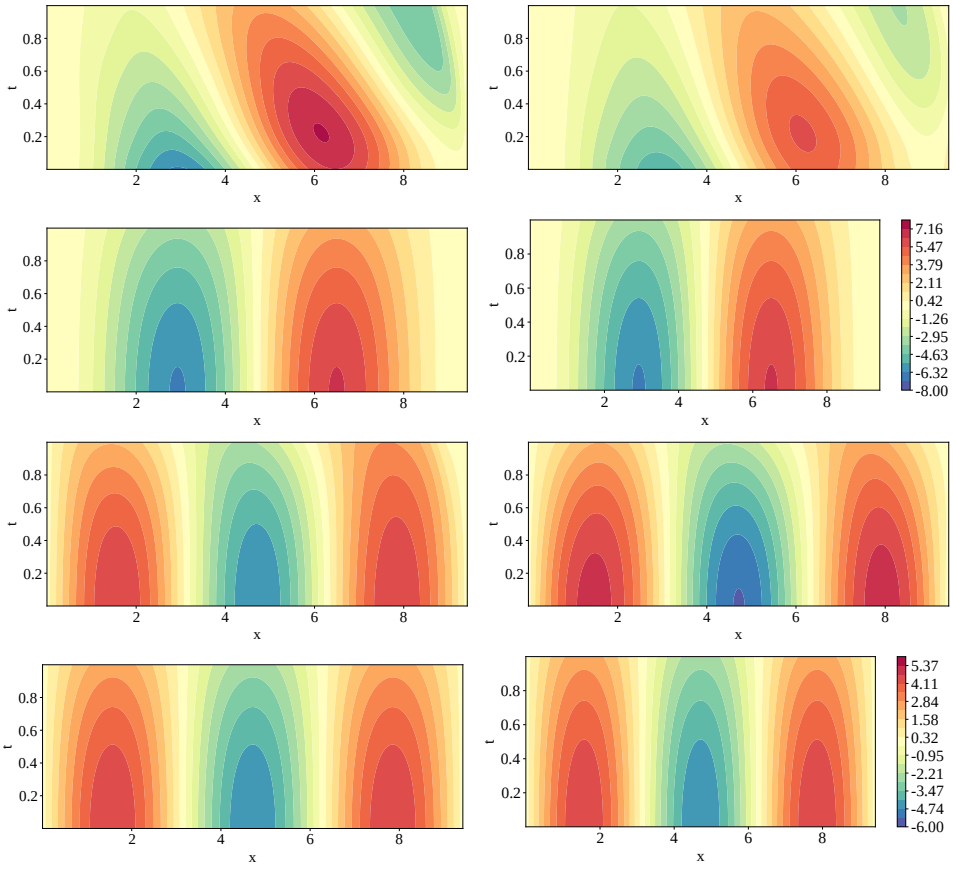


Figure 3.8: Timoshenko beam on the Winkler foundation **First row:** Predicted Displacement (u^*) **(Left.)** Using PINN **(Right.)** Using SA-PINN **(Second row, Left.)** Using causal PINN **(Right.)** Reference solution **Third row:** Predicted Rotation (θ^*) **(Left.)** Using PINN **(Right.)** Using SA-PINN **(Fourth row, Left.)** Using causal PINN **(Right.)** Reference solution

$$\begin{aligned} \theta(x,0) &= \frac{3\pi}{2}\cos(x) + \left(x - \frac{3\pi}{2}\right), \quad \theta_t(x,0) = 0 \\ u(x,0) &= \frac{3\pi}{2}\sin(x), \quad u_t(x,0) = 0 \\ \theta(0,t) &= \theta(3\pi,t) = u(0,t) = u(3\pi,t) = 0 \end{aligned} \tag{3.12}$$

The analytic solution for the rotation and vertical displacement is

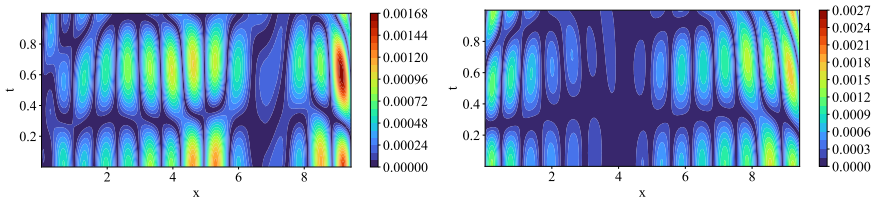


Figure 3.9: Timoshenko beam on the Winkler foundation at final time $t = 1$. **Left:** Absolute error in predicting displacement ($|u - u^*|$); **Right:** Absolute error in predicting rotation ($|\theta - \theta^*|$)

given as follows

$$\begin{aligned} \theta(x,t) &= \left(\frac{3\pi}{2}\cos(x) + \left(x - \frac{3\pi}{2}\right)\right)\cos(t) \\ u(x,t) &= \frac{3\pi}{2}\sin(x)\cos(t) \end{aligned} \tag{3.13}$$

Solving the Timoshenko beam model (3.11) - (3.12) would help engineers obtain more accurate predictions of beam deflections and rotations, especially for beams with high aspect ratios or subjected to high shear forces. This accuracy is crucial for assessing structural integrity, ensuring compliance with design criteria, and preventing potential failures.

Fig. 3.8 illustrates the predicted displacement and rotation throughout the entire space-time domain. Fig. 3.8(c) and Fig. 3.8(g) depict the displacement and rotation prediction using the causal PINN loss function. Fig. 3.8(a-b) and Fig. 3.8(e-f) depict the displacement and rotation prediction using vanilla PINN and SA-PINNs, respectively, illustrating its failure in prediction. Additional plots for adaptive activation PINN, gPINN and wavelet PINN are presented in Appendix B. Furthermore, Fig. 3.9 presents the absolute error in displacement and rotation resulting from the causal PINN loss function. The maximum error magnitude falls below 10^{-2}, clearly indicating the accuracy of causal PINN.

Table 3.4: Timoshenko Beam: $\mathcal{R}$ at $t = 1$ for $k = 1$

	PINN	SA-PINN	gPINN	Adap. PINN	Wav. PINN	Causal PINN
u^*	119.17	137.15	119.17	240.78	238.46	1.2×10^{-6}
θ^*	9.18	6.56	38.63	9.16	9.10	7.7×10^{-6}

Table 3.4 presents the relative percentage errors in predicting displacement and rotation for vanilla PINN, SA-PINN, gPINN, adaptive

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

3.5.3. LARGE SPACE-TIME HORIZON

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

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

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

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

EXTENDED SPATIAL DOMAIN

In this section, we consider the Timoshenko beam model in an extended domain in space. The spatial domain for the parent model is $x \in [0, 3\pi]$. Here, we utilize the parameters of the parent model and train the subsequent models for different spatial domains, in particular $x \in [0, 5\pi]$, $x \in [0, 6\pi]$, and $x \in [0, 7\pi]$. The aim is to observe the potential of the method in a larger domain, indicating that the method generalizes well. The results obtained with and without transfer learning

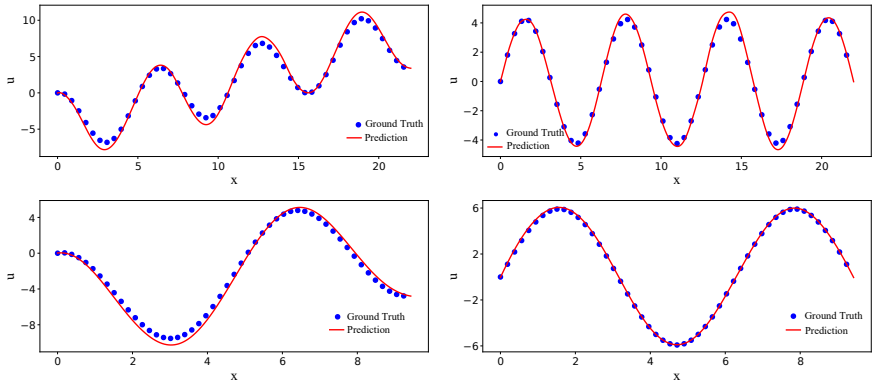


Figure 3.10: Timoshenko beam on the Winkler foundation: **Top** Prediction for extended domain in space, $x \in [0, 7\pi]$ **Left:** Displacement (u^*); **Right:** Rotation (θ^*). **Bottom** Prediction for extended domain in time for $t = 7$ **Left:** Displacement (u^*); **Right:** Rotation (θ^*).

are presented in Table 3.5, highlighting the superior accuracy achieved by the proposed method when utilizing parameters from the main model compared to training the model with Xavier initialization. Fig. 3.10 top row presents the predictions of displacement and rotation by the proposed method, indicating that the model generalizes well across the spatial domain, inheriting the underlying structure and symmetry of the solution.

Table 3.5: Timoshenko Beam: $\mathcal{R}$ for extension in the spatial domain with 3000 epochs

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

EXTENDED TEMPORAL DOMAIN

We now extend our investigation to the temporal domain based on successfully generalizing the proposed method in the spatial domain. By employing the trained parameters obtained from the parent model, we train the same model with an extension in time, considering different temporal domains, $t \in [0, 4]$, $t \in [0, 6]$, and $t \in [0, 7]$. The relative error

percentage for all cases of the extended temporal domains is presented in Table 3.6. We observe that the proposed method accurately predicts displacement and rotation, while the approach without transfer learning fails to provide the same level of accuracy. Fig. 3.10 bottom row shows the predictions obtained by the proposed method for displacement and rotation in an extended temporal domain. The results show that utilizing transfer learning for extended domains in space and time provides accurate results, conserving the structure and symmetry of the solution.

3.6. CONCLUSIONS

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

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

Table 3.6: Timoshenko Beam: $\mathcal{R}$ for extension in the temporal domain with 3000 epochs

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

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

Overall, our findings highlight the efficacy of the proposed methodology in simulating beam dynamics under diverse engineering scenarios. By leveraging transfer learning and a causality-respecting PINN framework, we can reduce training requirements while achieving accurate

results for various cases. This research opens up new possibilities for efficiently predicting the dynamics of structural elements, leading to advancements in structural engineering design, optimization, and control.

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

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

REFERENCES

[1] A. C. Lamprea-Pineda, D. P. Connolly, and M. F. Hussein. “Beams on elastic foundations–A review of railway applications and solutions”. In: *Transportation Geotechnics* 33 (2022), p. 100696. doi: 10.1016/j.trgeo.2021.100696.

[2] J. Deng, M. Shahroudi, and K. Liu. “Dynamic Stability and Responses of Beams on Elastic Foundations Under a Parametric Load”. In: *International Journal of Structural Stability and Dynamics* 23.02 (2023), p. 2350018. doi: 10.1142/S0219455423500189.

[3] E. Tsudik. *Analysis of structures on elastic foundations*. J. Ross Publishing, 2012.

[4] H. Kabir and M. Aghdam. “A robust Bézier based solution for nonlinear vibration and post-buckling of random checkerboard graphene nano-platelets reinforced composite beams”. In: *Composite Structures* 212 (2019), pp. 184–198. doi: 10.1016/j.compstruct.2019.01.041.

[5] M. Hetényi and M. I. Hetbenyi. *Beams on elastic foundation: theory with applications in the fields of civil and mechanical engineering*. Vol. 16. University of Michigan press Ann Arbor, MI, 1946.

[6] L. G. Petrosian. *Analysis of Structures on Elastic Foundation: Incorporating the Spectral Method of Boundary Elements*. CRC Press, 2022. doi: 10.1201/9781003273837.

[7] P. B. Chang, B. J. Williams, T. J. Santner, W. I. Notz, and D. L. Bartel. “Robust Optimization of Total Joint Replacements Incorporating Environmental Variables”. In: *Journal of Biomechanical Engineering* 121.3 (June 1999), pp. 304–310. issn: 0148-0731. doi: 10.1115/1.2798325.

[8] E. Madenci and I. Guven. *The finite element method and applications in engineering using ANSYS®*. Springer, 2015. doi: 10.1007/978-1-4899-7550-8.

[9] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440. doi: 10.1038/s42254-021-00314-5.

[10] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.* “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), pp. 583–589. doi: 10.1038/s41586-021-03819-2.

[11] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. “Machine learning and the physical sciences”. In: *Reviews of Modern Physics* 91.4 (2019), p. 045002. doi: 10.1103/RevModPhys.91.045002.

[12] M. U. Erdaş, M. Kopar, B. S. Yildiz, and A. R. Yildiz. “Optimum design of a seat bracket using artificial neural networks and dandelion optimization algorithm”. In: *Materials Testing* 65.12 (2023), pp. 1767–1775. doi: 10.1515/mt-2023-0201.

[13] C. M. Aye, N. Pholdee, A. R. Yildiz, S. Bureerat, and S. M. Sait. “Multi-surrogate-assisted metaheuristics for crashworthiness optimisation”. In: *International Journal of Vehicle Design* 80.2-4 (2019), pp. 223–240. doi: 10.1504/IJVD.2019.109866.

[14] A. Yildiz, N. Öztürk, N. Kaya, and F. Öztürk. “Integrated optimal topology design and shape optimization using neural networks”. In: *Structural and Multidisciplinary Optimization* 25 (2003), pp. 251–260. doi: 10.1007/s00158-003-0300-0.

[15] M. Raissi, P. Perdikaris, and G. E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational physics* 378 (2019), pp. 686–707. doi: 10.1016/j.jcp.2018.10.045.

[16] A. Karpatne, W. Watkins, J. Read, and V. Kumar. “Physics-guided neural networks (PGNN): An application in lake temperature modeling”. In: *arXiv preprint arXiv:1710.11431* 2 (2017). doi: 10.48550/arXiv.1710.11431.

[17] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar. “Theory-guided data science: A new paradigm for scientific discovery from data”. In: *IEEE Transactions on Knowledge and Data Engineering* 29.10 (2017), pp. 2318–2331. doi: 10.1109/TKDE.2017.2720168.

[18] A. Karpatne, R. Kannan, and V. Kumar. *Knowledge Guided Machine Learning: Accelerating Discovery Using Scientific Knowledge and Data*. CRC Press, 2022.

[19] Z. Meng, Q. Qian, M. Xu, B. Yu, A. R. Yıldız, and S. Mirjalili. “PINN-FORM: A new physics-informed neural network for reliability analysis with partial differential equation”. In: *Computer Methods in Applied Mechanics and Engineering* 414 (2023), p. 116172. doi: 10.1016/j.cma.2023.116172.

[20] S. Wang, S. Sankaran, and P. Perdikaris. “Respecting causality is all you need for training physics-informed neural networks”. In: *arXiv preprint arXiv:2203.07404* (2022). doi: 10.48550/arXiv.2203.07404.

[21] O. Fuks and H. A. Tchelepi. “Limitations of physics informed machine learning for nonlinear two-phase transport in porous media”. In: *Journal of Machine Learning for Modeling and Computing* 1.1 (2020). doi: 10.1615/JMachLearnModelComput.2020033905.

[22] J. Kim, K. Lee, D. Lee, S. Y. Jhin, and N. Park. “DPM: A novel training method for physics-informed neural networks in extrapolation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 8146–8154. doi: 10.48550/arXiv.2012.02681.

[23] T. Kapoor, A. Chandra, D. M. Tartakovsky, H. Wang, A. Nunez, and R. Dollevoet. “Neural oscillators for generalization of physics-informed machine learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 12. 2024, pp. 13059–13067. doi: 10.48550/arXiv.2308.08989.

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

[25] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. “Characterizing possible failure modes in physics-informed neural networks”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[26] A. Dekhovich, M. H. Sluiter, D. M. Tax, and M. A. Bessa. “iPINNs: Incremental learning for Physics-informed neural networks”. In: *arXiv preprint arXiv:2304.04854* (2023). doi: 10.48550/arXiv.2304.04854.

[27] T. De Ryck, F. Bonnet, S. Mishra, and E. de Bézenac. “An operator preconditioning perspective on training in physics-informed machine learning”. In: *arXiv preprint arXiv:2310.05801* (2023). doi: 10.48550/arXiv.2310.05801.

[28] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis. “Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems”. In: *Computer Methods in Applied Mechanics and Engineering* 365 (2020), p. 113028. doi: 10.1016/j.cma.2020.113028.

[29] P. Lippe, B. S. Veeling, P. Perdikaris, R. E. Turner, and J. Brandstetter. “PDE-refiner: Achieving accurate long rollouts with neural pde solvers”. In: *arXiv preprint arXiv:2308.05732* (2023). doi: 10.48550/arXiv.2308.05732.

[30] A. D. Jagtap and G. E. Karniadakis. “Extended Physics-informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition based Deep Learning Framework for Nonlinear Partial Differential Equations.” In: *AAAI spring symposium: MLPS*. Vol. 10. 2021.

[31] K. Shukla, A. D. Jagtap, and G. E. Karniadakis. “Parallel physics-informed neural networks via domain decomposition”. In: *Journal of Computational Physics* 447 (2021), p. 110683. doi: 10.1016/j.jcp.2021.110683.

[32] S. Niu, Y. Liu, J. Wang, and H. Song. “A decade survey of transfer learning (2010–2020)”. In: *IEEE Transactions on Artificial Intelligence* 1.2 (2020), pp. 151–166. doi: 10.1109/TAI.2021.3054609.

[33] D. Younesian, A. Hosseinkhani, H. Askari, and E. Esmailzadeh. “Elastic and viscoelastic foundations: a review on linear and nonlinear vibration modeling and applications”. In: *Nonlinear Dynamics* 97.1 (2019), pp. 853–895. doi: 10.1007/s11071-019-04977-9.

[34] S. Goswami, C. Anitescu, S. Chakraborty, and T. Rabczuk. “Transfer learning enhanced physics informed neural network for phase-field modeling of fracture”. In: *Theoretical and Applied Fracture Mechanics* 106 (2020), p. 102447. doi: 10.1016/j.tafmec.2019.102447.

[35] S. Li, G. Wang, Y. Di, L. Wang, H. Wang, and Q. Zhou. “A physics-informed neural network framework to predict 3D temperature field without labeled data in process of laser metal deposition”. In: *Engineering Applications of Artificial Intelligence* 120 (2023), p. 105908. doi: 10.1016/j.engappai.2023.105908.

[36] X. Liu, W. Peng, Z. Gong, W. Zhou, and W. Yao. “Temperature field inversion of heat-source systems via physics-informed neural networks”. In: *Engineering Applications of Artificial Intelligence* 113 (2022), p. 104902. doi: 10.1016/j.engappai.2022.104902.

[37] A. M. Roy and S. Guha. “A data-driven physics-constrained deep learning computational framework for solving von Mises plasticity”. In: *Engineering Applications of Artificial Intelligence* 122 (2023), p. 106049. doi: 10.1016/j.engappai.2023.106049.

[38] E. Haghighat, S. Abouali, and R. Vaziri. “Constitutive model characterization and discovery using physics-informed deep learning”. In: *Engineering Applications of Artificial Intelligence* 120 (2023), p. 105828. doi: 10.1016/j.engappai.2023.105828.

[39] B. Mu, B. Qin, S. Yuan, X. Wang, and Y. Chen. “PIRT: A Physics-Informed Red Tide Deep Learning Forecast Model Considering Causal-Inferred Predictors Selection”. In: *IEEE Geoscience and Remote Sensing Letters* 20 (2023), pp. 1–5. doi: 10.1109/LGRS.2023.3250642.

[40] M. Penwarden, A. D. Jagtap, S. Zhe, G. E. Karniadakis, and R. M. Kirby. “A unified scalable framework for causal sweeping strategies for Physics-Informed Neural Networks (PINNs) and their temporal decompositions”. In: *10.2139/ssrn.4388745* (2023).

[41] M. Bazmara, M. Silani, M. Mianroodi, *et al.* “Physics-informed neural networks for nonlinear bending of 3D functionally graded beam”. In: *Structures*. Vol. 49. Elsevier. 2023, pp. 152–162. doi: 10.1016/j.istruc.2023.01.115.

[42] T. Kapoor, H. Wang, A. Núñez, and R. Dollevoet. “Physics-Informed Neural Networks for Solving Forward and Inverse Problems in Complex Beam Systems”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023), pp. 1–15. doi: 10.1109/TNNLS.2023.3310585.

