ENHANCING INFRASTRUCTURE AND DYNAMIC SYSTEMS MODELING THROUGH THE SYNERGY OF PHYSICS-BASED MODELS AND MACHINE LEARNING

By

Xuyang Li

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Civil Engineering—Doctor of Philosophy
Computer Science—Dual Major

2024

ABSTRACT

The convergence of artificial intelligence (AI) with engineering and scientific disciplines has catalyzed transformative advancements in both structural health monitoring (SHM) and the modeling of complex physical systems. This dissertation explores the development and application of AI-driven methodologies with a focus on anomaly detection and inverse modeling for domain-specific and other scientific problems.

SHM is vital for the safety and longevity of structures like buildings and bridges. With the growing scale and potential impact of structural failures, there is a dire need for scalable, cost-effective, and passive SHM techniques tailored to each structure without relying on complex baseline models. Mechanics-Informed Damage Assessment of Structures, MIDAS, is introduced, which continuously adapts a bespoke baseline model by learning from the structure's undamaged state. Numerical simulations and experiments show that incorporating mechanical characteristics into the autoencoder improves minor damage detection and localization by up to 35% compared to standard autoencoders.

In addition to anomaly detection, NeuralSI was introduced for structural identification, estimating key nonlinear parameters in mechanical components like beams and plates by augmenting partial differential equations (PDEs) with neural networks. Using limited measurement data, NeuralSI is ideal for SHM applications where the exact state of a structure is often unknown. The model can extrapolate to both standard and extreme conditions using identified structural parameters. Compared to data-driven neural networks and other physics-informed neural networks (PINN), NeuralSI reduces interpolation and extrapolation errors in displacement distribution by two orders of magnitude.

Building on this approach, the research expands to broader systems governed by parameterized PDEs, which are critical in modeling various physical, industrial, and environmental phenomena. These systems often have unknown or unpredictable parameters that traditional methods struggle to estimate due to real-world complexities like multiphysics interactions and limited data. NeuroFieldID is introduced to estimate unknown field parameters from sparse observations by modeling them as functions of space or state variables using neural networks. Applied to several physical and biomedical problems, NeuroFieldID achieves a 100 times reduction in parameter estimation errors and a 10 times reduction in peak dynamic response errors, greatly enhancing the accuracy and efficiency of complex physics modeling.

ACKNOWLEDGEMENTS

I am deeply grateful to my advisors, Dr. Nizar Lajnef and Dr. Vishnu Boddeti, for their invaluable patience, guidance, and feedback throughout this journey. Their generous sharing of knowledge and expertise made this endeavor possible. I also wish to extend my thanks to my committee members, Dr. Weiyi Lu and Dr. Jiliang Tang, for their continued support.

I am thankful to my friends and lab mates, Hamed Bolandi, Hassene Hasni, Mahdi Masmoudi, Talal Salem, and Gautam Sreekumar, for their insightful comments. Many thanks to Laura Post for her efficient handling of the paperwork and administrative processes, as well as to Bailey Weber, Shelly Harbenski, the administration team, and Joseph Nguyen, the lab technician, for their invaluable assistance.

Lastly, I wish to convey my deepest gratitude and love to my parents, Qinjian Li and Guifeng Yang, for their unwavering love, support, and care. Their many years of guidance during my undergraduate studies provided the essential foundation for this work, and I am forever grateful for their encouragement and belief in me.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION AND DISSERTATION OVERVIEW

1.1 Motivation and vision

Interest in SHM technology has engaged many scientific communities for decades and has recently become the most popular method for monitoring and assessing the integrity of aging structures. These civil infrastructures often provide essential public facilities and services, making their safety a critical concern for society. However, civil infrastructures often exist in complex environments where many conditions are unknown or uncertain. In such cases, theoretical knowledge alone may not be sufficient for reliable structural assessment and condition prognosis. On the other hand, sensor monitoring data is often redundant and repetitive, lacking significant new information, which complicates the extraction of meaningful insights. By integrating advanced machine learning (ML) algorithms with mechanics domain knowledge, structural systems can be accurately and efficiently characterized as high-dimensional features using neural networks, distinctly separating undamaged states from damaged states.