[43] T. Kapoor, H. Wang, A. Núñez, and R. Dollevoet. “Physics-informed machine learning for moving load problems”. In: *arXiv preprint arXiv:2304.00369* (2023). doi: 10.48550/arXiv.2304.00369.

[44] J. Lee. “Physics Informed Neural Networks for Extreme Mechanics Problems”. In: *Available at SSRN 4362563* (). doi: 10.2139/ssrn.4362563.

[45] C. Xu, B. T. Cao, Y. Yuan, and G. Meschke. “Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios”. In: *Computer Methods in Applied Mechanics and Engineering* 405 (2023), p. 115852. doi: 10.1016/j.cma.2022.115852.

[46] A. Fallah and M. M. Aghdam. “Physics-informed neural network for bending and free vibration analysis of three-dimensional functionally graded porous beam resting on elastic foundation”. In: *Engineering with Computers* (2023), pp. 1–18. doi: 10.1007/s00366-023-01799-7.

[47] Q. Chen, Q. Ye, W. Zhang, H. Li, and X. Zheng. “TGM-Nets: A deep learning framework for enhanced forecasting of tumor growth by integrating imaging and modeling”. In: *Engineering Applications of Artificial Intelligence* 126 (2023), p. 106867. doi: 10.1016/j.engappai.2023.106867.

[48] N. Zobeiry and K. D. Humfeld. “A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications”. In: *Engineering Applications of Artificial Intelligence* 101 (2021), p. 104232. doi: 10.1016/j.engappai.2021.104232.

[49] S. Shen, H. Lu, M. Sadoughi, C. Hu, V. Nemani, A. Thelen, K. Webster, M. Darr, J. Sidon, and S. Kenny. “A physics-informed deep learning approach for bearing fault detection”. In: *Engineering Applications of Artificial Intelligence* 103 (2021), p. 104295. doi: 10.1016/j.engappai.2021.104295.

[50] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. “Automatic differentiation in pytorch”. In: (2017).

[51] X. Glorot and Y. Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[52] V. Bolon-Canedo and B. Remeseiro. “Feature selection in image analysis: a survey”. In: *Artificial Intelligence Review* 53.4 (2020), pp. 2905–2931. doi: 10.1007/s10462-019-09750-3.

[53] H. Kabir and N. Garg. “Machine learning enabled orthogonal camera goniometry for accurate and robust contact angle measurements”. In: *Scientific Reports* 13.1 (2023), p. 1497. doi: 10.1038/s41598-023-28763-1.

[54] E. Olivares, H. Ye, A. Herrero, B. A. Nia, Y. Ren, R. P. Donovan, *et al.* “Applications of information channels to physics-informed neural networks for WiFi signal propagation simulation at the edge of the industrial internet of things”. In: *Neurocomputing* 454 (2021), pp. 405–416. doi: 10.1016/j.neucom.2021.04.021.

[55] T. Whittaker, R. A. Janik, and Y. Oz. “Neural network complexity of chaos and turbulence”. In: *The European Physical Journal E* 46.7 (2023), p. 57. doi: 10.1140/epje/s10189-023-00321-7.

[56] R. Faria, B. Capron, A. Secchi, and M. De Souza Jr. “A data-driven tracking control framework using physics-informed neural networks and deep reinforcement learning for dynamical systems”. In: *Engineering Applications of Artificial Intelligence* 127 (2024), p. 107256. doi: 10.1016/j.engappai.2023.107256.

[57] F. Arnold and R. King. “State–space modeling for control based on physics-informed neural networks”. In: *Engineering Applications of Artificial Intelligence* 101 (2021), p. 104195. doi: 10.1016/j.engappai.2021.104195.

[58] H. T. Mai, T. T. Truong, J. Kang, D. D. Mai, and J. Lee. “A robust physics-informed neural network approach for predicting structural instability”. In: *Finite Elements in Analysis and Design* 216 (2023), p. 103893. doi: 10.1016/j.finel.2022.103893.

[59] T. Liu and H. Meidani. “Physics-Informed Neural Networks for System Identification of Structural Systems with a Multiphysics Damping Model”. In: *Journal of Engineering Mechanics* 149.10 (2023), p. 04023079. doi: 10.1061/JENMDT.EMENG-7060.

[60] L. D. McClenny and U. M. Braga-Neto. “Self-adaptive physics-informed neural networks”. In: *Journal of Computational Physics* 474 (2023), p. 111722. doi: 10.1016/j.jcp.2022.111722.

[61] J. Yu, L. Lu, X. Meng, and G. E. Karniadakis. “Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems”. In: *Computer Methods in Applied Mechanics and Engineering* 393 (2022), p. 114823. doi: 10.1016/j.cma.2022.114823.

[62] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis. “Adaptive activation functions accelerate convergence in deep and physics-informed neural networks”. In: *Journal of Computational Physics* 404 (2020), p. 109136. doi: 10.1016/j.jcp.2019.109136.

[63] Z. Uddin, S. Ganga, R. Asthana, and W. Ibrahim. “Wavelets based physics informed neural networks to solve non-linear differential equations”. In: *Scientific Reports* 13.1 (2023), p. 2882. doi: 10.1038/s41598-023-29806-3.

[64] A. Öchsner. *Classical beam theories of structural mechanics*. Vol. 42. Springer, 2021. doi: 10.1007/978-3-030-76035-9.

4

NEURAL OSCILLATORS FOR GENERALIZATION OF PIML

A primary challenge of physics-informed machine learning (PIML) is its generalization beyond the training domain, especially when dealing with complex physical problems represented by partial differential equations (PDEs). This chapter aims to enhance the generalization capabilities of PIML, facilitating practical, real-world applications where accurate predictions in unexplored regions are crucial. We leverage the inherent causality and temporal sequential characteristics of PDE solutions to fuse PIML models with recurrent neural architectures based on systems of ordinary differential equations, referred to as neural oscillators. Through effectively capturing long-time dependencies and mitigating the exploding and vanishing gradient problem, neural oscillators foster improved generalization in PIML tasks. Extensive experimentation involving time-dependent nonlinear PDEs and biharmonic beam equations demonstrates the efficacy of the proposed approach. Incorporating neural oscillators outperforms existing state-of-the-art methods on benchmark problems across various metrics. Consequently, the proposed method improves the generalization capabilities of PIML, providing accurate solutions for extrapolation and prediction beyond the training data.

Apart from minor updates, this chapter has been published as: Kapoor, T., Chandra, A., Tartakovsky, D. M., Wang, H., Nunez, A., & Dollevoet, R. (2024, March). Neural oscillators for generalization of physics-informed machine learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 12, pp. 13059-13067).

4.1. INTRODUCTION

In machine learning and artificial intelligence, *generalization* refers to the ability of a model to perform on previously unseen data beyond its training domain. This entails prediction of outcomes for a sample $\mathbf{x}$ that lies outside the convex hull of the training set $X = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$, where $\mathbf{N}$ is the number of training samples [1]. Current deep-learning models exhibit robust generalization on tasks like image [2], and speech recognition [3], among others [4]. In physical sciences, state-of-the-art deep-learning models, also known as data-driven approaches, learn patterns and correlations from training data but lack intrinsic comprehension of the underlying governing laws of the problem [5, 6]. Despite their effective approximation of complex functions and relationships, these data-driven methods face challenges in generalizing to scenarios significantly different from the training distribution, resulting in a physical-agnostic methodology [7].

Limitations of data-driven methods, characterized by their inability to adhere to physical laws and their agnosticism towards underlying physics, underscore the need for deep learning models capable of effectively capturing fundamental physical phenomena, such as their structure and symmetry [8]. Adopting such learning approaches promises to enhance the generalization capabilities of the model significantly. Consequently, a growing interest has been in embedding physics principles into machine learning to develop physics-aware models such as physics-informed neural networks (PINNs) [9]. PINNs consider mathematical models of the underlying physical process, represented as partial differential equations (PDEs), and integrate them into the loss function during training.

Despite their popularity, experimental evidence suggests that PINNs might fail to generalize. Minimizing the PDE residual in PINN does not straightforwardly control the generalization error [10, 11]. Although PINNs and their subsequent enhancement aim to incorporate soft or hard physical constraints for robustness, they often struggle to achieve strong generalization [12–14]. Hence, simply embedding physical equations into the loss function need not necessarily guarantee genuine physics awareness or robustness beyond the training domain. Ideally, a physics-informed model must reproduce known physics in the training domain and exhibit predictive capabilities for new scenarios while respecting conservation laws and effectively handling variations and uncertainties in real-world applications. Attaining this level of physics awareness remains a crucial challenge in developing dependable and powerful physics-informed machine learning methods [15, 16].

One way to enhance the extrapolation power of PINNs is to dynamically manipulate the gradients of the loss terms, building upon a gradient-based optimizer [12]. This method shares similarities with gradient-based techniques employed in domain generalization tasks

[17]. However, one drawback of such methods is the need for training until a specific user-defined tolerance in the loss is achieved, resulting in convergence issues and increased computational costs. We adopt a different strategy to tackle the generalization challenge by leveraging the inherent *causality* present in PDE solutions [18]. Leveraging causality enables us to enhance generalizability by learning the underlying dynamics that preserve the structure and symmetry of the underlying problem.

A recurrent neural network (RNN) might be capable of learning the dynamics owing to its remarkable success in various sequential tasks. Gated architectures, like long short-term memory (LSTM) [19] and gated recurrent unit (GRU) [20], have been mooted to address the exploding and vanishing gradient problem (EVGP) in vanilla RNNs [21]. However, EVGP can remain a concern as presented by Li *et al.* RNNs with orthogonality constraints on recurrent weight matrices are used to tackle EVGP [23–26]. While this strategy alleviates EVGP, it may reduce expressivity and hinder performance in practical tasks [26]. We posit that *neural oscillators* [27] offers a practical means to achieve high expressibility and mitigate EVGP. Neural oscillators use ordinary differential equations (ODEs) to update the hidden states of the recurrent unit, enabling efficient dynamic learning.

This chapter introduces a new approach to address the generalization challenge. It employs a physics-informed neural architecture that learns the underlying dynamics in the training domain, followed by a neural oscillator to exploit the causality and learn temporal dependencies between the solutions at subsequent time levels. This extension of a physics-informed architecture helps increase the accuracy of a generalization task since neural oscillators carry a hidden state that retains information from previous time steps, enabling the model to capture and leverage temporal dependencies in the data.

We consider two different neural oscillators: *coupled oscillatory recurrent neural network* (CoRNN) [28] and *long expressive memory* (LEM) [29]. Both methods use a coupled system of ODEs to update the hidden states. We ascertain the relative performance of these two oscillators on three benchmark nonlinear problems: viscous Burgers equation, Allen–Cahn equation, and Schrödinger equation. Additionally, we evaluate the performance of our method in generalizing a solution for the Euler–Bernoulli beam equation. To showcase the performance of the proposed framework for higher-dimensional PDEs, we performed an experiment on 2D Kovasznay flow as presented in supplementary material **SM**§C provided at [30]

The remainder of this chapter is structured as follows. The “Related Work" section provides an overview of pertinent literature and recent studies related to the current work. In the “Method" section, our approach for enhancing the generalization of physics-informed machine

learning through integration with a neural oscillator is explained in detail. Our method is validated through a series of numerical experiments in the “Numerical Experiments" section. Finally, key findings and implications of this study are collated in the “Conclusions" section.

4.2. RELATED WORK

PIML

Our research aims to advance physics-informed models, a subset of machine learning techniques that address physical problems formulated as PDEs. PIML encompasses a range of methodologies, including physics-informed [31], physics-based [32], physics-guided [33], and theory-guided [32] approaches. The review papers [31, 32] provide a comprehensive overview of progress in PIML. Recently, PIML has demonstrated considerable utility in scientific and engineering disciplines, encompassing fluid dynamics [34] and materials science [35], among others. Our primary focus is to improve PIML variants that integrate governing equations into the loss function during training to foster generalization, which involves advancing PINNs and their variations, such as causal PINNs [18], and self-adaptive PINNs [36].

DOMAIN GENERALIZATION

Domain generalization focuses on training models to effectively handle unseen domains with diverse data distributions, even when trained on data from related but distinct domains [4, 17]. In contrast, domain adaptation involves transferring knowledge from a labeled source domain to an unlabeled or partially labeled target domain, assuming access to some labeled data in the target domain [37]. Our research shares the core principles with these fields but differs in that we learn *exclusively* from a single training set without using multiple domains, as in domain generalization, or having access to any target domain data, as in domain adaptation. Moreover, we do not employ any transfer learning techniques. Our task is to train *solely* on the training set and *directly deploy* the trained model on the test region.

GENERALIZATION IN PIML

Despite limited research on the generalization of physics-informed models, some studies have specifically focused on generalizing PINNs. One noteworthy approach is the dynamic pulling method (DPM) [12], which utilizes a gradient-based technique to extend the solution of nonlinear benchmark problems beyond the trained convex hull X, focusing on generalizing solutions in the temporal domain. Other investigations have centered on generalizing the parameter space

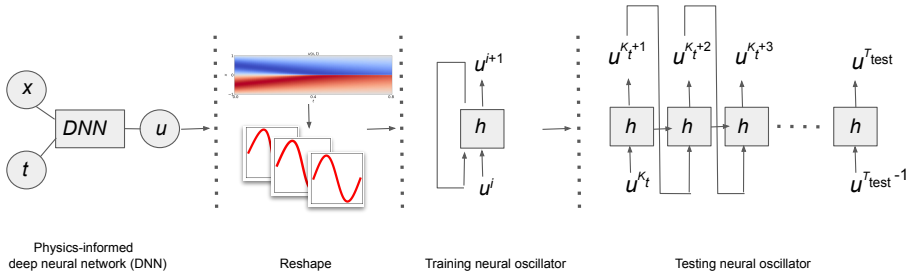


Figure 4.1: The proposed framework in which a physics-informed architecture (e.g., PINN or its variants) learns a solution in the convex hull X_1. After reshaping, these solutions are represented sequentially and processed by one of the neural oscillators. The neural oscillator is finally tested in the convex testing hull X_2, where the output of the last prediction step is the input for the next prediction step. Here, $1 \leq i \leq k_t$, $i \in \mathbb{Z}$, and $h = [y, z]$. The dotted lines separate different stages of training and testing the framework.

for parametric PDEs, employing techniques like curriculum learning, sequence-to-sequence learning [38] and incremental learning [39]. However, these approaches involve training and testing *within the convex hull* of the parameter space, which differs from the focus and approach to our work.

NEURAL OSCILLATORS

Oscillator networks are ubiquitous in natural and engineering systems, exemplified by pendulums (classical mechanics) and heartbeats (biology). A growing trend involves building RNN architectures based on ODEs and dynamical systems [40–43]. Recent research has abstracted the fundamental nature of functional brain circuits as networks of oscillators, constructing RNNs using simpler mechanistic systems represented by ODEs while disregarding complex biological neural function details. Driven by the *long-term memory* of these oscillators and inspired by the *universal approximation property* [27], our goal is to integrate them with physics-informed models to enhance generalization.

4.3. METHOD

The proposed framework comprises a feedforward neural network informed by physics (such as PINN, causal PINN, self-adaptive PINN, or any other physics-guided architecture), followed by a neural oscillator.

For example, we combine PINN with the coupled oscillatory recurrent neural network (CoRNN) or the long expressive memory (LEM) model. The output of the PINN serves as input to the oscillator. The PINN learns a solution within a convex training hull $X_1 = D \times T$, where $D \in \mathbb{R}^d$ is the d-dimensional spatial domain and $T \in \mathbb{R}$ is the temporal domain of the PDE. In our experiments, $d = 1$.

The neural oscillator processes the output of PINN as sequential data and predicts solutions within a different convex testing hull X_2. The hulls are distinct, X_1 and X_2, and $X_2 \not\subset X_1$. For example, $X_2 = D \times T^{'}$, where D is the same spatial domain but $T^{'} \in \mathbb{R}$ is the extrapolated temporal domain with $\inf(T^{'}) \geq \sup(T)$, which implies that testing is performed on time $t^{'} \in T^{'} \geq t \in T$.

The PINN maps the input space X_1 onto the solution space $\mathcal{U}$, such that a solution of the PDE $u \in \mathcal{U}$. This mapping enables learning the evolution of u from a given initial condition. The abstract formulation of an operator $\mathcal{N}$ incorporating the PDE and initial and boundary conditions is

$$\mathcal{N}(u) = f, \tag{4.1}$$

where f is the source term. The loss function of an abstract PINN is formed by minimising the residuals of (4.1) along with the available data on boundaries and at the initial time.

Following the PINN training on X_1, its testing is conducted on k_t uniform time steps in T and k_x uniform locations in D making a total of $k_t \cdot k_x$ testing points within X_1. The solution obtained from the PINN is reshaped to be further fed into the neural oscillator (shown in Fig. 4.1).

Conventional feed-forward neural networks lack explicit mechanisms to learn dependencies among outputs, presenting a fundamental *challenge in handling temporal relationships*. To mitigate this challenge, recurrent neural architectures preserve a hidden state to retain information from previous time steps, thereby improving sequence learning. We employ neural oscillators to treat the PINN outputs as a sequence. The motivation arises from feed-forward neural networks, where all outputs are independent, whereas sequence learning requires capturing temporal dependencies. Neural oscillators capture these dependencies through feedback loops and hidden states, enabling information propagation and temporal dependency capture.

While training an oscillator, its hidden states are updated using the current input and the previous hidden states, akin to vanilla RNNs. The fundamental distinction between vanilla or gated RNNs and neural oscillators lies in the hidden state update methodology. In neural oscillators, these updates are based on systems of ODEs, in contrast to algebraic equations used in typical RNNs. When employing CoRNN, the hidden states are updated through the second-order ODE

$$\mathbf{y}'' = \sigma\big(\mathbf{W}\mathbf{y} + \mathcal{W}\mathbf{y}' + \mathbf{V}\mathbf{u} + \mathbf{b}\big) - \gamma\mathbf{y} - \epsilon\mathbf{y}'. \tag{4.2}$$

Here, $\mathbf{y} = \mathbf{y}(t) \in \mathbb{R}^m$ is the hidden state of the RNN with weight matrices $\mathbf{W}, \mathcal{W} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{m\times k_x}$; t corresponds to the time levels at which the testing of PINNs has been performed; $\mathbf{u} = \mathbf{u}(t) \in \mathbb{R}^{k_x}$ is the PINN solution; $\mathbf{b} \in \mathbb{R}^m$ is the bias vector; and $\gamma, \epsilon > 0$ are the oscillatory parameters. We set the activation function $\sigma : \mathbb{R} \mapsto \mathbb{R}$ to $\sigma(u) = \tanh(u)$. Introducing $\mathbf{z} = \mathbf{y}'(t) \in \mathbb{R}^m$, we rewrite (4.2) as the first-order system

$$\mathbf{y}' = \mathbf{z}, \quad \mathbf{z}' = \sigma(\mathbf{W}\mathbf{y} + \mathcal{W}\mathbf{z} + \mathbf{V}\mathbf{u} + \mathbf{b}) - \gamma\mathbf{y} - \epsilon\mathbf{z}. \tag{4.3}$$

We use an explicit scheme with a time step $0 < \Delta t < 1$ to discretize these ODEs,

$$\begin{aligned} \mathbf{y}_n &= \mathbf{y}_{n-1} + \Delta t \mathbf{z}_n, \\ \mathbf{z}_n &= \mathbf{z}_{n-1} + \Delta t \sigma(\mathbf{W}\mathbf{y}_{n-1} + \mathcal{W}\mathbf{z}_{n-1} + \mathbf{V}\mathbf{u}_n + \mathbf{b}) \\ &\quad - \Delta t\gamma\mathbf{y}_{n-1} - \Delta t\epsilon\mathbf{z}_{\bar{n}}. \end{aligned} \tag{4.4}$$

Similarly, LEM updates the hidden states by solving the ODEs

$$\begin{aligned} \mathbf{y}' &= \hat{\sigma}(\mathbf{W}_2\mathbf{y} + \mathbf{V}_2\mathbf{u} + \mathbf{b}_2) \odot [\sigma(\mathbf{W}_y\mathbf{z} + \mathbf{V}_y\mathbf{u} + \mathbf{b}_y) - \mathbf{y}] \\ \mathbf{z}' &= \hat{\sigma}(\mathbf{W}_1\mathbf{y} + \mathbf{V}_1\mathbf{u} + \mathbf{b}_1) \odot [\sigma(\mathbf{W}_z\mathbf{y} + \mathbf{V}_z\mathbf{u} + \mathbf{b}_z) - \mathbf{z}] \end{aligned} \tag{4.5}$$

In addition to previously defined quantities, $\mathbf{W}_{1,2}, \mathbf{W}_{y,z} \in \mathbb{R}^{m\times m}$ and $\mathbf{V}_{1,2}, \mathbf{V}_{y,z} \in \mathbb{R}^{m\times k_x}$ are the weight matrices; $\mathbf{b}_{1,2}$ and $\mathbf{b}_{y,z} \in \mathbb{R}^m$ are the bias vectors; $\hat{\sigma}$ is the sigmoid activation function; and $\odot$ refers to the componentwise product of vectors. A discretization of (4.5) similar to CoRNN yields

$$\begin{aligned} \mathbf{\Delta t}_n &= \Delta t \hat{\sigma}(\mathbf{W}_1\mathbf{y}_{n-1} + \mathbf{V}_1\mathbf{u}_n + \mathbf{b}_1) \\ \overline{\mathbf{\Delta t}}_n &= \Delta t \hat{\sigma}(\mathbf{W}_2\mathbf{y}_{n-1} + \mathbf{V}_2\mathbf{u}_n + \mathbf{b}_2) \\ \mathbf{z}_n &= (1 - \mathbf{\Delta t}_n) \odot \mathbf{z}_{n-1} \\ &\quad + \mathbf{\Delta t}_n \odot \sigma(\mathbf{W}_z\mathbf{y}_{n-1} + \mathbf{V}_z\mathbf{u}_n + \mathbf{b}_z) \\ \mathbf{y}_n &= (1 - \overline{\mathbf{\Delta t}}_n) \odot \mathbf{y}_{n-1} \\ &\quad + \overline{\mathbf{\Delta t}}_n \odot \sigma(\mathbf{W}_y\mathbf{z}_n + \mathbf{V}_y\mathbf{u}_n + \mathbf{b}_y). \end{aligned} \tag{4.6}$$

Both CoRNN and LEM are augmented with a linear output state $\omega_n \in \mathbb{R}^{k_x}$ with $\omega_n = \mathcal{Q}\mathbf{y}_n$ and $\mathcal{Q} \in \mathbb{R}^{k_x\times m}$.

We train the PINN and the neural oscillator separately to leverage the *resolution-invariance* property of physics-informed learning during training. While neural oscillators require evenly spaced data, a PINN can be trained discretization-invariantly, allowing flexibility in handling multi-resolution data, such as using different sampling techniques [13]. The PINN is trained until a predefined epoch or until its validation error stabilizes in consecutive epochs and is then employed in inference to generate training data for the oscillator. Subsequently, the oscillator

Table 4.1: Generalization Accuracy of Nonlinear Benchmark PDEs

	L2-norm			Variance score			Max Error			Mean Error		
	DPM	CoRNN	LEM	DPM	CoRNN	LEM	DPM	CoRNN	LEM	DPM	CoRNN	LEM
Vis. Burgers	0.083	0.004	**1e-3**	0.621	0.995	**0.999**	1.534	0.103	**0.024**	0.277	0.022	**0.003**
Allen–Cahn	0.182	0.005	**0.004**	0.967	0.995	**0.995**	0.836	0.320	**0.137**	0.094	0.035	**0.034**
Schrödinger	0.141	0.042	**0.003**	-3.257	0.925	**0.994**	3.829	0.659	**0.094**	0.868	0.925	**0.028**

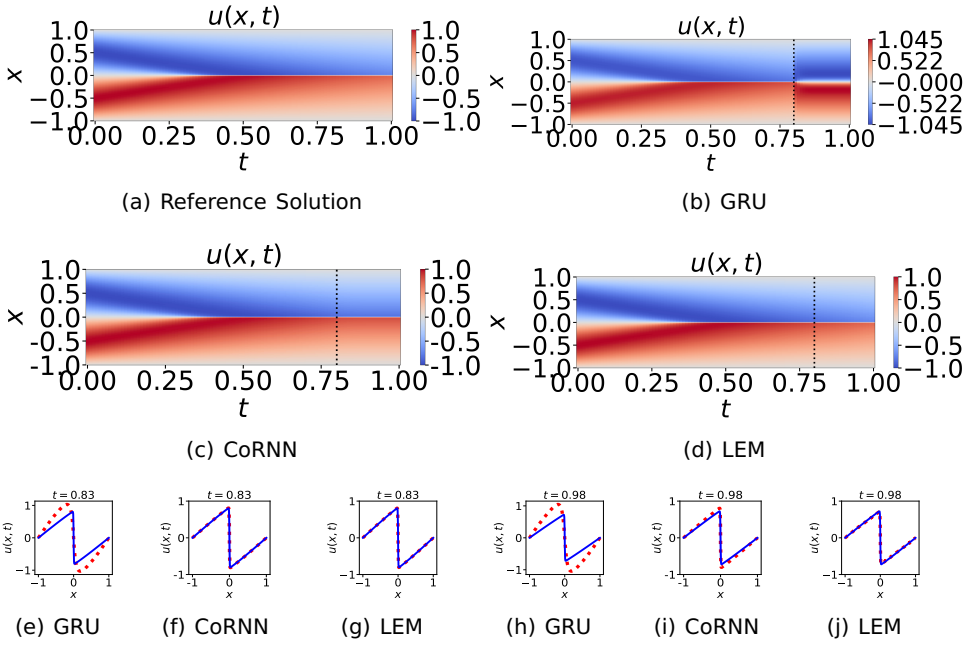


Figure 4.2: Top two rows: the complete reference solution and predictions for viscous Burgers equation. The black vertical line delineates the region before which the PINN has been trained. The region after the black vertical line represents the generalization domain. The meaning of the vertical line remains the same in the following figures. Bottom: the solution snapshots at $t = \{0.83, 0.98\}$ obtained in the generalization region, where blue represents the reference solution, and red refers to the recurrent method. The colors are used consistently for the following figures.

learns a mapping between the PINN outputs from one-time level to the next, forming a sequential relationship.

We validate the proposed framework on three time-dependent nonlinear PDEs and a fourth-order biharmonic beam equation. The software and hardware environments used to perform the experiments

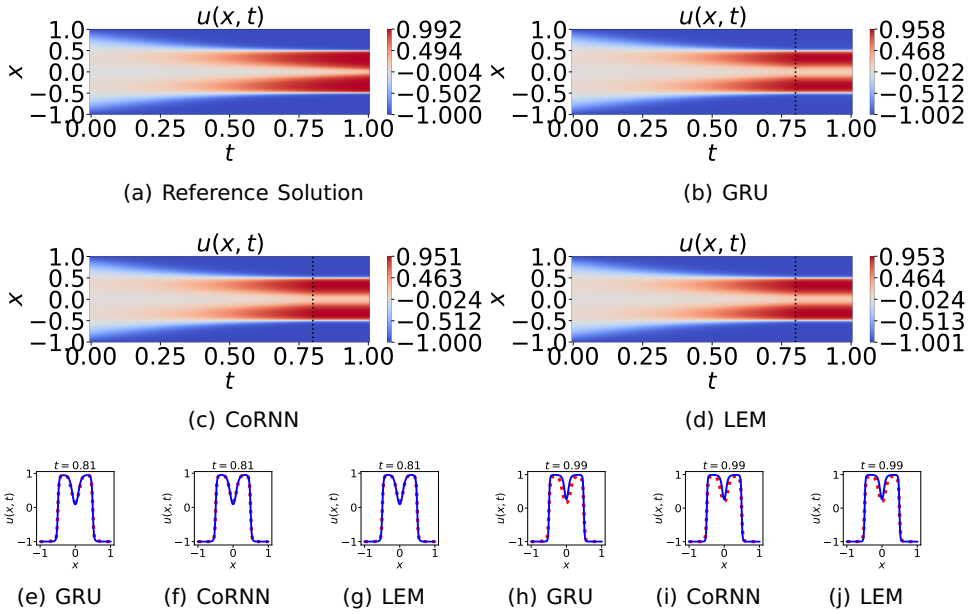
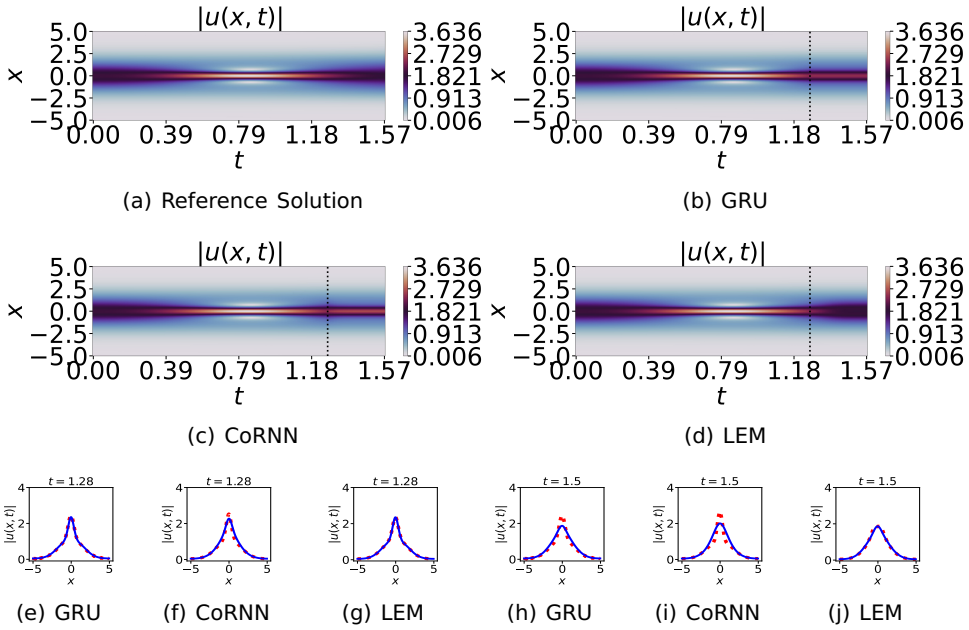


Figure 4.3: Top two rows: the complete reference solution and predictions for the Allen-Cahn equation. Bottom: the solution snapshots at $t = \{0.81, 0.99\}$ obtained in the generalization region.

are as follows: Ubuntu 20.04.6 LTS, Python 3.9.7, Numpy 1.20.3, Scipy 1.7.1, Matplotlib 3.4.3, TensorFlow-gpu 2.9.1, PyTorch 1.12.1, CUDA 11.7, and NVIDIA Driver 515.105.01, i7 CPU, and NVIDIA GeForce RTX 3080.

PDES

The four equations—viscous Burgers equation, Allen-Cahn (AC) equation, nonlinear Schrödinger equation (NLS) and Euler-Bernoulli beam equation—along with their boundary and initial conditions are provided in the supplementary material **SM**§B. For training/testing, we divide the entire time domain into two segments: $T := [0, T_{\text{train}}]$ and $T^{'} := (T_{\text{train}}, T_{\text{test}}]$, where $T_{\text{test}} > T_{\text{train}} > 0$. Our task is to predict the PDE solution in the convex testing hull $X_2 = D \times T^{'}$ after the model has been trained on the convex training hull $X_1 = D \times T$. For all the problems, $T_{\text{train}} = 0.8T_{\text{test}}$, dividing the training and test sets in the ratio $4:1$, following the work of DPM [12] to maintain uniformity. The domain for each PDE, i.e., D, T and T', is defined in **SM**§B.



(a) Reference Solution (b) GRU

(c) CoRNN (d) LEM

(e) GRU (f) CoRNN (g) LEM (h) GRU (i) CoRNN (j) LEM

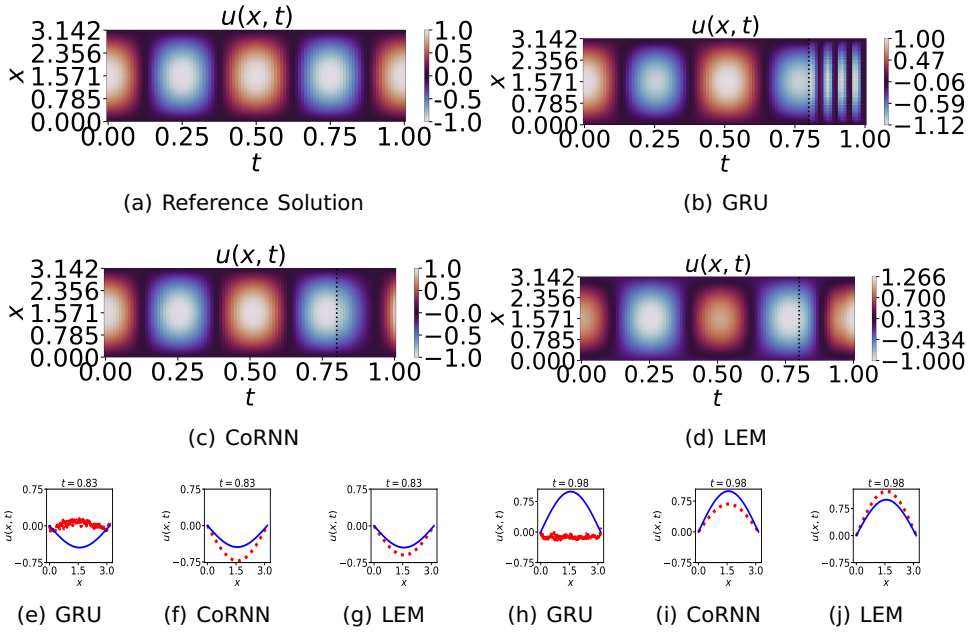
Figure 4.4: Top two rows: the complete reference solution and predictions for the Schrödinger equation. Bottom: the solution snapshots at $t = \{1.28, 1.5\}$ obtained in the generalization region.

BASELINES

Our objective is to make predictions beyond X_1, i.e., on X_2, and to assess how well the trained models generalize. We compare the performance of PINNs with CoRNN or LEM on this task. We also compare our approach to the state-of-the-art DPM [12]. A comparative analysis is also carried out when traditional recurrent networks, RNN, LSTM, and GRU, are augmented with the physics-informed model instead of the oscillatory networks. This analysis provides insight into how well the oscillatory methods perform relative to traditional recurrent networks and gradient techniques when confronted with generalization tasks.

HYPERPARAMETERS

To predict a solution to Burgers equation in X_1 using PINNs, 1600 training points are used, comprising 1000 residual points and 600 points for boundary and initial time. The feedforward neural network has two inputs, space $x \in D$ and time $t \in T$. Four hidden layers, each containing 20 neurons, and hyperbolic tangent (tanh) activation function are used to predict the approximation of the solution $u \in \mathcal{U}$. Optimization is performed using the LBFGS algorithm for 3500 epochs. For the



(a) Reference Solution (b) GRU

(c) CoRNN (d) LEM

(e) GRU (f) CoRNN (g) LEM (h) GRU (i) CoRNN (j) LEM

Figure 4.5: Top two rows: the complete reference solution and predictions for the Euler–Bernoulli beam equation. Bottom: the solution snapshots at $t = \{0.83, 0.98\}$ obtained in the generalization region.

Euler-Bernoulli beam equation, 16000 training points are distributed as 10000 residual points and 6000 points designated for both initial and boundaries. The hyperparameters are kept the same as in the viscous Burgers equation. Allen-Cahn and Schrödinger equations are simulated using the software DeepXDE [44] with the default hyperparameters described therein.

The input and output size of the recurrent networks is taken to be k_x, with a single hidden layer of size 32. The sequence length is chosen to be k_t. The exact values of k_x and k_t are defined in the “Train and test criteria" subsection for each equation. Adam optimizer is used to train the recurrent networks. The learning rates for LEM, CoRNN, GRU, LSTM, and RNN are 0.001, 0.001, 0.01, 0.01, and 0.01, respectively, across all equations. For Schrödinger equation, a learning rate of 0.01 is used to train the LEM. In the case of CoRNN, two additional hyperparameters, γ and ϵ, are set to 1.0 and 0.01, respectively. The number of epochs executed for Burgers and Allen–Cahn equations is $20,000$, while for Schrödinger equation, it is $30,000$. Lastly, $200,000$ epochs are performed for the Euler-Bernoulli beam equation.

EVALUATION METRICS

For the first three experiments, the errors are reported relative to the numerical solutions of the corresponding PDEs. The reference for the Euler-Bernoulli beam equation is an analytical solution described in **SM**§B. As the criteria for assessment, we employ standard evaluation metrics: the relative errors in the L2-norm, the explained variance score, the maximum error, and the mean absolute error, defined in **SM**§D. Each of these metrics provides distinct insights into the performance. Furthermore, we present visual snapshots of both the reference and approximate solutions at specific time instances. Additional snapshots and contour results are provided in **SM**§C.

TRAIN AND TEST CRITERIA

The trained PINN is tested on $k_t \cdot k_x$ points in X_1. For the Burgers equation and the Euler-Bernoulli beam equations, we set $k_x = 256$ and $k_t = 80$. For the Allen-Cahn equation, $k_x = 201$ and $k_t = 80$. For the Schrödinger equation, $k_x = 256$ and $k_t = 160$.

The PINN output provides input to train the neural oscillators, adhering to the specified hyperparameter configuration. After training the neural oscillator on X_1, testing is extended to X_2. This testing sequence commences at $\inf(T')$ as the initial input. The ensuing output is then utilized as the input for the subsequent sequence (shown in Fig. 4.1). Such testing is crucial since, in practical scenarios, knowledge about the solution u in X_2 is absent. Thus, the solely available information for generalization is derived from the predicted solution within X_2. This testing process is iterated until reaching $\sup(T')$. The domains X_1, X_2 and T' for all the equations are provided in **SM**§B.

4.4. EXPERIMENTAL RESULTS

Tables 4.1 and 4.2 collate the overall performance metrics for the oscillator-based methods (LEM, CoRNN) in comparison with DPM, RNN, LSTM and GRU. The results show that LEM exhibits significantly superior performance across all the benchmark problems.

VISCOUS BURGERS EQUATION

Figure 4.2 provides a visual comparison between the reference solution (Fig. 4.2(a)) and its counterparts generated with GRU, CoRNN and LEM (Figs. 4.2(b)–4.2(d), respectively). GRU struggles to accurately capture the solution of Burgers equation, leading to the loss in prediction accuracy as time t increases. Our methods based on CoRNN and LEM exhibit notably improved predictive accuracy, even when t approaches the end of the time domain. Figures 4.2(e)–4.2(j) provide further insights

into the solution at time instances $t = 0.83, 0.98$. They reveal that LEM outperforms the alternative methods across the entire space-time domain. The performance of CoRNN is comparable to that of LEM, producing reasonably accurate predictions. These findings underscore the significance of neural oscillators in precise generalization. Additional experiments on *sensitivity analysis* of oscillator parameter (Δt) along with an *ablation study* on CoRNN parameters ϵ and γ is presented in **SM**§C. Additionally, the generalization in parametric space [45] is also presented in **SM**§C.

ALLEN-CAHN EQUATION

In Figure 4.3, the reference solution of the Allen-Cahn equation is compared to its counterparts generated with GRU, CoRNN and LEM. Our oscillator-based methods (CoRNN and LEM) yield the most precise approximations in the generalization domain (Figs. 4.3(a)–4.3(d)). The LEM-based solution exhibits a nearly symmetric behavior with respect to $x = 0$, demonstrating its ability to preserve the symmetry and structure of the solution. At $t = 0.81$, all three methods display a similar level of accuracy (Figs. 4.3(e)–4.3(g)). However, as time advances, e.g., at $t = 0.99$, the performance of LEM surpasses that of the other techniques throughout the extrapolation domain (Figs. 4.3(h)–4.3(j)).