Additionally, the response (vibrations) of civil infrastructures, or more generally, the response of dynamic systems, is often governed by differential equations. While the functional form of these PDEs is usually known, the parameters within the equations are often unknown or vary with complex spatiotemporal dynamics. In many cases, the governing equations hold, but if the parameters change, it may appear as though the equation does not hold. However, traditional data-driven or ML models typically require large amounts of response data to accurately learn these dynamics. They often perform poorly when applied to prognosis analysis, which is crucial for predicting future structural integrity and preventing system failures. In contrast, by integrating ML algorithms with known knowledge of differential equations, it is possible to accurately estimate the underlying physical parameters that govern the differential equation and system response, enabling robust prediction of future behavior.

This research aims to bridge the gap between theoretical and data-driven approaches by integrating ML to enhance infrastructure and dynamic systems modeling, focusing on anomaly detection and inverse modeling (parameter estimation). In response to these challenges, this work develops novel frameworks for predictive modeling and assessment in complex environments, contributing to safer and more resilient infrastructure systems.

Specifically, for anomaly detection, we aim to develop advanced ML algorithms that can sift through large volumes of sensor data to identify subtle patterns and deviations that may indicate early signs of structural damage. These algorithms will be designed to differentiate between normal operational variations and actual anomalies, ensuring high sensitivity and specificity in damage detection. By leveraging domain knowledge in mechanics, we will ensure that the models are not only accurate but also interpretable, allowing for practical application in real-world monitoring systems.

For parameter estimation, our focus is on integrating ML with the known differential equations governing the response of dynamic systems. We seek to create models capable of estimating field parameters within these equations, even under complex and uncertain environmental conditions. This approach will enable accurate predictions of future behavior in both infrastructure and other dynamic systems, allowing for proactive maintenance and intervention. Ultimately, this contributes to the longevity and safety of critical infrastructure, as well as the reliability of various dynamic systems in broader applications.

1.2 Background and state of knowledge

For the past few decades, SHM has proven to be an effective approach and provided reliable condition assessments in civil infrastructures [1, 2]. One of the biggest tasks for SHM is to identify structural damages. Damages can be indicated by the changes in structural properties and mechanical behaviors [3]. Besides traditional visual inspections, many advanced techniques are recently utilizing sensors [2, 4] and image-based methods [5, 6] to capture structural property variations and therefore, indicate and quantify damages precisely and accurately. Those methods often establish a baseline [7, 8] for a single parameter or in a local area level of a structure.

SHM measures various parameters (i.e., structural response, temperature, velocity, etc.) to help diagnose the health status of structure systems [4]. Among the existing SHM sensing technologies, strain gauges are commonly used to measure strain response and evaluate the structure's health status [9]. Conventional SHM tools and methods tend to provide data snapshots at specific time instances [10], making their interpretation prone to erroneous instantaneous measurements, such as faulty sensors or variations in environmental stimuli. This significantly limits the ability to identify the real source of abnormal responses and could lead to missed damage events. The tremendous amount of SHM data collected needs to be denoised (i.e., missing data needs to be restored) and effectively processed to obtain the desired outcome that accurately represents the information under consideration. The strain output from strain gauges may provide misleading information due to the overlap between strain events (i.e., resulting from structural damages or environmental factors) [10]. Addressing such a research dilemma, the structure can be monitored continuously and data is recorded for the whole loading event.

Besides, many data compression algorithms are developed to reduce data size for efficient monitoring, such as dimension reduction [10, 11], feature extraction [12], correlation functions [13], and hidden Markov model (HMM) [14]. Data reduction was also implemented in several studies, including in wireless sensor networks [13, 11] and for the monitoring of long-span bridges [15, 16]. It was shown that the reduced data could be used to determine damage in gusset plates [17]. and pavement structures [18]. A support vector machine method was used to process sensor data and detect fatigue cracking in steel bridge structures [19].