SCHRÖDINGER EQUATION

Figure 4.4 illustrates a comparison between the reference solution of Schrödinger equation and its counterparts generated with GRU, CoRNN and LEM. Rather than plotting the real and imaginary parts of this solution, Figs. 4.4(a)–4.4(d) exhibit its magnitude, $|u(x, t)|$; the solutions are visually indistinguishable. The three approximations are accurate at time $t = 1.28$ (Figs. 4.4(e)–4.4(g)), but the GRU- and CoRNN-based solutions at $t = 1.5$ have errors around $x = 0$ whereas the LEM-based solution retains its accuracy within that region (Figs. 4.4(f)–4.4(j)).

EULER-BERNOULLI BEAM EQUATION

In Figure 4.5, we compare the analytical solution of the Euler-Bernoulli beam equation to approximate solutions obtained with GRU, CoRNN and LEM. The intricacy of this linear equation stems from the presence of fourth-order derivatives [46, 47], rendering it a compelling challenge for the proposed methodology . The visual comparison afforded by Figs. 4.5(a)–4.5(d) demonstrates the superiority of the LEM-based solution and the inferiority of the GRU-based one. At $t = 0.83$, all three approximations are qualitatively correct, with various degrees of accuracy (Figs. 4.5(e)–4.5(h)). At $t = 0.98$, the GRU-based solution is not only inaccurate but is also qualitatively incorrect, while the

oscillator-based approximators correctly predict the behavior of the system (Figs. 4.5(h)–4.5(i)).

4.5. CONCLUSIONS

We introduced a method that combines neural oscillators with physics-informed neural networks to enhance performance in unexplored regions. This novel approach enables the model to learn the long-time dynamics of solutions to the governing partial differential equations. We demonstrated the effectiveness of our method on three benchmark nonlinear PDEs: viscous Burgers, Allen-Cahn, and Schrödinger equations, as well as the biharmonic Euler-Bernoulli beam equation. Our results showcase the improved generalization performance of the PIML augmented with neural oscillators, which outperforms state-of-the-art methods in various metrics. The codes to reproduce the presented results are provided at https://github.com/taniyakapoor/AAAI24_Generalization_PIML.

Table 4.2: The generalization accuracy in terms of the relative errors in the L2-norm, the explained variance error, the max error, and the mean absolute error for various PDEs. Higher (or lower) values are preferred, corresponding to ↑ (or ↓).

	L2-norm (↓)				**Variance Score (↑)**				**Max Error (↓)**				**Mean Error (↓)**			
	RNN	**LSTM**	**GRU**	**LEM**	**RNN**	**LSTM**	**GRU**	**LEM**	**RNN**	**LSTM**	**GRU**	**LEM**	**RNN**	**LSTM**	**GRU**	**LEM**
Vis. Burgers	0.4154	0.4635	0.3768	**0.0001**	0.584	0.536	0.623	**0.999**	0.594	1.040	0.544	**0.024**	0.266	0.285	0.253	**0.003**
Allen–Cahn	0.0058	0.0570	0.0093	**0.0049**	0.995	0.946	0.991	**0.995**	0.145	0.399	0.176	**0.137**	0.040	0.194	0.050	**0.034**
Schrödinger	0.3170	0.5022	0.0218	**0.0034**	0.440	0.272	0.961	**0.994**	1.695	0.153	0.434	**0.094**	0.260	0.326	0.075	**0.028**
Euler–Bernoulli	4.6509	2.1198	2.9176	**0.0593**	-0.844	-0.258	-0.566	**0.940**	1.997	1.465	2.044	**0.267**	0.797	0.500	0.604	**0.091**

REFERENCES

[1] R. Balestriero, J. Pesenti, and Y. LeCun. “Learning in high dimension always amounts to extrapolation”. In: *arXiv preprint arXiv:2110.09485* (2021). doi: 10.48550/arXiv.2110.09485.

[2] Z. Su, K. Yao, X. Yang, K. Huang, Q. Wang, and J. Sun. “Rethinking data augmentation for single-source domain generalization in medical image segmentation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 2023, pp. 2366–2374. doi: 10.48550/arXiv.2211.14805.

[3] C. Chen, Y. Hu, Q. Zhang, H. Zou, B. Zhu, and E. S. Chng. “Leveraging modality-specific representations for audio-visual speech recognition via reinforcement learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 2023, pp. 12607–12615. doi: 10.48550/arXiv.2212.05301.

[4] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy. “Domain generalization: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022). doi: 10.1109/TPAMI.2022.3195549.

[5] H. Liu, T. Zhang, N. Anoop Krishnan, M. M. Smedskjaer, J. V. Ryan, S. Gin, and M. Bauchy. “Predicting the dissolution kinetics of silicate glasses by topology-informed machine learning”. In: *Npj Materials Degradation* 3.1 (2019), p. 32. doi: 10.1038/s41529-019-0094-1.

[6] M. Alber, A. Buganza Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, *et al.* “Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences”. In: *Npj Digital Medicine* 2.1 (2019), p. 115. doi: 10.1038/s41746-019-0193-y.

[7] J. Gu, Z. Gao, C. Feng, H. Zhu, R. Chen, D. Boning, and D. Pan. “NeurOLight: A Physics-Agnostic Neural Operator Enabling Parametric Photonic Device Simulation”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 14623–14636.

[8] K. Lee, N. Trask, and P. Stinis. “Machine learning structure preserving brackets for forecasting irreversible processes”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 5696–5707.

[9] M. Raissi, P. Perdikaris, and G. E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational physics* 378 (2019), pp. 686–707. doi: 10.1016/j.jcp.2018.10.045.

[10] S. Mishra and R. Molinaro. “Estimates on the generalization error of physics-informed neural networks for approximating PDEs”. In: *IMA Journal of Numerical Analysis* 43.1 (2023), pp. 1–43. doi: 10.48550/arXiv.2006.16144.

[11] S. Mishra and R. Molinaro. “Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs”. In: *IMA Journal of Numerical Analysis* 42.2 (2022), pp. 981–1022. doi: 10.1093/imanum/drab032.

[12] J. Kim, K. Lee, D. Lee, S. Y. Jhin, and N. Park. “DPM: A novel training method for physics-informed neural networks in extrapolation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 8146–8154. doi: 10.48550/arXiv.2012.02681.

[13] A. Daw, J. Bu, S. Wang, P. Perdikaris, and A. Karpatne. “Rethinking the importance of sampling in physics-informed neural networks”. In: *arXiv preprint arXiv:2207.02338* (2022). doi: 10.48550/arXiv.2207.02338.

[14] L. Fesser, R. Qiu, and L. D’Amico-Wong. “Understanding and Mitigating Extrapolation Failures in Physics-Informed Neural Networks”. In: *arXiv preprint arXiv:2306.09478* (2023). doi: 10.48550/arXiv.2306.09478.

[15] O. Fuks and H. A. Tchelepi. “Limitations of physics informed machine learning for nonlinear two-phase transport in porous media”. In: *Journal of Machine Learning for Modeling and Computing* 1.1 (2020). doi: 10.1615/JMachLearnModelComput.2020033905.

[16] Y. Shin, J. Darbon, and G. E. Karniadakis. “On the convergence and generalization of physics informed neural networks”. In: *arXiv e-prints* (2020), arXiv–2004. doi: 10.48550/arXiv.2004.01806.

[17] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu. “Generalizing to unseen domains: A survey on domain generalization”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022). doi: 10.48550/arXiv.2103.03097.

[18] S. Wang, S. Sankaran, and P. Perdikaris. “Respecting causality is all you need for training physics-informed neural networks”. In: *arXiv preprint arXiv:2203.07404* (2022). doi: 10.48550/arXiv.2203.07404.

[19] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014). doi: 10.48550/arXiv.1406.1078.

[21] R. Pascanu, T. Mikolov, and Y. Bengio. “On the difficulty of training recurrent neural networks”. In: *International Conference on Machine Learning*. Pmlr. 2013, pp. 1310–1318. doi: 10.48550/arXiv.1211.5063.

[22] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao. “Independently recurrent neural network (indrnn): Building a longer and deeper rnn”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5457–5466. doi: 10.48550/arXiv.1803.04831.

[23] M. Henaff, A. Szlam, and Y. LeCun. “Recurrent orthogonal networks and long-memory tasks”. In: *International Conference on Machine Learning*. PMLR. 2016, pp. 2034–2042. doi: 10.48550/arXiv.1602.06662.

[24] M. Arjovsky, A. Shah, and Y. Bengio. “Unitary evolution recurrent neural networks”. In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1120–1128. doi: 10.48550/arXiv.1511.06464.

[25] S. Wisdom, T. Powers, J. Hershey, J. Le Roux, and L. Atlas. “Full-capacity unitary recurrent neural networks”. In: *Advances in Neural Information Processing Systems* 29 (2016). doi: 10.48550/arXiv.1611.00035.

[26] G. Kerg, K. Goyette, M. Puelma Touzel, G. Gidel, E. Vorontsov, Y. Bengio, and G. Lajoie. “Non-normal recurrent neural network (nnrnn): learning long time dependencies while improving expressivity with transient dynamics”. In: *Advances in Neural Information Processing Systems* 32 (2019). doi: 10.48550/arXiv.1905.12080.

[27] S. Lanthaler, T. K. Rusch, and S. Mishra. “Neural Oscillators are Universal”. In: *arXiv preprint arXiv:2305.08753* (2023). doi: 10.48550/arXiv.2305.08753.

[28] T. K. Rusch and S. Mishra. “Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies”. In: *arXiv preprint arXiv:2010.00951* (2020). doi: 10.48550/arXiv.2010.00951.

[29] T. K. Rusch, S. Mishra, N. B. Erichson, and M. W. Mahoney. “Long expressive memory for sequence modeling”. In: *arXiv preprint arXiv:2110.04744* (2021). doi: 10.48550/arXiv.2110.04744.

[30] T. Kapoor, A. Chandra, D. M. Tartakovsky, H. Wang, A. Nunez, and R. Dollevoet. “Neural oscillators for generalization of physics-informed machine learning”. In: *arXiv preprint arXiv:2308.08989* (2023). doi: 10.48550/arXiv.2308.08989.

[31] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440. doi: 10.1038/s42254-021-00314-5.

[32] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. “Scientific machine learning through physics–informed neural networks: where we are and what’s next”. In: *Journal of Scientific Computing* 92.3 (2022), p. 88. doi: 10.1007/s10915-022-01939-z.

[33] A. Daw, A. Karpatne, W. D. Watkins, J. S. Read, and V. Kumar. “Physics-guided neural networks (pgnn): An application in lake temperature modeling”. In: *Knowledge Guided Machine Learning*. Chapman and Hall/CRC, 2022, pp. 353–372. doi: 10.48550/arXiv.1710.11431.

[34] M. Raissi, A. Yazdani, and G. E. Karniadakis. “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations”. In: *Science* 367.6481 (2020), pp. 1026–1030. doi: 10.1126/science.aaw4741.

[35] E. Zhang, M. Dao, G. E. Karniadakis, and S. Suresh. “Analyses of internal structures and defects in materials using physics-informed neural networks”. In: *Science advances* 8.7 (2022), eabk0644. doi: 10.1126/sciadv.abk0644.

[36] L. D. McClenny and U. M. Braga-Neto. “Self-adaptive physics-informed neural networks”. In: *Journal of Computational Physics* 474 (2023), p. 111722. doi: 10.1016/j.jcp.2022.111722.

[37] J. Shen, Y. Qu, W. Zhang, and Y. Yu. “Wasserstein distance guided representation learning for domain adaptation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 2018.

[38] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. “Characterizing possible failure modes in physics-informed neural networks”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[39] A. Dekhovich, M. H. Sluiter, D. M. Tax, and M. A. Bessa. “iPINNs: Incremental learning for Physics-informed neural networks”. In: *arXiv preprint arXiv:2304.04854* (2023). doi: 10.48550/arXiv.2304.04854.

[40] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. “Neural ordinary differential equations”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[41] Y. Rubanova, R. T. Chen, and D. K. Duvenaud. “Latent ordinary differential equations for irregularly-sampled time series”. In: *Advances in Neural Information Processing Systems* 32 (2019). doi: 10.48550/arXiv.1907.03907.

[42] B. Chang, M. Chen, E. Haber, and E. H. Chi. “AntisymmetricRNN: A dynamical system view on recurrent neural networks”. In: *arXiv preprint arXiv:1902.09689* (2019). doi: 10.48550/arXiv.1902.09689.

[43] T. K. Rusch and S. Mishra. “Unicornn: A recurrent model for learning very long time dependencies”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9168–9178. doi: 10.48550/arXiv.2103.05487.

[44] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis. “DeepXDE: A deep learning library for solving differential equations”. In: *SIAM Review* 63.1 (2021), pp. 208–228. doi: 10.48550/arXiv.1907.04502.

[45] T. Kapoor, A. Chandra, D. M. Tartakovsky, H. Wang, A. Nunez, and R. Dollevoet. “Neural oscillators for generalization of physics-informed machine learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 12. 2024, pp. 13059–13067. doi: 10.48550/arXiv.2308.08989.

[46] T. Kapoor, H. Wang, A. Nunez, and R. Dollevoet. “Physics-informed neural networks for solving forward and inverse problems in complex beam systems”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023). doi: 10.1109/TNNLS.2023.3310585.

[47] Q. Cao, S. Goswami, and G. E. Karniadakis. “LNO: Laplace neural operator for solving differential equations”. In: *arXiv preprint arXiv:2303.10528* (2023). doi: 10.1038/s42256-024-00844-4.

5

EXTRAPOLATION OF DYNAMIC SOLVERS FOR STRUCTURES

Computer-aided simulations are routinely used to predict the performance of a prototype. High-fidelity physics-based simulators might be computationally expensive for design and optimization, spurring the development of cheap deep-learning surrogates. The resulting surrogates often struggle to generalize and predict novel scenarios beyond their training domain. We propose a two-stage methodology addressing the challenge of generalization. It employs physics-based simulators, supplemented with ordinary differential equations integrated into the recurrent architecture, to learn the intrinsic dynamics. The proposed approach captures the inherent causality and generalizes the dynamics irrespective of a data source. The presented numerical experiments encompass four fundamental structural engineering scenarios, including beams on Winkler foundations based on Euler-Bernoulli and Timoshenko theories, beams under moving loads, and catenary-pantograph interactions in railways. The proposed methodology outperforms conventional recurrent methods and remains invariant to data sources, showcasing its efficacy. Numerical experiments highlight its prospects for design optimization, predictive maintenance, and enhancing safety measures.

Apart from minor updates, this chapter has been submitted for publication as: Kapoor, T., Wang, H., Stamou, A., Sayed. K., Nunez, A., Tartakovsky, D. M., & Dollevoet, R. Generalizing beam dynamics from simulator data with neural differential equations.

5.1. INTRODUCTION

Computer-aided dynamic simulations are pivotal across engineering industries [1], including structural, railway, automotive, manufacturing, and aerospace, among others. These simulations offer a cost-effective and efficient alternative to physical prototyping, saving time and resources. Simulating complex engineering systems and subsystems aids in comprehensive testing and validation and assists in analyzing and optimizing their performance and designs [2].

Several methods, including numerical solvers and deep learning, are utilized to simulate the dynamics of engineering structures. For instance, the finite element method (FEM) serves as a backbone for commercial engineering software and is widely used for modeling, designing, and optimizing structural dynamics [3]. Furthermore, deep learning-based methods are employed as surrogates to simulate the underlying dynamics [4]. One such deep learning-based approach is physics-informed machine learning, integrating physical principles into neural network architecture [5]. These simulators collectively provide robust tools for realistic and efficient simulations of engineering systems.

Industrial scenarios such as design optimization necessitate numerous simulations with varying materials and conditions influencing the dynamics. Numerical solvers require repetitive iterations for each parameter change, increasing computational cost [6]. Each iteration is particularly challenging for long-time integration problems, where the long-term behavior of the engineering systems needs to be analyzed. Additionally, as the number of degrees of freedom increases, the number of simulations escalates exponentially, rendering the process laborious and time-consuming. Similarly, in deep learning, predictions outside the training domain are challenging for engineering simulations [7], requiring multiple model training for varying parameter values. Thus, long-time predictions and incorporating parameter variations exacerbate the simulation challenge across numerical and deep learning methodologies.

Current engineering dynamic simulators are matured, yet predictions outside the training domain, termed extrapolation or generalization, remain challenging [8]. An ideal simulator should provide accurate predictions inside the training domain and reasonable accuracy for generalization scenarios. Mitigating the problem of generalization necessitates a method that can predict out-of-domain and complement the current dynamic solvers. A potential approach addressing this challenge involves leveraging the current simulation strategies and coupling them with a method to provide reliable generalization predictions, reducing the problem to a two-stage strategy.

The first stage involves the traditional solver simulating the problem precisely in the training domain. The second stage utilizes the predictions from the first stage to generalize beyond the training

domain. For example, for a moving load problem like a train on the railway track, the quantity of interest is track deflection under different loadings. For certain loadings, deflections could be simulated using a traditional solver, such as FEM, and deflections for unseen loadings could be predicted in a second stage utilizing the FEM deflections.

For the second stage, prior knowledge available is the simulated data in the training domain, and utilizing it to predict in the generalized domain is non-trivial. A potential way is to leverage the causality of the underlying dynamics from the trained domain and use them in an auto-regressive approach to predict the quantities of interest in the untrained domain. Learning the inherent causality would allow reasonable generalization by capturing the underlying dynamics, structure, and symmetry of the problem.

Attention-based and recurrent neural architectures are viable for sequence modeling and prediction tasks. However, attention-based models require substantial data and disregard the underlying sequential causality inherent in the physical simulations. Recurrent neural networks (RNNs) are well-known to model time-series data. However, long sequences pose exploding and vanishing gradients problem (EVGP), which can even be observed with advanced gated architectures [9] like the long short-term memory (LSTM) [10] and gated recurrent unit (GRU) [11]. The challenge of EVGP could be mitigated by employing ordinary differential equations (ODEs) to update the hidden states of the recurrent architecture, facilitating efficient dynamic learning. For instance, recent methods like coupled-oscillatory recurrent neural network (CoRNN) [12] and long expressive memory (LEM) [13] do not exhibit the EVGP and perform well for several sequential artificial intelligence (AI) tasks. This work proposes to merge dynamic simulations with recurrent architectures employing ODEs to update the hidden states, namely, CoRNN and LEM. The proposed two-stage methodology exploits the causality and learns temporal and parametric dependencies, potentially enhancing the accuracy of generalization predictions of engineering dynamic simulations.

Five fundamental problems within structural engineering are examined to validate the proposed methodology. Simulation of beam dynamics under dynamic loading conditions is crucial for precise structural analysis and design [14]. Beams on Winkler foundations are crucial in civil engineering, providing stability and support by distributing loads, minimizing buckling [15] and resisting against vibrations [16]. Concrete beams are used extensively in the construction industry [17]. They are commonly used in applications like railway tracks, pile foundations, and composite elastomers, where understanding their behavior is essential for maintaining structural integrity and optimizing designs.

The first two cases are based on well-known Euler-Bernoulli and Timoshenko beam theories on the Winkler foundation. Both beam

theories are fundamental in structural mechanics. Euler-Bernoulli beam theory models the behavior of slender, linear beams under various loading conditions, assuming the plane sections of the beam to remain plane and perpendicular to the cross-section where no longitudinal stresses or strains occur. The Euler-Bernoulli theory also assumes the shear deformations and rotational effects to be negligible [18]. This simplification allows the calculation of stresses and deflections in beams under static and dynamic loads by reducing the three-dimensional problem of beam bending to a one-dimensional model [19]. The governing equation is derived from equilibrium conditions (using Newton's laws or Lagrangian mechanics), material constitutive laws, and geometric properties, leading to a fourth-order differential equation.

The Timoshenko beam theory builds on the Euler-Bernoulli theory by including the effects of shear force. Unlike the Euler-Bernoulli theory, which assumes that beam cross-sections remain flat and perpendicular to the neutral axis after deformation, Timoshenko theory accounts for additional angular rotation due to shear strain [20], affecting the displacement by shear and bending deformations. The theory introduces an extra degree of freedom, represented by the angular rotation. Therefore, the key quantities of interest are the displacement and the angular rotation. These quantities of interest are computed by solving the governing equations, consisting of two coupled second-order partial differential equations.

The third case is the moving load problem, studying the deflection of the beam under different loadings. Understanding beam behavior under moving loads is essential for structural health monitoring and maintaining infrastructure integrity. For the final experiment, real-world catenary-pantograph interactions in railway systems are studied. Comprehending the vertical displacement of the catenary contact wire under various train speeds is critical for railway infrastructure safety and design. Varying speeds induce dynamic loads affecting the stability of the catenary. Accurate contact wire uplift predictions aid engineers in estimating the bad state of the catenary, which directly influences the power supply safety of the traction power system, averting potential disruptions and accidents. The final experiment deals with estimating the unknown force applied on a system of Timoshenko beams by solving an inverse problem.

The main contributions of this work can be summarized as follows,

- This work introduces a two-stage approach for generalizing engineering dynamic simulations. First, dynamics are simulated using a state-of-the-art simulator preferred for the application, followed by classical mathematical models (ODEs) infused in neural architecture to generalize the dynamics.

- Generalizing simulations is a longstanding challenge in AI for

engineering and industry. The proposed work tackles this issue through a resolution invariant pipeline where both stages can process data at different resolutions.

- The proposed workflow efficiently generalizes the dynamics in the time domain for spatio-temporal engineering systems without using any data from the untrained time domain.
- Furthermore, for spatio-temporal parametric systems, the workflow eliminates tedious re-meshing and re-simulation in computer-aided simulation software for novel parameters belonging to the parameter space.
- The performance of the proposed approach is evaluated on four different dynamic simulation problems in structural engineering, including real-world catenary-pantograph systems, demonstrating superior performance compared to traditional recurrent architectures.

The rest of the chapter is structured as follows. Section II details the related works to this chapter. Section III provides an overview of the problem statement. Section IV presents the proposed two-stage methodology in detail to enhance the generalization. Section V presents the performed numerical experiments to validate the proposed methodology. The main conclusions drawn from this study are collated in section VI.

5.2. RELATED WORK

This work focuses on generalizing spatio-temporal parametric engineering dynamics in temporal and parametric space. Industrial applications often necessitate simulating spatio-temporal parametric systems as discussed in the works of [21–24], among others. In particular, [22, 24] discuss the challenges and importance of spatio-temporal simulations in industrial settings. In addition to sharing the core idea of simulating spatio-temporal parametric dynamics, this work proposes a method to generalize the dynamics in untrained domains, making it more applicable to real-world situations where systems may need to be deployed in unseen situations for which data collection may not even be possible at priori [25].

In other works, generalization of deep learning-based methods for industrial tasks has been explored in [25–27], among others, aligning with the motivation of this work. However, this work proposes a generic two-step approach enabling the investigation through the state-of-the-art simulator and coupling it with ODE-based recurrent architecture to capture the intrinsic dynamics. This approach democratizes the

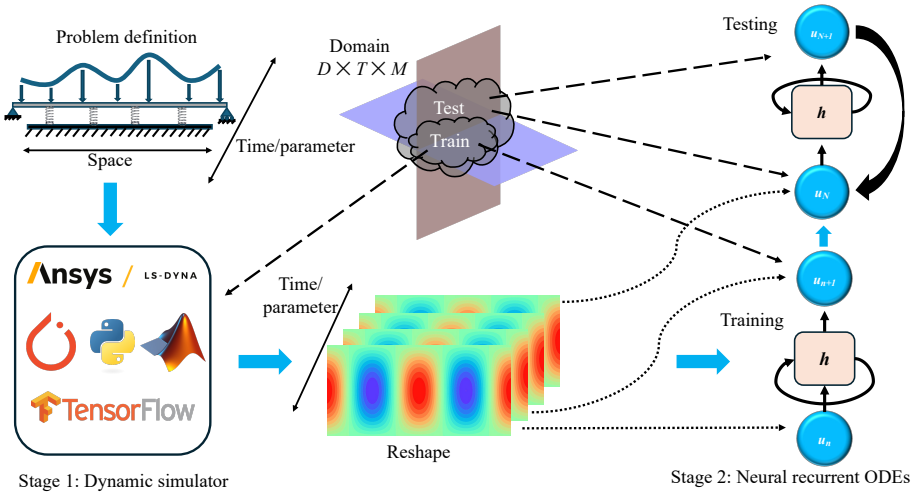


Figure 5.1: Schematic of the proposed methodology: two-stage methodology starts with a definition of the problem and training and testing domain in which generalization is sought. A dynamic simulator is then utilized in the first stage to simulate the problem in the training domain, whose solutions are reshaped to fed into the second stage comprising of neural recurrent ODEs. Neural ODEs are trained on the simulator predictions. The trained neural ODEs are used to predict for novel time and parameters in the testing domain.

challenge of generalization and hence could be used by a larger industrial and engineering community.

The second stage in the proposed methodology encompasses employing ODEs in the recurrent neural architecture to enhance generalization. Pioneered by the seminal works of [28], neural ODEs have been explored for different applications, for instance, cooling-system prediction [29], process quality evaluation [30], and remaining useful life estimation [31]. However, this work posits that neural differential equations-based architectures capture the causality better than the gated architectures, improving the generalization ability of the learned model and making it distinct from the aforementioned works.

5.3. PROBLEM STATEMENT

This section presents an abstract formulation of the generalization problem. In general, spatio-temporal parametric systems are governed by an operator $\mathcal{D}[u(x,t;\mu)] = 0$, where, $(x,t) \in D \times T$. Here, $D \subset \mathbb{R}$

and $T \subset \mathbb{R}$ represent the spatial and temporal domain, respectively. Additionally, $\mu \in M \subset \mathbb{R}$ represents the parameters, and $u \in \mathcal{U} \subset \mathbb{R}^d$ is the quantity of interest in a d-dimensional space. The operator $\mathcal{D}$ could be explicitly known, for instance, the Euler-Bernoulli and Timoshenko partial differential equations (PDEs) governing the beam dynamics, or could be black-box as in the case of real-world catenary-pantograph system.

The proposed approach transcends traditional limitations and seamlessly applies across temporal and parametric domains. Hence, to formalize the generalization problem, the domain Ω hereafter invariantly represents the domain T or M. The entire spatio-temporal parametric space is divided into two disjoint sets $\mathcal{X}_1$ and $\mathcal{X}_2$, where $\mathcal{X}_1 := D \times \Omega$ and $\mathcal{X}_2 := D \times \Omega'$. Here $\Omega^{'}$ is the generalized temporal or parametric domain with $\inf(\Omega^{'}) \geq \sup(\Omega)$, which implies that testing is performed for $\omega^{'} \in \Omega^{'} \geq \omega \in \Omega$. Concretely, the temporal or parametric space is divided into two segments: $\Omega := [0, \Omega_{\text{train}}]$ and $\Omega^{'} := (\Omega_{\text{train}}, \Omega_{\text{test}}]$, where $\Omega_{\text{test}} > \Omega_{\text{train}} > 0$. The numerical or deep learning-based simulator is used in the first stage to simulate the dynamics in $\mathcal{X}_1$, and the problem reduces to predicting the dynamics in the testing domain $\mathcal{X}_2$ in the second stage. Concretely, the objective is to make predictions beyond X_1, i.e., on X_2, and to assess how well the trained models could be generalized through the two-stage training strategy.

The next section presents the proposed two-stage methodology for generalizing the dynamic simulators.

5.4. METHODOLOGY

The proposed two-stage methodology merges dynamic simulators and neural ODE-based methods to generalize the dynamics. The key steps of the proposed two-stage methodology are presented in Algorithm 2 and Fig. 5.1. The first stage entails simulating the engineering dynamics through a preferable simulator tailored for the application, providing flexibility to the workflow. Simulators are computational tools in form of software or modules that act as surrogates for the real-world systems through mathematical models and algorithms, allowing engineers and scientists to analyze and predict the performance of complex systems under various conditions without the need for physical prototypes.

However, numerical methods like FEM or a deep learning approach like physics-informed neural networks (PINNs) [32] simulate the problem in a confined domain. To mitigate the challenge of predictions in a larger domain, the second stage in the proposed methodology employs neural ODEs to capture the intrinsic dynamics from the data generated in the first stage and generalize the dynamics to larger domains, making them further applicable and advantageous for real-world systems. The following subsections describe both the stages of the proposed methodology in detail.

5.4.1. FIRST STAGE - NUMERICAL/PIML SIMULATOR

This chapter employs two dynamic simulators to exemplify that the proposed methodology remains invariant to the simulator employed in the first stage. The first simulator employed is an advanced version of a physics-informed neural network that enforces causality in the learning algorithm termed causal PINN [33]. Causal PINN predictions of beam dynamics are utilized to test temporal generalizations. The simulator causal PINN could be described by considering an abstract PDE with implicit initial and boundary conditions defined by

$$\mathcal{K}(x,t) := \mathcal{D}[u](x,t) - f(x,t) \quad \forall (x,t) \in D \times T \subset \mathbb{R}^{\mathrm{d}} \times \mathbb{R}, \tag{5.1}$$

where $\mathcal{K}$ is the abstract physical equation and $\mathcal{D}[\cdot]$ denotes the differential operator, u as the quantity of interest, $x \in D \subset \mathbb{R}^{\mathrm{d}}$, $t \in T \subset \mathbb{R}$ for $d \geq 1$. The spatial domain D is contained in the d-dimensional Cartesian space and, T denotes the temporal domain, $f(x,t)$ refers to the external force.

5

Causal PINN is based on a feedforward neural network, where the inputs (x,t) map to output (u) through an iterative composition of hidden layers. To train causal PINN, the loss function containing the physical model of a PDE is minimized along with initial and boundary conditions. The loss function is formulated such that the network first minimizes the loss corresponding to lower times before resolving the solutions at higher times. Mathematically, the loss function is defined as

$$J = \min_{\theta} \frac{1}{N} \left[L(t_1) + \sum_{i=2}^{N} e^{-\eta \sum_{k=1}^{i-1} L(t_k)} L(t_i) \right], \tag{5.2}$$

where the loss components are defined as

$$L(t_n) = \frac{1}{N} \sum_{n=1}^{N} ||\mathcal{K}(x,t_n)||^2, \tag{5.3}$$

where (x,t_n) represents the training tuple for each time step n. The total number of training points inside the computational domain is denoted by N, and η is the causality hyperparameter, which depends on the complexity of the problem. Minimizing the loss function (5.2) using a suitable optimization algorithm provides optimal parameters θ. Following the causal PINN training, its testing is conducted on k_t uniform time steps in T and k_x uniform locations in D, making a total of $k_t \cdot k_x$ testing points within the training domain. Following the causal PINN training, its testing is conducted on k_t uniform time steps in T and k_x uniform locations in D, making a total of $k_t \cdot k_x$ testing points within the training domain.

Algorithm 2 Proposed two-stage methodology for generalizing dynamic simulators

Input: Problem and domain, $(x, t; \mu) \in D \times T \times M$.
Output: Generalized dynamic predictions, $u(x, t; \mu) \in \mathcal{X}_2$.

Stage 1: simulation with causal PINN or black-box FEM:

if Causal PINN **then**
  Train causal PINN.
  Test causal PINN for $u(x, t; \mu) \in \mathcal{X}_1$ at k_t time steps.
else
  Collect spatio-temporal data for black-box FEM in $\mathcal{X}_1$.
end if

Stage 2: Training neural differential equations:

Integrate ODEs into the recurrent architecture for updating hidden states.
Utilize predictions from Stage 1 to train LEM or CoRNN in $\mathcal{X}_1$.

Loop for Generalization:

for $i = (k_t + 1)$ to $(k_t + m)$ **do**
  Input condition at time step $i - 1$.
  Use trained model from Stage 2 to predict dynamics at parametric or time step i in $\mathcal{X}_2$.
end for
return $u(x, t; \mu) \in \mathcal{X}_2$

The second simulator employed is a black-box FEM. The physical models, which are the PDEs governing the beam dynamics, are available for causal PINN. Several systems in engineering and industry do not possess a white-box operator and are rather black-box. The efficacy of the proposed methodology is demonstrated for such scenarios by taking the solutions from FEM as the output of the first stage, considering the governing model and solution strategy as a black box. FEM predictions are utilized to test parametric generalizations. Specifically, in one of the numerical experiments, deflection profiles of beams with different loading are taken as the output of the first stage. Here, the parameter is the load, and the quantity of interest is the deflection profile. Another numerical experiment studies the real-world uplift of catenary contact wire in railway systems, depending on the different speeds of the train. The solutions obtained from the causal PINN and FEM are reshaped for further use in the second stage.

5.4.2. SECOND STAGE - NEURAL ODE

Generalization is an open challenge for both methods, causal PINN and FEM. The second stage aims to mitigate this challenge by employing neural ODE-based architecture to capture temporal and parametric dependency. Neural ODEs model the evolution of hidden states over

time using differential equations, leveraging the longstanding potential of ODEs in handling complex temporal dynamics. In particular, this work employs two neural ODE methods, CoRNN and LEM, processing the outputs from the first stage as sequential data and predicting the dynamics outside the training domain. CoRNN employs second-order ODE to model the dynamics of hidden states preserving long-term dependencies and effectively mitigating the vanishing and exploding gradient problem. By incorporating damping factors and oscillatory components in the ODE, CoRNN enforces the computed hidden states to remain within bounds, enhancing the training stability. LEM employs a system of first-order coupled differential equations to update the hidden states. The coupled equations allow LEM to maintain a robust representation of sequential data, addressing the exploding and vanishing gradient issue typical with recurrent neural network-based methods. n the following, the ODEs used for CoRNN and LEM are presented in detail.

CORNN

CoRNN updates the hidden states, $\boldsymbol{y} = \boldsymbol{y}(\omega) \in \mathbb{R}^m$ by solving the following second-order ODE,

$$\boldsymbol{y}'' = \sigma\big(\boldsymbol{W}\boldsymbol{y} + \mathcal{W}\boldsymbol{y}' + \boldsymbol{V}\boldsymbol{u} + \boldsymbol{b}\big) - \gamma\boldsymbol{y} - \epsilon\boldsymbol{y}'. \tag{5.4}$$

Here $\boldsymbol{y}$ represents the hidden state, $\boldsymbol{y}'$ and $\boldsymbol{y}''$ denote the first and second derivative of the hidden state respectively. The activation function is $\sigma(u) = \tanh(u)$. The weight tensors for the hidden state and its first derivative are represented by $\boldsymbol{W}, \mathcal{W} \in \mathbb{R}^{m\times m}$ and the bias term is $\boldsymbol{b} \in \mathbb{R}^m$. The weight tensor for the input ($\boldsymbol{u} \in \mathbb{R}^{k_x}$) is given by $\boldsymbol{V} \in \mathbb{R}^{m\times k_x}$. The terms $\gamma, \epsilon > 0$ are the hyperparameters representing oscillation frequency and damping.

The motivation to employ ODE (5.4) to update the hidden states is attributed to its underlying capabilities in modeling complicated nonlinear oscillations [12]. The dynamics of the ODE could be analyzed for a simplified case by setting $k_x = m = 1$ in (5.4) with an identity activation function $\sigma(u) = u$. Considering the terms $\boldsymbol{W} = \mathcal{W} = \boldsymbol{V} = \boldsymbol{b} = \epsilon = 0$, the ODE reduces to, $\boldsymbol{y}'' + \gamma\boldsymbol{y} = 0$, modeling the well-known spring-mass simple harmonic motion with frequency γ. Including further terms in this simplified ODE, like $\epsilon > 0$, induces damping in the system. For a non-zero vector $\boldsymbol{V}$, the system experiences a driving force proportional to the input signal $\boldsymbol{u}$, where $\boldsymbol{V}$ and $\boldsymbol{b}$ modulate the influence of this force. The tensor $\boldsymbol{W}$ affects the oscillation frequency, while $\mathcal{W}$ influences the damping effect within the system. Additionally, introducing the tanh activation function introduces a nonlinear dynamic response in the oscillator.

Substituting $\boldsymbol{z} = \boldsymbol{y}'(\omega) \in \mathbb{R}^m$, (5.4) could be transformed to the first-order system

$$\boldsymbol{y}' = \boldsymbol{z}, \quad \boldsymbol{z}' = \sigma(\boldsymbol{W}\boldsymbol{y} + \mathcal{W}\boldsymbol{z} + \boldsymbol{V}\boldsymbol{u} + \boldsymbol{b}) - \gamma\boldsymbol{y} - \epsilon\boldsymbol{z}. \tag{5.5}$$

Discretizing (5.5) using an explicit scheme with a time step $0 < \Delta t < 1$,

$$\begin{aligned} \boldsymbol{y}_n &= \boldsymbol{y}_{n-1} + \Delta t \boldsymbol{z}_n, \\ \boldsymbol{z}_n &= \boldsymbol{z}_{n-1} + \Delta t \sigma(\boldsymbol{W}\boldsymbol{y}_{n-1} + \mathcal{W}\boldsymbol{z}_{n-1} + \boldsymbol{V}\boldsymbol{u}_n + \boldsymbol{b}) \\ &\quad - \Delta t \gamma \boldsymbol{y}_{n-1} - \Delta t \epsilon \boldsymbol{z}_n. \end{aligned} \tag{5.6}$$

In the coupled-ODE system with $m > 1$, each neuron updates its hidden state by incorporating external input signals and other neurons. The diagonal components of $\boldsymbol{W}$, along with the scalar hyperparameter γ, regulate the intrinsic oscillatory frequency of individual neurons, whereas the diagonal elements of $\mathcal{W}$, together with the hyperparameter ϵ, govern the damping effects. The off-diagonal entries of these matrices serve to modulate the interaction dynamics between neurons. Further deep networks yield rich global dynamics, suggesting that such oscillator networks can achieve high expressivity, making them capable of approximating complex outputs from sequential inputs [12]. Finally, the output is computed through a learnable linear transformation, $\nu_n \in \mathbb{R}^{k_x}$ with $\nu_n = \mathcal{Q}\mathbf{y}_n$ and $\mathcal{Q} \in \mathbb{R}^{k_x \times m}$.

LEM

Akin to CoRNN, LEM uses a system of differential equations to update the hidden states. However, the system of equations to be solved in LEM is,

$$\begin{aligned} \boldsymbol{y}' &= \hat{\sigma}(\boldsymbol{W}_2\boldsymbol{y} + \boldsymbol{V}_2\boldsymbol{u} + \boldsymbol{b}_2) \odot [\sigma(\boldsymbol{W}_y\boldsymbol{z} + \boldsymbol{V}_y\boldsymbol{u} + \boldsymbol{b}_y) - \boldsymbol{y}] \\ \boldsymbol{z}' &= \hat{\sigma}(\boldsymbol{W}_1\boldsymbol{y} + \boldsymbol{V}_1\boldsymbol{u} + \boldsymbol{b}_1) \odot [\sigma(\boldsymbol{W}_z\boldsymbol{y} + \boldsymbol{V}_z\boldsymbol{u} + \boldsymbol{b}_z) - \boldsymbol{z}]. \end{aligned} \tag{5.7}$$

In addition to the previously stated learnable quantities, LEM additionally learns weight tensors $W_{1,2}, W_{y,z} \in \mathbb{R}^{m \times m}$, weight tensors for input $V_{1,2}, V_{y,z} \in \mathbb{R}^{m \times k_x}$, bias vectors $b_{1,2}$, and $b_{y,z} \in \mathbb{R}^m$. The function $\hat{\sigma}$ and $\odot$ represent the sigmoid activation function and componentwise product of vectors, respectively. $\boldsymbol{y}$ and $\boldsymbol{z}$ are hidden state vectors. The output of LEM is linearly transformed in the same way as in the case of CoRNN. A discretization of (5.7) using explicit Euler scheme results in,

$$\begin{aligned} \boldsymbol{\Delta t}_n &= \Delta t \hat{\sigma}(\boldsymbol{W}_1\boldsymbol{y}_{n-1} + \boldsymbol{V}_1\boldsymbol{u}_n + \boldsymbol{b}_1) \\ \overline{\boldsymbol{\Delta t}}_n &= \Delta t \hat{\sigma}(\boldsymbol{W}_2\boldsymbol{y}_{n-1} + \boldsymbol{V}_2\boldsymbol{u}_n + \boldsymbol{b}_2) \\ \boldsymbol{z}_n &= (1 - \boldsymbol{\Delta t}_n) \odot \boldsymbol{z}_{n-1} \\ &\quad + \boldsymbol{\Delta t}_n \odot \sigma(\boldsymbol{W}_z\boldsymbol{y}_{n-1} + \boldsymbol{V}_z\boldsymbol{u}_n + \boldsymbol{b}_z) \\ \boldsymbol{y}_n &= (1 - \overline{\boldsymbol{\Delta t}}_n) \odot \boldsymbol{y}_{n-1} \\ &\quad + \overline{\boldsymbol{\Delta t}}_n \odot \sigma(\boldsymbol{W}_y\boldsymbol{z}_n + \boldsymbol{V}_y\boldsymbol{u}_n + \boldsymbol{b}_y). \end{aligned} \tag{5.8}$$

The proposed methodology utilizes the simulator solutions to extrapolate through the neural ODE methods, CoRNN, and LEM. However, a key challenge for recurrent neural architectures is the exploding and vanishing gradient problem. The theoretical analysis of CoRNN and LEM presented in [12, 13], and collated in the form of propositions below, motivate employing them for extrapolating the beam dynamics. The detailed proofs of bounds can be found in [12, 13]. The proposed methodology utilizes the simulator solutions to extrapolate through the neural ODE methods, CoRNN, and LEM. However, a key challenge for recurrent neural architectures is the exploding and vanishing gradient problem. The theoretical analysis of CoRNN and LEM presented in [12, 13], motivate employing them for extrapolating the beam dynamics. The detailed proofs of bounds can be found in [12, 13].