Furthermore, it is worth noting that the emerging field of ML [20, 4, 21, 22, 23] has been employed in structural damage detection and structural condition assessment of civil infrastructures. Studies indicate that using ML tends to show better performance in terms of speed and accuracy compared with the traditional SHM tools [24, 25]. A residual convolutional neural network (CNN) was proposed for structural modal identification using de-noised signals [26]. Moreover, CNN was also used for real-time monitoring and vibration-based structural condition assessment [27, 28, 29]. Besides feed-forward neural networks, it is worth mentioning that several sequence-based models were also employed to detect, localize, and quantify structural defects. On the other hand, long short-term memory (LSTM) was investigated for damage detection in wind turbine blades [30], rubber bearing [31], and offshore structures [32]. Autoencoders were used to identify the loose steel bolts in the bridge using a reconstruction model of structural responses to detect concrete cracks using structural images [33, 34]. Moreover, autoencoders were used to identify the loose steel bolts in the bridge using reconstruction of structural responses and detect concrete cracks using structural images [35, 36], autoencoders were also used to compress and recover strain data measured from a long-span suspension bridge for data anomaly detection [15]. Compressive sensing was developed and the reconstructed responses were used for structural damage detection and localization [37]. Last but not least, special types of ML models such as zero-shots and few-shots learning and detection are widely used to recognize and disjoin training classes and unseen classes or anomalies, where no anomaly or few anomaly data are used for model training [38, 39, 40].

Other research employs ML [23, 41] for SHM applications. However the overly used damage data in model training [29] is unrealistic and does not represent the situations that occur in real life. This is because damages rarely happen and damages can exist in many forms. Furthermore, structures vary from one to another, and building models for specific structures would be time-consuming and even impractical, not to mention the complicated environmental conditions and different characteristics of damage. However, for most of the built structures, it is extremely difficult to establish baseline characteristics. Most existing structures lack historical information and current service condition data. Even for newly constructed structures, it is impractical to have numerical or theoretical models to represent the built conditions which are generally different from the as-designed parameters due to multiple effects such as uncertainty in boundary conditions, environmental condition variations, temporal variability of materials, and unpredictable construction/manufacturing constraints. This typically means that baseline models are extremely hard to generate using classical computational approaches such as Finite Element Models (FEM).

Another challenge is to estimate an accurate model that clearly describes structural behavior in complex environments. Unlike structural identification which focuses on estimating the structural characteristics and behaviors [42, 43], structural parameter identification narrows the scope on determining the exact parametric values of a known physical or governing model. And this assumption is valid for most structural components such as beams and plates.

The learned parameters can be utilized to predict structural response under different loading conditions [44] and also track any deviations that could indicate damage or degradation in a particular component of the structure over time [45, 46]. Meanwhile, parameters for a large structure system can be identified separately based on each component. This helps assemble a mathematical model that can accurately represent the behavior of the whole structure system. The built model could further help assess the health condition of the structural system and provide valuable information for decision-making. Structural parameter identification could also provide more detailed and accurate information about the physical properties of the structure, such as its material properties [47, 44] and important geometric dimensions. This can be especially useful in some cases where the structure models are complex, and building models for a structured system could be time-consuming, unnecessary, and inaccurate.

In the dynamic analysis of civil structural systems, prior research efforts primarily focused on matching experimental data with either mechanistic models (i.e., known mechanical models) [48, 49] or with black box models with only input/output information (i.e., purely data-driven approaches), [50, 51, 52]. Examples of these approaches include eigensystem identification algorithms [53], frequency domain decomposition [54], stochastic optimization techniques [55], and sparse identification [56]. A majority of these approaches, however, fail to capture highly non-linear behaviors [43] or ignore prior knowledge of the structural model [56].