The next section presents the numerical experiments to validate the proposed methodology.

5.5. NUMERICAL EXPERIMENTS

Four distinct numerical experiments concerning generalizing dynamic simulations are presented. The complexity of the experiments ranges from fundamental beam theories for beams to real-world catenary-pantograph interactions in railway systems. The first two experiments involve simulating PDEs modeled by the Euler-Bernoulli and Timoshenko theories on the Winkler foundation, where generalization is sought in the temporal domain. The final two experiments aim to generalize in the parametric space. The third experiment is the moving load problem, which aims to predict the mid-point beam deflection under various loading conditions. The final experiment involves predicting catenary contact wire uplift for novel train speeds, considering it as a parameter. The following subsections present the test cases, hyperparameters, and error metrics, along with discussions of the performance of the proposed methodology. The final experiment aims to solve an inverse problem by estimating the applied force on a system of Timoshenko beams.

5.5.1. TEST CASES

EULER-BERNOULLI AND TIMOSHENKO BEAM ON THE WINKLER FOUNDATION

The first two experiments are the PDEs governing the Euler-Bernoulli and Timoshenko beams on the Winkler foundation, described in detail in [6]. For the dynamic simulator in the first stage, causal PINN is employed. Four hidden layers, 200 neurons, and the tanh activation function are utilized for training causal PINN. L-BFGS optimizer is utilized with a learning rate 0.1, with 10000 epochs. The causality hyperparameter is 5, and $\mathcal{X}_1$ is divided into 100-time steps. The training utilized 500 initial

points, 1000 boundary points, and 10000 interior points. For both cases, the training and testing dataset is divided as $T_{\text{train}} = 0.8T_{\text{test}}$.

The objective is to make predictions beyond $\mathcal{X}_1$, i.e., on $\mathcal{X}_2$, and to evaluate the potential of the proposed method for temporal generalization. The training dataset for the second stage, LEM or CoRNN, is generated by testing causal PINN at 256 spatial locations across 160-time steps, implying $k_x = 256$ and $k_t = 160$. Finally, LEM and CoRNN testing is performed on the untrained temporal domain for 40 time steps.

MOVING LOAD

This experiment aims to validate the potential of the proposed methodology in generalizing in the parametric domain. Despite being a fundamental problem across structural engineering, simulating moving load problems within commercial finite element packages is computationally expensive [34], and generalizing it would aid the engineers in downstream predictions of deflection profiles at a reduced computational cost. The studied moving load problem represents a train-track or catenary-pantograph interaction in railway systems. In particular, a point force moving across a simply supported beam is considered.

A black-box finite element-based method is used to compute 100 mid-point deflection profiles of beams for varying loading ranging from 1 N to 6 N. Training and testing dataset is divided as $M_{\text{train}} = 0.8M_{\text{test}}$, i.e., the first 80 equispaced deflection profiles between the loads 1 N to 5 N are used to train the LEM and CoRNN. Specifically, the training dataset size for the proposed method represents mid-point deflection at 344 temporal locations across 80 different loadings. The beam deformations are predicted for 19 unseen equispaced loadings between 5 N to 6 N through LEM and CoRNN.

CATENARY CONTACT WIRE UPLIFT

The final experiment aims to validate the method of generalizing a real-world catenary-pantograph interaction in the parametric domain. In the first stage, the interaction between the pantograph head and contact wire is modeled, and the calculation of dynamic contact wire uplift due to the contact forces is performed in a black-box sense. The method employed a validated finite element model using the absolute nodal coordinate formulation (ANCF) characterizing catenary nonlinearity. Additionally, a simplified lumped mass model simulated the three critical modes of the pantograph [35] as shown in Fig. 5.2. However, for this work, only the data of catenary contact wire uplift deflections for different speeds of train [36] serves as the output from

the first stage, and the model and method details are treated as black-box.

The train speed spans from $50\,\mathrm{km\,h^{-1}}$ to $90\,\mathrm{km\,h^{-1}}$, incrementing in intervals of $10\,\mathrm{km\,h^{-1}}$. The dataset contains the catenary contact wire deflection at 353 spatial locations across five speeds. The recurrent networks in the second stage are trained on two specific speeds, $50\,\mathrm{km\,h^{-1}}$ and $60\,\mathrm{km\,h^{-1}}$. The trained recurrent models are used for predicting the contact wire deflection for three novel train speeds, which are $70\,\mathrm{km\,h^{-1}}$, $80\,\mathrm{km\,h^{-1}}$ and $90\,\mathrm{km\,h^{-1}}$.

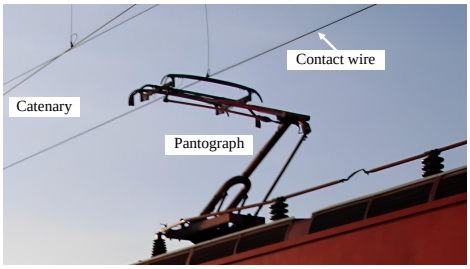
INVERSE PROBLEM FOR TIMOSHENKO DOUBLE BEAM SYSTEM

In addition, an experiment regarding inverse problem is carried out to showcase the potential of the proposed method in handling ill-posed problems. Inverse problems involve determining unknown parameters or functions based on known observables within a system. Such problems are typically ill-posed, requiring additional data at specific locations for the observables. These unknowns, often referred to as quantities of interest, include force functions, initial conditions, boundary conditions, or parameters. The proposed two-stage method predicts quantities of interest in unseen domains. An inverse problem is solved for a Timoshenko double beam system connected by a Winkler foundation [14] to estimate the unknown force function acting on the system, given displacement profiles of the beam system.

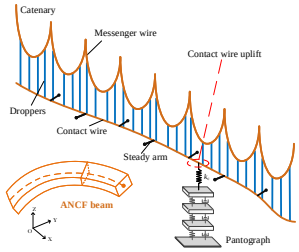
5.5.2. BASELINES, HYPERPARAMETERS AND ERROR METRICS

The second stage in the proposed methodology is based on recurrent neural architectures. Hence, the comparisons for all the experiments are carried out with traditional sequential architectures, RNN, LSTM, and GRU replacing LEM and CoRNN in the second stage. In addition, comparisons with further advanced methods for modeling sequential data, such as neural ordinary differential equation (NODE) [28] and transformers have been carried out. All methods employed the Adam optimizer to train the recurrent architecture. Consistent hyperparameters are chosen across different methods, and different hyperparameters are mentioned as follows. All methods employed the Adam optimizer to train the recurrent architecture.

For Euler-Bernoulli and Timoshenko beam experiments, the learning rate and hidden size are 0.0001 and 32, respectively. CoRNN and LEM-specific parameters, Δt, γ, and ϵ, are taken to be 0.05, 1, and 0.01, respectively. Specific to the transformer, the number of attention heads is 8. The model consists of 6 encoder-decoder layers. The dimension of the feedforward network within the transformer is 512. A total of 400000 epochs are carried out for the Euler-Bernoulli case and 200000 for the Timoshenko case.



(a) Real-world catenary-pantograph setup in railways



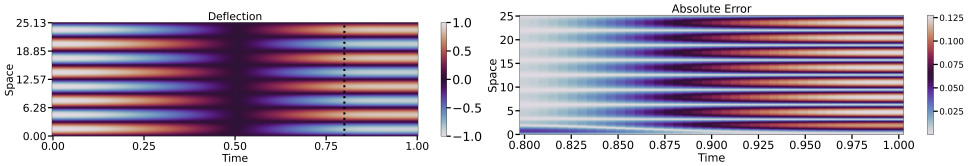
(b) Catenary-pantograph interaction simulation model. The Catenary is modeled by ANCF beam elements (contact and messenger wires) and cable elements (droppers). The pantograph is a lumped mass model with three degrees of freedom.

Figure 5.2: Catenary-pantograph experiment: Real and model perspectives.

For the moving load problem, the chosen hyperparameters, including the learning rate, hidden size, number of epochs, and Δt, maintain consistent values across all models: 0.00001, 160, 20000, and 0.05, respectively. For training the transformer, the number of attention heads is 8, with three encoder-decoder layers. The dimension of the feedforward network within the transformer is 128. Similarly, for CoRNN, the hyperparameters γ and ϵ remain fixed at 1 and 0.01, respectively.

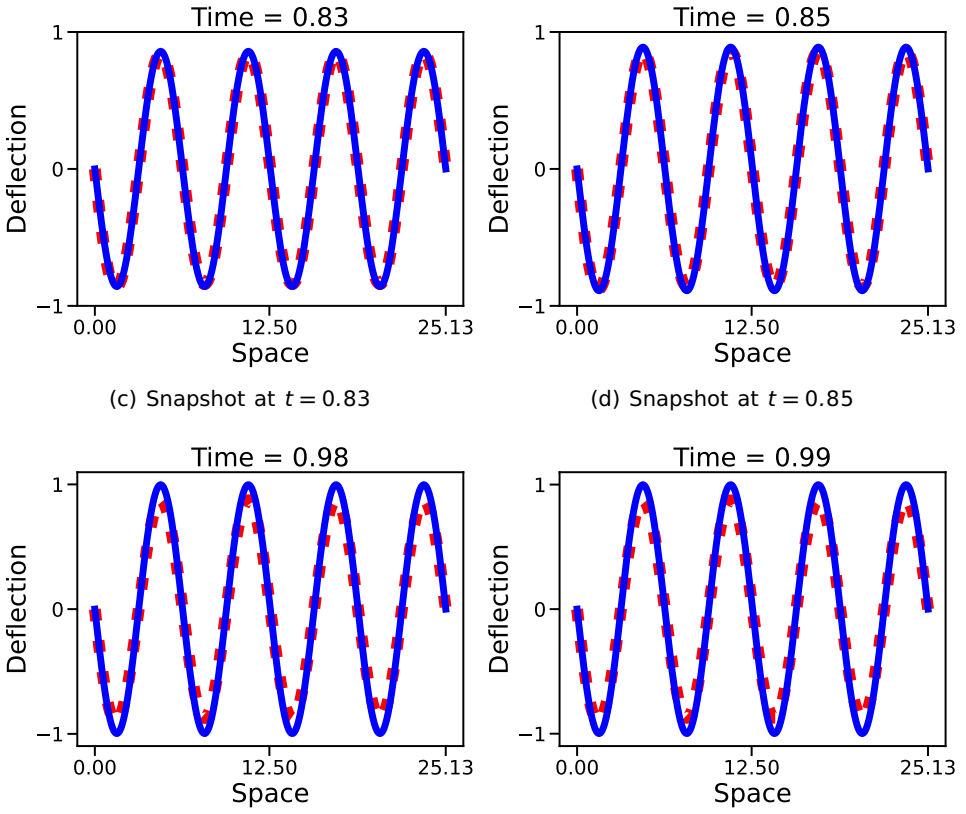
Subsequently, for the catenary-pantograph experiment, the hyperparameters for all sequential models, comprising the learning rate, hidden size, number of epochs, and Δt, are configured to 0.001, 128, 20000, and 0.1, respectively. Analogously, in line with prior instances, the hyperparameters γ and ϵ for CoRNN are set at 1.0 and 0.01, respectively. For the transformer, the number of attention heads is 8 with six transformer layers and 512 as the dimension of the feedforward network.

Four different error metrics are used to evaluate the results. First, relative $L2$-norm of the quantity of interest $\hat{u}$ which is defined with respect to the ground truth u as $\frac{\|\hat{u}-u\|_2}{\|u\|_2}$. The second error metric is the maximum absolute error (max error) computed as $\max_{i=1}^{n} |u_i - \hat{u}_i|$. Here, $|.|$ represents the absolute value function. Here, u_i represents the ground truth at the i^{th} data point and $\hat{u}_i$ represents the predicted value at the i^{th} data point.



(a) Deflection causal PINN-LEM

(b) Absolute error for deflection in generalized temporal domain



(c) Snapshot at $t = 0.83$

(d) Snapshot at $t = 0.85$

(e) Snapshot at $t = 0.98$

(f) Snapshot at $t = 0.99$

Figure 5.3: Euler-Bernoulli beam on the Winkler foundation: Various analyses and temporal snapshots.

The third error metric is the explained variance score given by

$$1 - \frac{\sum_{i=1}^{n}(u_i - \hat{u}_i)^2}{\sum_{i=1}^{n}(u_i - \bar{u})^2}$$

where n is the number of data points, and $\bar{u}$ represents the mean

of the ground truth. Finally, the last metric is the mean absolute error calculated by $\frac{1}{n}\sum_{i=1}^{n}|u_i - \hat{u}_i|$, where the symbols have their same meaning.

5.5.3. RESULTS

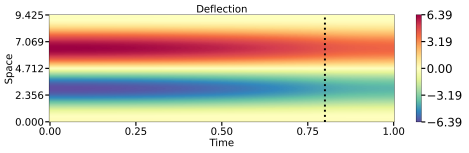
EULER-BERNOULLI AND TIMOSHENKO BEAM ON THE WINKLER FOUNDATION

Fig. 5.3 and 5.4 represent the predictions obtained by LEM for the Euler-Bernoulli and Timoshenko experiments, respectively. Fig. 5.3 top row left depicts the deflection of the beam over unseen time, and the right of the top row shows the absolute error obtained in the deflection prediction. The second row presents the snapshots of deflection predictions at four instances of unseen time. The red dots indicate predictions, and the solid blue line represents the deflection. Table 5.1 compares the neural differential equation-based methods with the traditional sequential methods for generalization. While RNN and other sequential methods exhibit some generalization ability, LEM outperforms them across all metrics, as evidenced by Table 5.1. Additionally, the other sequential methods demonstrate less accuracy in predicting unseen domains. Thus, both Fig. 5.3. Table 5.1 collectively demonstrates that LEM achieves higher accuracy predicting beam deflection on the Winkler foundation than alternative methods.

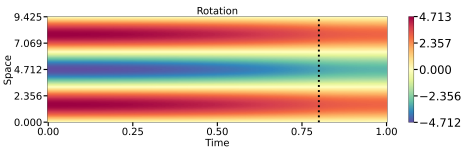
Fig. 5.4 top two rows depict the Timoshenko beam deflection and rotation on the Winkler foundation over time. The second row in Fig. 5.4 displays the absolute errors of deflection and rotation observed in the generalized time, showing a minor increase in error as time progresses. The last row presents the snapshots of deflection and rotation predictions at two different instants of time. The red dots indicate predictions, and a solid blue line represents the actual deflection simulated using causal PINN. Table 5.1 compares the proposed and other sequential methods for Timoshenko beam deflection and rotation for the unseen temporal domain. LEM outperforms them across all metrics, as seen by the results provided in Table 5.1.

MOVING LOAD

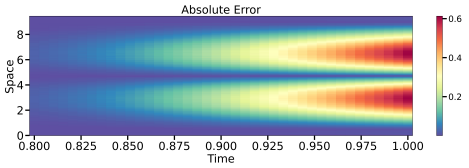
Fig. 5.5 illustrates the beam deflections for a moving load across a simply supported beam. The top two rows in Fig. 5.5 show the predicted deflections and the obtained absolute errors for different loading for LEM. The absolute error increases minorly with an increase in the loading. The last two rows of Fig. 5.5 present the snapshots of the predicted deflections. The red dots depict predictions, while the blue solid line represents actual deflection simulated using the finite element method. Table 5.1 compares LEM and CoRNN with other sequential methods, showcasing the better generalization abilities of the proposed



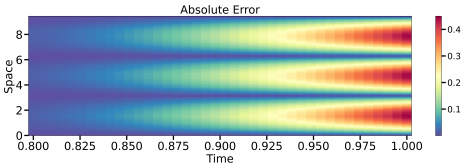
(a) Deflection causal PINN-LEM



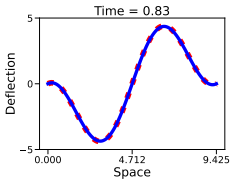
(b) Rotation causal PINN-LEM



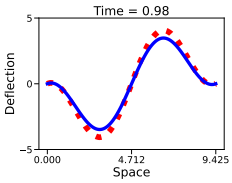
(c) Absolute error for deflection in generalized temporal domain



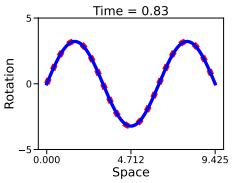
(d) Absolute error for rotation in generalized temporal domain



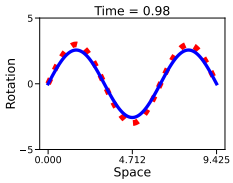
(e) Deflection snapshot at $t = 0.83$



(f) Deflection snapshot at $t = 0.98$



(g) Rotation snapshot at $t = 0.83$



(h) Rotation snapshot at $t = 0.98$

Figure 5.4: Timoshenko beam on the Winkler foundation: Detailed analysis and temporal snapshots.

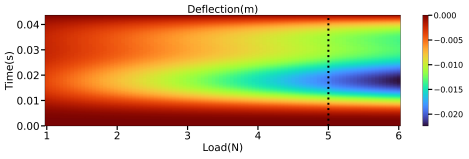
method.

CATENARY CONTACT WIRE UPLIFT

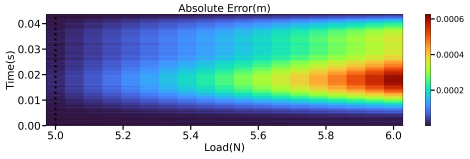
Fig. 5.6. illustrates catenary contact wire uplift deflections at varying speeds ($70\,\mathrm{km\,h^{-1}}$, $80\,\mathrm{km\,h^{-1}}$, and $90\,\mathrm{km\,h^{-1}}$). The red dots depict predictions, while the blue solid line represents actual deflection simulated using the finite element method. Fig. 5.6. (a), (c) and (e) represent the contact wire uplift predictions at $70\,\mathrm{km\,h^{-1}}$, $80\,\mathrm{km\,h^{-1}}$, and $90\,\mathrm{km\,h^{-1}}$, respectively. Fig. 5.6. (b), (d) and (f) show absolute errors in those uplift predictions. The errors for different speeds fall within a similar range, yet it is evident that the error increases minorly with the increase in train speed. Table 5.1 compares LEM and CoRNN with other sequential methods, showcasing that LEM performs better in all metrics. Hence, Fig. 5.6. and Table 5.1 demonstrate that LEM predicts catenary contact wire uplift more accurately than other methods.

Table 5.1: Beam deflection, rotation, moving load, and catenary uplift. Higher (or lower) values are preferred by ↑ (or ↓).

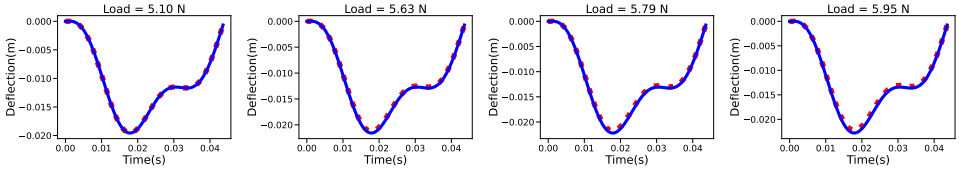
	RNN	LSTM	GRU	CoRNN	LEM	Transformer	NODE
Euler-Bernoulli Beam Deflection							
L2-norm(↓)	0.012	0.4925	1.9471	0.0194	**0.0080**	0.0084	0.0081
Max error(↓)	0.2527	0.7814	1.8173	0.1842	**0.1276**	0.1345	0.2305
Explained Variance score(↑)	0.9874	0.5074	-0.9470	0.9805	**0.9919**	0.9915	0.9918
Mean absolute error(↓)	0.0585	0.4163	0.8015	0.0749	**0.0466**	0.0470	0.0410
Timoshenko Beam Deflection							
L2-norm(↓)	0.3813	0.3167	0.3798	0.2060	**0.0058**	0.0228	0.0095
Max error(↓)	2.9385	2.7387	2.9378	2.2792	**0.6126**	1.0249	0.6649
Explained Variance score(↑)	0.6186	0.6832	0.62017	0.7939	**0.9941**	0.9771	0.9904
Mean absolute error(↓)	1.2769	1.1622	1.2734	0.9330	**0.1270**	0.2758	0.1789
Timoshenko Beam Rotation							
L2-norm(↓)	0.3813	0.3146	0.3797	0.2059	**0.0055**	0.0228	0.0095
Max error(↓)	2.1680	2.0224	2.1683	1.6808	**0.4496**	0.7523	0.4907
Explained Variance score(↑)	0.6183	0.6847	0.6198	0.7935	**0.9942**	0.9766	0.9902
Mean absolute error(↓)	1.1368	1.0310	1.1334	0.8306	**0.1105**	0.2452	0.1593
Moving Load Mid-Point Beam Deflection							
L2-norm(↓)	0.5579	0.3803	0.6026	0.0039	**0.0002**	0.1307	0.0084
Max error(↓)	0.0190	0.0194	0.0188	0.0024	**0.0006**	0.0156	0.0042
Explained Variance score(↑)	0.2974	0.2714	0.3444	0.9927	**0.9995**	0.7056	0.9802
Mean absolute error(↓)	0.0084	0.0068	0.0088	0.0006	**0.0001**	0.0039	0.0009
Catenary Uplift with Different Train Speeds							
L2-norm(↓)	1.49e-6	1.54e-7	3.0e-7	6.38e-8	**6.43e-8**	1.01e-6	1.32e-6
Max error(↓)	0.0845	0.0085	0.0135	0.0063	**0.0063**	0.0194	0.0259
Explained Variance score(↑)	0.7728	0.9722	0.9562	0.9934	**0.9932**	0.8451	0.7961
Mean absolute error(↓)	0.0022	0.0015	0.0019	0.0009	**0.0009**	0.0043	0.0041



(a) Mid-point deflection FEM-LEM



(b) Absolute error in mid-point deflection for unseen loadings



(c) Snapshot at Load 5 N (d) Snapshot at Load 5.63 N (e) Snapshot at Load 5.79 N (f) Snapshot at Load 5.95 N

Figure 5.5: Mid-point deflection for Moving load: Analysis and load snapshots.

INVERSE PROBLEM FOR TIMOSHENKO DOUBLE BEAM SYSTEM

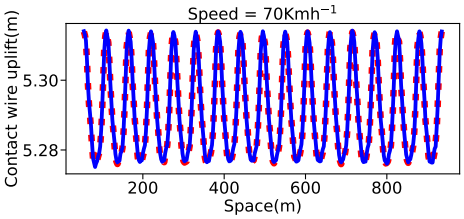
To demonstrate the capability of the two-stage approach for out-of-domain prediction in inverse problems, a Timoshenko double-beam

system is considered [14]. The goal is to predict the unknown force acting on the beam system outside the training domain. PINN simulates the force function in the first stage. In the second stage, this data is utilized to train LEM, with testing performed in the out-of-domain region. Fig. 5.7 presents the force function acting on the beam and the absolute error in predicting the force in an extrapolation scenario. The error metric results for this experiment include the L2 error, maximum absolute error, mean absolute error, and explained variance score, obtained as 0.0224, 0.1349, 0.0309, and 0.9720, respectively. The results illustrate the proposed method's efficacy in extrapolating an inverse problem.

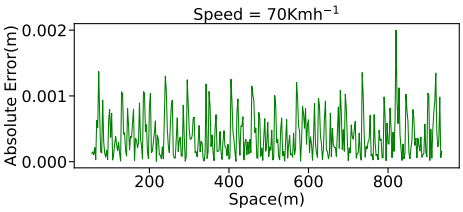
5.6. CONCLUSIONS

This work addressed the longstanding challenge of generalizing simulations for engineering and industry. A resolution-invariant pipeline is proposed, processing data at various resolutions by introducing a novel two-stage approach combining state-of-the-art simulators with classical mathematical models infused in neural recurrent architectures. The presented approach efficiently generalizes dynamics in the time and parameter space without relying on data from the untrained domain, thus eliminating the need for tedious re-meshing and re-simulation in computer-aided simulation software. Moreover, the proposed workflow demonstrated superior performance over traditional recurrent architectures, as evidenced by its successful application to various dynamic simulation problems in structural engineering, including real-world scenarios like catenary-pantograph interactions in railway systems. This research holds promise in various industrial applications for enhancing simulation efficiency and accuracy in out-of-domain predictions.

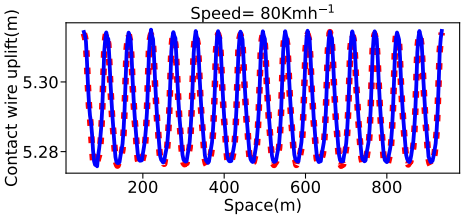
Potential future works include extending the proposed methodology to high-dimensional problems and complex geometries such as shells and bridges. Another direction could focus on accelerating the simulations through optimization techniques and parallel computing, enabling real-time predictions. Additionally, industrial applications often contend with uncertainties and stochastic variations. Enhancing the method to account for uncertainties in factors such as material properties, loading conditions, and external influences would allow for probabilistic predictions, thereby increasing the robustness and applicability of the method in diverse industrial scenarios.



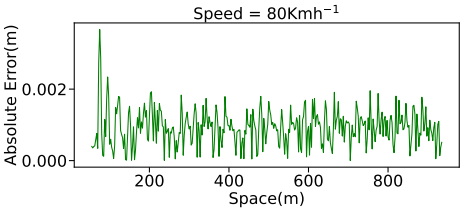
(a) Contact wire uplift prediction at 70 km h^{-1}



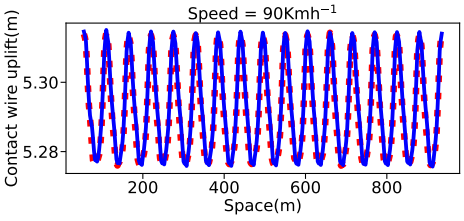
(b) Absolute error at 70 km h^{-1}



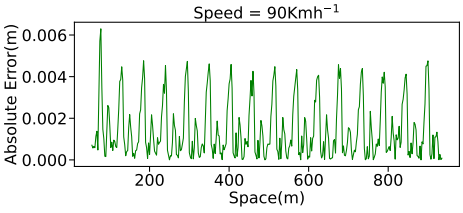
(c) Contact wire uplift prediction at 80 km h^{-1}



(d) Absolute error at 80 km h^{-1}



(e) Contact wire uplift prediction at 90 km h^{-1}



(f) Absolute error at 90 km h^{-1}

Figure 5.6: Contact wire uplift with different train speeds: **Left column:** Contact wire uplift prediction for unseen speeds. **Right column:** Absolute errors in contact wire uplift predictions for different train speeds.

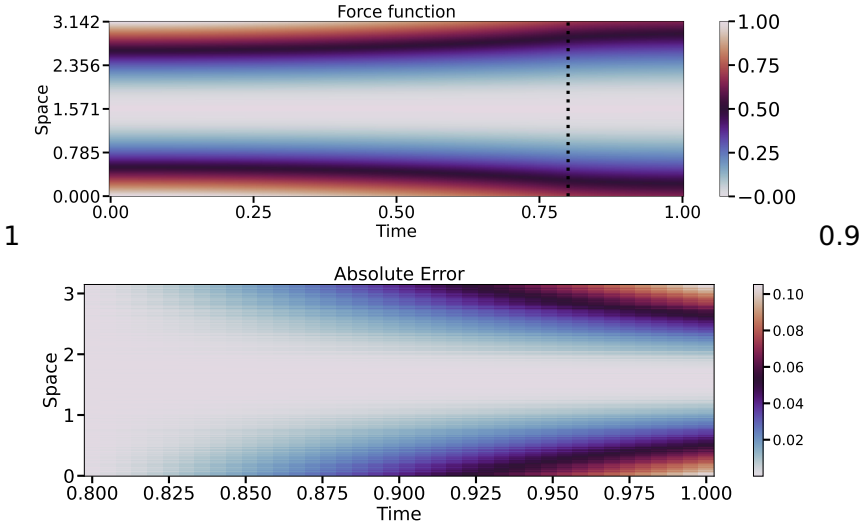


Figure 5.7: Approximation of the force function (**top**) for the inverse problem and corresponding absolute error in extrapolation (**bottom**).

REFERENCES

[1] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann. “Experimentable digital twins—Streamlining simulation-based systems engineering for industry 4.0”. In: *IEEE Transactions on Industrial Informatics* 14.4 (2018), pp. 1722–1731. doi: 10.1109/TII.2018.2804917.

[2] F. Tao, H. Zhang, A. Liu, and A. Y. Nee. “Digital twin in industry: State-of-the-art”. In: *IEEE Transactions on Industrial Informatics* 15.4 (2018), pp. 2405–2415. doi: 10.1109/TII.2018.2873186..

[3] R. J. Smit, W. M. Brekelmans, and H. E. Meijer. “Prediction of the mechanical behavior of nonlinear heterogeneous systems by multi-level finite element modeling”. In: *Computer Methods in Applied Mechanics and Engineering* 155.1-2 (1998), pp. 181–192. doi: 10.1016/S0045-7825(97)00139-4.

[4] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. J. Inman. “A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications”. In: *Mechanical Systems and Signal Processing* 147 (2021), p. 107077. doi: 10.1016/j.ymssp.2020.107077.

[5] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440. doi: 10.1038/s42254-021-00314-5.

[6] T. Kapoor, H. Wang, A. Núñez, and R. Dollevoet. “Transfer learning for improved generalizability in causal physics-informed neural networks for beam simulations”. In: *Engineering Applications of Artificial Intelligence* 133 (2024), p. 108085. doi: 10.1016/j.engappai.2024.108085.

[7] Y. Wang, Y. He, B. Kang, J. Liu, and C. Sun. “ProbSparse Attention-based Fault Diagnosis for Industrial Robots Under Different Working Conditions”. In: *IEEE Transactions on Instrumentation and Measurement* (2024). doi: 10.1109/TIM.2024.3374323.

[8] X. Fu, X. Yang, P. Zanchetta, M. Tang, Y. Liu, and Z. Chen. “An adaptive data-driven iterative feedforward tuning approach based on fast recursive algorithm: with application to a linear motor”. In: *IEEE Transactions on Industrial Informatics* 19.4 (2022), pp. 6160–6169. doi: 10.1109/TII.2022.3202818.

[9] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao. “Independently recurrent neural network (indrnn): Building a longer and deeper rnn”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5457–5466. doi: 10.48550/arXiv.1803.04831.

[10] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014). doi: 10.48550/arXiv.1406.1078.

[12] T. K. Rusch and S. Mishra. “Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies”. In: *International Conference on Learning Representations*. 2021. doi: 10.48550/arXiv.2010.00951.

[13] T. K. Rusch, S. Mishra, N. B. Erichson, and M. W. Mahoney. “Long Expressive Memory for Sequence Modeling”. In: *International Conference on Learning Representations*. 2022. doi: 10.48550/arXiv.2110.04744.

[14] T. Kapoor, H. Wang, A. Nunez, and R. Dollevoet. “Physics-informed neural networks for solving forward and inverse problems in complex beam systems”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023). doi: 10.1109/TNNLS.2023.3310585.

[15] M. Eisenberger and J. Clastornik. “Vibrations and buckling of a beam on a variable Winkler elastic foundation”. In: *Journal of sound and vibration* (). doi: https://doi.org/10.1016/0022-460X(87)90469-X.

[16] F. Zonzini, M. Zauli, M. Mangia, N. Testoni, and L. De Marchi. “Model-assisted compressed sensing for vibration-based structural health monitoring”. In: *IEEE Transactions on Industrial Informatics* (). doi: 10.1109/TII.2021.3050146.

[17] M. David, W. Derigent, G. Loubet, A. Takacs, and D. Dragomirescu. “Communicating materials: Communicating concrete development for construction industry”. In: *IEEE Transactions on Industrial Informatics* (). doi: 10.1109/TII.2024.3352261.

[18] F. P. Beer, E. R. Johnston, J. T. DeWolf, D. F. Mazurek, and S. Sanghi. *Mechanics of Materials*.

[19] E. Carrera, G. Giunta, and M. Petrolo. *Beam structures: classical and advanced theories*. John Wiley & Sons, 2011.

[20] A. Öchsner and A. Öchsner. "Timoshenko beam theory". In: *Classical Beam theories of structural mechanics* (2021), pp. 67–104.

[21] H.-P. Deng, Y.-B. He, B.-C. Wang, and H.-X. Li. "Physics-dominated neural network for spatiotemporal modeling of battery thermal process". In: *IEEE Transactions on Industrial Informatics* (2023). doi: 10.1109/TII.2023.3266404.

[22] X. Lu, F. Yin, C. Liu, and M. Huang. "Online spatiotemporal extreme learning machine for complex time-varying distributed parameter systems". In: *IEEE Transactions on Industrial Informatics* 13.4 (2017), pp. 1753–1762. doi: 10.1109/TII.2017.2666841.

[23] X. Lu, D. Xu, and W. Liu. "Development of spatiotemporal recurrent neural network for modeling of spatiotemporal processes". In: *IEEE Transactions on Industrial Informatics* 17.1 (2020), pp. 189–198. doi: 10.1109/TII.2020.2967810.

[24] K. Xu, H. Yang, C. Zhu, X. Jin, B. Fan, and L. Hu. "Deep extreme learning machines based two-phase spatiotemporal modeling for distributed parameter systems". In: *IEEE Transactions on Industrial Informatics* 19.3 (2022), pp. 2919–2929. doi: 10.1109/TII.2022.3165870.

[25] J. Li, Y. Wang, Y. Zi, H. Zhang, and C. Li. "Causal consistency network: A collaborative multimachine generalization method for bearing fault diagnosis". In: *IEEE Transactions on Industrial Informatics* 19.4 (2022), pp. 5915–5924. doi: 10.1109/TII.2022.3174711.

[26] C. Xie and S. Chen. "A Physics-Guided Reversible Residual Neural Network Model: Applied to Build Forward and Inverse Models for Turntable Servo System". In: *IEEE Transactions on Industrial Informatics* 19.4 (2022), pp. 5882–5890. doi: 10.1109/TII.2022.3200668.

[27] C. Liu and K. Gryllias. "Simulation-driven domain adaptation for rolling element bearing fault diagnosis". In: *IEEE Transactions on Industrial Informatics* 18.9 (2021), pp. 5760–5770. doi: 10.1109/TII.2021.3103412.

[28] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. "Neural ordinary differential equations". In: *Advances in Neural Information Processing Systems* 31 (2018).

[29] Z. Yuan, Y. Wang, X. Ban, C. Ning, H.-N. Dai, and H. Wang. “Autonomous-Jump-ODENet: Identifying Continuous-Time Jump Systems for Cooling-System Prediction”. In: *IEEE Transactions on Industrial Informatics* (2022). doi: 10.1109/TII.2022.3207835.

[30] S. Wang, Y. Wang, B. Yang, F. Mo, and Z. Zhang. “Variational Bayesian Learning with reliable likelihood approximation for accurate Process Quality Evaluation”. In: *IEEE Transactions on Industrial Informatics* (2023). doi: 10.1109/TII.2023.3264288.

[31] L. Qin, S. Zhang, T. Sun, and X. Zhao. “An Interpretable Neuro-Dynamic Scheme With Feature-Temporal Attention for Remaining Useful Life Estimation”. In: *IEEE Transactions on Industrial Informatics* (2023). doi: 10.1109/TII.2023.3333933.

[32] M. Raissi, P. Perdikaris, and G. E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational physics* 378 (2019), pp. 686–707. doi: 10.1016/j.jcp.2018.10.045.

[33] S. Wang, S. Sankaran, and P. Perdikaris. “Respecting causality for training physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 421 (2024), p. 116813. doi: 10.48550/arXiv.2203.07404.

[34] L. Frỳba. *Vibration of solids and structures under moving loads*. Vol. 1. Springer Science & Business Media, 2013. doi: 10.1007/978-94-011-9685-7.

[35] Y. Song, H. Wang, and Z. Liu. “An investigation on the current collection quality of railway pantograph-catenary systems with contact wire wear degradations”. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–11. doi: 10.1109/TIM.2021.3078530.

[36] Y. Song, H. Wang, G. Frøseth, P. Nåvik, Z. Liu, and A. Rønnquist. “Surrogate modelling of railway pantograph-catenary interaction using deep Long-Short-Term-Memory neural networks”. In: *Mechanism and Machine Theory* 187 (2023), p. 105386. doi: 10.1016/j.mechmachtheory.2023.105386.

6

CONCLUSIONS AND RECOMMENDATIONS

This dissertation develops and implements physics-informed machine learning-based methodologies for simulating fundamental structural and railway dynamics expressed in the form of beam systems. The research objectives are outlined in Chapter 1, and six research questions are posed to achieve the research objectives. Answers to the questions are elaborated throughout Chapters 2-5 and are summarized in this chapter. Future research directions are recommended based on the main findings and their implications.

6.1. CONCLUSIONS

This section presents the main findings answering the six research questions as follows:

1. How to predict beam deformations through PIML-based methodologies, mitigating the challenge of multiscale coefficients?

   The research tackles the challenge of PINNs in simulating and developing neural surrogate models for multiscale complex beam PDEs to predict beam deformations. A framework is introduced in Chapter 2, integrating nondimensional fourth-order physical equations into the vanilla PINN loss function. Through nondimensionalization, the coefficients of the PDE terms are scaled, which assists in efficiently training the neural networks. Moreover, employing dimensionless equations eliminates the influence of measurement units, generalizing the methodology to multiple scenarios and facilitating comparative analysis across diverse physical systems, expressed in ratios and parameters. Consequently, the integration of dimensionless equations into PINNs not only enhances the convergence of the

optimizer but also generalizes the approach to adhere to distinct scenarios, contributing to the field of structural engineering.

The efficacy of this proposed approach is also validated in Chapter 3 to simulate beam dynamics on a Winkler foundation in a large domain. Notably, the nondimensional form of the beam PDE is utilized, further underlining the significance and applicability of this framework. In addition, Chapters 4 and 5 demonstrate the utility of the nondimensionalization approach by employing it in the simulation for beam deformation out-of-domain prediction. By consistently utilizing the nondimensional form of the physical model, these chapters underscore the robustness and versatility of the proposed methodology across a range of simulation scenarios.

2. How effectively do physics-informed algorithms tackle inverse beam dynamic problems and predict underlying dynamics and unknown parameters from noisy data?

   The inverse problem in engineering involves estimating unknown parameters or functions from a set of measured data. In PINNs, this problem is typically addressed by fitting the measured data and known physical laws to train the neural network. However, measured data can be affected by various noise sources, posing challenges in accurately estimating the quantity of interest. The noise can render the measured data unreliable, leading to inaccuracies in estimating unknown parameters or functions by the neural network. Consequently, in such scenarios, the optimizer of the neural network may not converge.

   In many real-world scenarios, data is often noisy. Chapter 2 employs noisy data to predict the parameters of beam dynamics. The proposed nondimensional PINN framework in Chapter 2 is utilized to tackle ill-posed inverse problems in complex systems, aiming to identify unknown model parameters and the applied force on beam components. This approach utilizes data from indirect measurements such as beam displacement and cross-sectional rotations, exhibiting robustness to noise and the ability to accommodate uncertainty in measurement data. Hence, PINN-based methodology is well-suited for real-world applications with incomplete or uncertain data. PINNs are effective in integrating incomplete or noisy information with prior physical knowledge.

3. How to simulate beam deformations within large spatiotemporal domains using PIML?

   Another open problem encountered by vanilla PINNs pertains to simulating beam deformation in large space-time domains. This challenge arises from the training process, as vanilla PINNs prioritize

training at higher time levels due to implicit gradient bias. This results in violations of temporal causality and inaccurate solutions, particularly for problems highly dependent on initial conditions.

An inherent causal structure characterizes physical systems. For example, the deflection of a beam at any given time is causally linked to its previous state (deflection), the physical properties of the beam, and the external forces acting upon it. This causality is fundamental to accurately modeling beam behavior in response to loads, rendering it a valuable tool in engineering and physics. PINN models can effectively learn complex solutions to PDEs when causal relationships are considered, enabling progressive sequential-time learning of the solution.

Chapter 3 proposes training PINNs while respecting causality within the context of structural engineering, referred to as causal PINN. Our goal is to address the training challenge and achieve precise predictions of beam dynamics in large space-time domains. This challenge is tackled by modifying the training approach of PINNs and enforcing training at lower time levels before progressing to subsequent ones. Consequently, a weighted loss function is employed, incorporating a causality parameter to preserve the inherent physical causality governing beam dynamics. The causal PINN approach, validated through numerical experiments, demonstrates enhanced accuracy in prediction.

4. How to accelerate and generalize PIML-based methodologies for simulating similar beam dynamic problems in large spatiotemporal domains?

   Chapter 3 proposes the integration of transfer learning with causal PINN. Transfer learning involves taking a pre-trained model (developed for one task) and reusing it as the starting point for a model on a second task. Employing transfer learning entails leveraging the knowledge (i.e., model parameters) from previously solved problems. This approach allows the model to accelerate the initial learning phase, significantly reducing training time and computational cost.

   Incorporating transfer learning enables the PINN models to quickly adapt to new problems without starting from scratch, enhancing their generalizability and efficiency. The proposed approach decreases the computational cost and speeds up convergence in subsequent simulations. Consequently, the proposed approach improves simulation performance and efficiency for dynamic beam simulations on elastic foundations, indicating the broad application of PIML in practical engineering problems and digital twin models.

5. How to develop PIML-based frameworks capable of predicting out-

of-domain for canonical problems, including nonlinear and high-order problems?

Chapter 4 investigates this question, focusing on the limitations of PIML methodologies in generalizing the PDE solutions beyond the training domain. The research therein aims to enhance the generalization capabilities of PIML, thereby facilitating its practical application in real-world scenarios where accurate predictions in unexplored scenarios are essential.

Chapter 3 discusses the importance of incorporating causality for simulating PDEs. Leveraging the inherent causality and temporal sequential characteristics of PDE solutions, Chapter 4 introduces integrating PIML models with recurrent neural architectures. These architectures are based on systems of ordinary differential equations called neural oscillators. By effectively capturing long-time dependencies and addressing issues such as the exploding and vanishing gradient problem, neural oscillators improve generalization in PIML predictions.

The research introduces combining neural oscillators with physics-informed machine learning to enhance performance in unseen regions. This hybrid approach enables the model to learn the long-time dynamics of solutions to the governing partial differential equations. The effectiveness of the proposed approach is demonstrated through experimentation on three benchmark nonlinear PDEs: viscous Burgers, Allen-Cahn, and Schrödinger equations, as well as the biharmonic Euler-Bernoulli beam equation. Results highlight the improved generalization performance of PIML augmented with neural oscillators, outperforming state-of-the-art methods across various metrics.

Hence, incorporating neural oscillators surpasses existing state-of-the-art methods on benchmark problems across various metrics, thereby enhancing the generalization capabilities of PIML and providing accurate solutions for extrapolation and prediction beyond the training data.

6. How to generalize beam dynamic solvers to predict out-of-domain scenarios, particularly for beam systems like catenaries with varying train speeds?

   This question is investigated in Chapter 5, addressing the long-standing challenge of generalizing simulations for engineering and industry. A resolution-invariant pipeline is proposed, processing data at various resolutions. A two-stage approach is proposed, combining state-of-the-art simulators with classical mathematical models infused in neural recurrent architectures.

   Firstly, the method employs state-of-the-art simulators tailored to

the specific application. Secondly, the predictions from the first stage are utilized to train a recurrent neural architecture for generalizing the dynamics. In particular, Chapter 5 utilizes causal physics-informed neural networks and black-box finite element simulations in the first stage, followed by ODEs integrated into the recurrent architecture in the second stage to capture the intrinsic dynamics and overcome generalization limitations.

By integrating ODEs with recurrent networks, the model efficiently captures the inherent causality and generalizes the dynamics irrespective of the data source. The presented numerical experiments encompass fundamental structural engineering scenarios. In particular, four different beam dynamics are studied, including beams on Winkler foundations, beams under moving loads, and real-world catenary-pantograph interactions in the railway systems, to exhibit the potential of the proposed methodology.

Numerical experiments show that the proposed method efficiently generalizes dynamics in the time and parameter space without relying on data from the untrained domain, thus eliminating the need for tedious re-meshing and re-simulation in computer-aided simulation software.

Numerical experiments highlight its prospects for design optimization, predictive maintenance, and enhancing safety measures. This research holds promise in various industrial applications for enhancing simulation efficiency and accuracy in out-of-domain predictions.

6.2. FUTURE RESEARCH DIRECTIONS

Motivated by the research and conclusions of this dissertation, this section presents future recommendations in the form of chapter-wise future research directions and general recommendations.

6.2.1. CHAPTER-WISE RESEARCH DIRECTIONS

- Chapter 2 studied beam deformations within double-beam systems. This investigation lays the foundation for potential extensions to more complex systems involving multiple interconnected beams resembling bridge structures. Understanding and simulating the dynamics of bridges holds promise for enhancing structural designs, leading to safer and more efficient infrastructure.
- Moreover, chapter 2 tackles inverse problems related to predicting beam parameters by utilizing simulation data of beam deformation and rotation. While this approach provides valuable insights, future advancements could leverage real sensor measurements to refine predictions and estimate unknown parameters within inverse

problems. Incorporating sensor data into the modeling process can enhance accuracy and reliability, contributing to robust engineering solutions and informed decision-making in structural design and analysis.

- Chapter 3 presents the causality-respecting framework for beam simulation in large domains. This work focuses on the single beam on a Winkler foundation that can be extended to simulate beam deformation for multiple connected systems for large domains. Also, transfer learning is studied in this chapter to utilize the trained weights as initialization for similar models. Different transfer learning strategies can be utilized to advance the simulation of beam deformation.
- Chapter 4 introduces an innovative framework that facilitates out-of-domain prediction for canonical problems through PIML. This framework represents an advancement in scenarios where traditional approaches struggle to extrapolate beyond the training data. The proposed methodology offers a versatile solution that can be extended to tackle higher-dimensional problems, providing a scalable approach to simulate complex systems across diverse domains, from engineering and physics to finance and healthcare.
- Chapter 5 introduces a framework tailored for out-of-domain prediction within the context of beam dynamic simulations. The chapter employs a combination of numerical methods and machine learning techniques to develop simulators capable of accurately predicting the behavior of beams in scenarios beyond those covered by the training data. The framework incorporates data from physical experiments or prototypes, providing accurate beam dynamic simulations in real-world scenarios and enhancing their applicability to practical engineering problems. Moreover, the versatility of this framework enables its potential utilization across various domains for extrapolation tasks. Beyond beam dynamics, similar methodologies could be applied to a wide range of engineering and scientific problems where accurate prediction of system behavior is crucial. By leveraging this framework, researchers and practitioners can explore new avenues for predictive modeling and gain deeper insights into complex systems.

6.2.2. FUTURE PIML RESEARCH DIRECTIONS

- Developing faster and more accurate PIML algorithms is an important research direction for advancing PIML-based simulations in various fields. Speed is crucial for real-time applications, enabling rapid decision-making and analysis of dynamic systems. Accuracy ensures reliable predictions, enhancing the utility of PIML models in

critical scenarios such as structural design or medical diagnosis. By enhancing algorithmic efficiency and precision, researchers can improve the scalability and applicability of PIML methods across diverse domains. Real-world applications entail optimizing computational workflows, reducing training and inference times, and minimizing computational resource requirements. Additionally, advancements in accuracy involve refining model architectures, enhancing regularization techniques, and integrating domain-specific knowledge to improve predictive performance. Overall, the development of faster and more accurate PIML algorithms drives innovation, accelerates research progress, and paves new possibilities for solving complex real-world problems efficiently.

- Benchmark physical model datasets are crucial for validating and benchmarking new algorithms. These datasets provide standardized references for evaluating algorithm performance, ensuring reliability and applicability across diverse scenarios. Researchers can assess the effectiveness, efficiency, and scalability of the algorithm by utilizing benchmark datasets, facilitating comparisons with existing methods. Moreover, benchmark datasets foster collaboration and knowledge sharing among research communities, driving advancements in the field. Additionally, they serve as valuable educational resources, enabling students and practitioners to gain practical experience with real-world problems. Overall, the availability of benchmark physical model datasets accelerates innovation, promotes reproducibility, and enables researchers to address complex challenges.

- Engineering applications often contend with uncertainties and stochastic variations. A prospective direction for future research involves enhancing the developed models to account for these uncertainties and enable probabilistic predictions. A crucial step in this process would be developing models capable of quantifying uncertainty, increasing their applicability to real-world problems. By incorporating probabilistic predictions, the models could gain greater trustworthiness, and their accuracy and confidence levels could be better established. Moreover, this approach would mitigate uncertainties arising from the specific computer system and seeds under investigation, leading to more applicable and generalizable models.

- Developing foundational models for a specific class of systems, where variations in initial conditions, boundary conditions, or force functions lead to predictable deformations, is crucial. Such a model would enable efficient prediction of deformations across various scenarios. This foundational model would be a valuable tool for engineers, scientists, inframanagers and practitioners, facilitating rapid

analysis, design optimization, and decision-making processes. Foundation models by predicting for novel scenarios in real-time, might identify the gaps between theory and practice and increase trust and reliance on AI. Moreover, it would pave the way for advancements in predictive modeling, enabling more effective exploration and utilization of these systems in various applications. Ultimately, the development of such foundational models contributes to the advancement of scientific knowledge and the innovation of practical solutions in engineering and related fields.

ACKNOWLEDGEMENTS

First and foremost, I am deeply thankful to my grandmother and my parents. Words cannot fully express my gratitude for everything you have done for me. Your unwavering support has been the cornerstone of my happiness and success. I cannot even imagine a single day without your calls. I thank my brother Abhi for always supporting and believing in me. I dedicate this thesis to my brother because I won't be here without him. It's his dream. To my sister Simi, thank you for all the unforgettable moments and the endless laughter.

This dream of becoming a Dr. would not have been possible without my supervisors, who gave me this opportunity. I am deeply grateful to my supervisors, Dr. Hongrui Wang, Dr. Alfredo Núñez, and Prof. Rolf Dollevoet, for allowing me to work with them on such a fascinating topic.

Hongrui, I especially appreciate the flexibility you gave me to explore and solve various complex problems. I value the freedom you gave me to navigate my research and the guidance you provided whenever needed. I am equally thankful to my co-supervisor, Alfredo, for your insightful feedback and research expertise, which have been instrumental in my development. Special thanks to my promotor, Rolf, for several fruitful discussions and his availability whenever I need him. Thank you very much for everything.

A big thanks to our section head, Prof. Zili Li, for the financial support for my research visits to conferences, workshops, hackathons, summer schools, and autumn schools. I am also immensely grateful to our section secretary, Jacqueline, for her constant assistance with minor and significant challenges since my arrival in the Netherlands. A special thanks to Ellard for translating my thesis and engaging in enriching discussions about my research work and beyond.

I am thankful to all the members of the Railway Engineering Section for our stimulating discussions and activities outside of work. Of course, a big thanks to those with whom I spent much time and have many unforgettable memories: Li, Wassamon, Yuanchen, Gokul, Nikhil, and Shashanka—my big special thanks to Anahat Kelkar for designing my thesis cover. You are the best. I appreciate your patiently listening to my thesis and designing an excellent cover beyond my imagination. I am also thankful to Aleksandr Dekhovich for helping me with the steps for defense and for exciting discussions. Thanks to Guanqun for several wonderful trips and meals we enjoyed together. Thanks, Chinmay, for several interesting discussions and endless laughter. Thanks, Bram, for

translating the proposition into Dutch. Thanks, Gerlof, for the amazing dinner party.

I also extend my gratitude to my collaborators, Prof. Daniel M. Tartakovsky, Chinmay Datar, Abhishek Chandra, Dr. Mitrofan Curti, Dr. Felix Dietrich, Anastasios Stamou, and Kareem El Sayed, for the productive and successful collaborations. Thankful to Dr. Mitrofan Curti for providing me with office space and an exciting presentation during my research visit. I also thank Dr. Alexander, Dr. Panos and Dr. Raj for several interesting discussions. I am grateful to Prof. George Karniadakis for inviting me to present my research work.

I sincerely thank my thesis committee members, Prof. Daniel M. Tartakovsky, Prof. Stephen Roberts, Prof. Siddhartha Mishra, and Prof. Kees Vuik, for reviewing my thesis. I am grateful to have you all for my thesis evaluation.

Daniel, I learned a lot working with you and how patiently you manage everything, and I am genuinely thankful for the time you spent on several exciting discussions. Your encouragement means a lot to me. Steve, I cannot express how much I enjoy our discussions. Grateful for your trust in me. Sid, thank you for introducing me to this topic and for your availability whenever I need you. I am truly honored to have had the opportunity to benefit from their vast expertise.

DISCLAIMER

Lastly, I would like to thank funding agencies as this research was partly supported by ProRail and Europe's Rail Flagship Project IAM4RAIL - Holistic and Integrated Asset Management for Europe's RAIL System. Funded by the European Union. Views and opinion expressed are however those of the authors(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101101966.

CURRICULUM VITæ

Taniya Kapoor

09-08-1996 Born in New Delhi, India.

RESEARCH INTERESTS

Fusion of scientific machine learning and structural engineering. Broadly, the research focuses on SciML, physics-informed machine learning, partial differential equations, and computational sciences.

WORK EXPERIENCE

July 2024 – Present
Postdoctoral Researcher
Delft University of Technology, The Netherlands

Nov 2021 – Jun 2024
PhD
Delft University of Technology, The Netherlands
Thesis: *Physics-informed machine learning: from methods to beam structures*

Mar 2021 – Oct 2021
Master Internship
ETH Zürich, Switzerland

EDUCATION

Sep 2020 – Aug 2021
Master of Science
High Performance Scientific Computing
Université de Lille, France

Jul 2017 – Jun 2019
Master of Science
Applied Mathematics
South Asian University, India

Jul 2014 – Jun 2017
Bachelor of Science
Mathematics (Honours)
University of Delhi, India

REVIEWER ACTIVITIES

For Journals	IEEE TNNLS, EAAI, Engineering with Computers, Thin-Walled Structures, Applied mathematical modeling, SoftwareX.
For Conferences	NeurIPS 2024, NeurIPS DLDE 2023 Workshop, ICLR 2024 AI4DiffEqtnsInSci Workshop, IJCNN, EURODYN.

TEACHING ACTIVITIES

Supervision	Supervised nine master students on a course project related to physics-informed machine learning for beam dynamics (3 months).
	Supervised two master students on a research project focused on the generalization of beam dynamics using neural ODEs (4 months).
	Co-supervised a master thesis on neural networks infused with physics for beam systems (1 year).

COURSE MATERIAL

Course Project	Data Science and AI for Engineers
Tutorials	A1, MUDE, Dynamica

LIST OF PUBLICATIONS

DISSERTATION RELATED PUBLICATIONS

4. **Kapoor, T.**, Wang, H., Stamou, A., Sayed, K.E., Nunez, A., Tartakovsky, D. M., & Dollevoet, R. (2024). Generalizing beam dynamics from simulator data with neural differential equations. Under review.

3. **Kapoor, T.**, Chandra, A., Tartakovsky, D. M., Wang, H., Nunez, A., & Dollevoet, R. (2024). Neural oscillators for generalization of physics-informed machine learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 12, pp. 13059-13067).

2. **Kapoor, T.**, Wang, H., Núñez, A., & Dollevoet, R. (2024). Transfer learning for improved generalizability in causal physics-informed neural networks for beam simulations. Engineering Applications of Artificial Intelligence, 133, 108085.

1. **Kapoor, T.**, Wang, H., Núñez, A., & Dollevoet, R. (2023). Physics-informed neural networks for solving forward and inverse problems in complex beam systems. IEEE Transactions on Neural Networks and Learning Systems.

OTHER PUBLICATIONS

6. Datar, C., **Kapoor, T.**, Chandra, A., Sun, Q., Burak, I., Bolager, E. L., Veselovska, A., Fornasier, M., & Dietrich, F. (2024). Solving partial differential equations with sampled neural networks. Under review.

5. Chandra, A., **Kapoor, T.**, Daniels, B., Curti, M., Tiels, K., Tartakovsky, D. M., & Lomonova, E. A. (2023). Neural oscillators for magnetic hysteresis modeling. arXiv preprint arXiv:2308.12002.

4. **Kapoor, T.**, Chandra, A., Tartakovsky, D., Wang, H., Núñez, A., & Dollevoet, R. (2023). Neural oscillators for generalizing parametric PDEs. The Symbiosis of Deep Learning and Differential Equations III, NeurIPS.

3. Kapoor, S., Chandra, A., **Kapoor, T.**, & Curti, M. Gradient weighted physics-informed neural networks for capturing shocks in porous media flows. Machine Learning and the Physical Sciences Workshop, NeurIPS.

2. **Kapoor, T.**, Wang, H., Núñez, A., & Dollevoet, R. (2023). Physics-informed machine learning for moving load problems. Proceedings of the Eurodyn, XII International Conference on Structural Dynamics

1. **Kapoor, T.**, Wang, H., Núñez, A., & Dollevoet, R. (2022). Predicting traction re- turn current in electric railway systems through physics-informed neural networks. IEEE Symposium Series on Computational Intelligence (SSCI).

POSTER PRESENTATIONS

7. **Kapoor, T.**, Chandra, A., Tartakovsky, D. M., Wang, H., Nunez, A., & Dollevoet, R. (2024). Neural Oscillators for Generalization of Physics-Informed Machine Learning. Scientific Machine Learning, Emerging Topics.

6. **Kapoor, T.**, Chandra, A., Tartakovsky, D. M., Wang, H., Nunez, A., & Dollevoet, R. (2024). Neural Oscillators for Generalization of Physics-Informed Machine Learning. AAAI Conference on Artificial Intelligence.

5. **Kapoor, T.**, Chandra, A., Tartakovsky, D. M., Wang, H., Nunez, A., & Dollevoet, R. (2023). Neural oscillators for generalizing parametric PDEs. The Symbiosis of Deep Learning and Differential Equations III, NeurIPS.

4. Kapoor, S., Chandra, A., **Kapoor, T.**, & Curti, M. (2023). Gradient weighted physics-informed neural networks for capturing shocks in porous media flow. Machine Learning and the Physical Sciences, NeurIPS.

3. **Kapoor, T.**, Wang, H., Nunez, A., & Dollevoet, R. (2023). PINNs for complex beam systems. CWI Autumn School.

2. Chandra, A., **Kapoor, T.**, Tartakovsky, D. M., Wang, H., Nunez, A., & Dollevoet, R. (2023). Neural Oscillators for Magnetic Hysteresis Modeling. CWI Autumn School.

1. **Kapoor, T.**, Molinaro, R., & Mishra, S. (2022). Physics Informed Neural Networks for Approximating Fully Nonlinear PDEs. London Mathematical Society Workshop on the Mathematics of Deep Learning.