For accurately identifying the structural system, traditional methods temp to map external excitation to the corresponding structural response using state-space models [57, 58, 59] and sparse component analysis [60, 61, 62]. Besides, many model updating approaches [46] such as Bayesian updating [63, 64, 65] and FEM updating [66, 67, 68, 69] has been applied. Recently, the rapid development of sensing technologies not only contributes to the data-driven methods but also significantly enhances the governing equations discovery/approximation methods [70, 71, 47, 65, 72, 73]. Moreover, ML approaches have been widely utilized in structural system identification due to their nonlinear characteristics in modeling [74, 43]. Different network architectures are implemented, such as LSTM [75, 76] and CNN [77, 78, 79]. Recently, research focuses on modeling with PINNs [78, 80, 81] with the augmented knowledge of constitutive equations (ODEs or PDEs), boundary and initial conditions [82, 83], which acts as a penalizing term to restrict the space and provide more precise acceptable solutions.

Significant efforts have been directed toward physics-driven discovery or approximation of governing equations [84, 85, 86]. Such studies have further been amplified by the rapid development of advanced sensing techniques and ML methods [18, 87, 88, 89]. Most of the work to date has mainly focused on ordinary differential equation (ODE) systems [90, 86, 91] have been widely adopted due to their capacity to learn and capture the governing dynamic behavior from directly collected measurements [92, 91, 93]. They represent a significant step above the direct fitting of a relation between input and output variables. In structural engineering applications, Neural ODEs generally approximate the time derivative of the main physical attribute through a neural network. It has also been widely used in many other real-life problems, such as in the fields of hydrology [94], fluids [95], climate models [96], chemistry [97], causal inference [92], and structures [86]. Compared to direct fitting from traditional ML methods, Neural ODEs build a connection between input and output variables in a brand new perspective. Many applications have employed Neural ODE for dynamic structure parameter identification in both linear and nonlinear cases [86, 92, 91]. Few studies have explored Neural PDEs using lie point symmetry data augmentation [98], PINNs [99], weather and ocean wave data [100], message passing [101], fluid dynamics [98], and graph neural networks (GNN) [102, 103, 104]. And a NeuralPDE solver package was developed in Julia [99] based on PINN.

Besides directly employing neural differential equations for parameter recovery or model identification, some data-driven discovery algorithms for the estimation of parameters in differential equations are introduced. These methods typically referred to as PINNs include differential equations, constitutive equations, and initial and boundary conditions in the loss function of the neural network and adopt automatic differentiation to compute derivatives of the network parameters [82, 105]. Research also focuses on using PINNs for many other applications. Some focus on structural applications such as response prediction in gusset plates [106, 107], wind turbines [108], seismic response [109], and glass structure material [110]. PINNs are also widely used on other subjects such as climate modeling [111], transportation [112], fluid mechanics [113], electromagnetic analysis [114].

1.3 Research hypothesis and objectives

The main hypothesis of this research is that the governing physics underlying civil infrastructure and general dynamic systems are reliable and that by integrating AI with physics-based knowledge, we can more effectively capture and represent the underlying features and information, leading to improved SHM and dynamic system modeling.

The objective of this research is to develop a robust modeling and identification framework for engineering and dynamic systems, beginning with a focus on civil infrastructure and extending to more generalized scientific fields.

In the first part of the research, the goal is to build an automated SHM framework using sensor data, advanced AI technologies, as well as domain knowledge. This framework aims to better capture the underlying features of civil infrastructure, thereby improving the performance of damage detection and localization.

In the second part, the goal is to incorporate physics-based knowledge, specifically differential equations, to precisely and quantitatively model civil infrastructure and dynamic systems. By using observations of infrastructural or dynamic responses, key structural parameters that govern the behavior of these systems can be identified and estimated, leading to more accurate modeling of the system. The proposed framework is also expected to be generalized for various scientific applications.

1.4 Outline

This dissertation is organized as follows: Chapter 2 details the development of a mechanics-informed autoencoder for damage assessment of structures. This unsupervised ML model is built and trained using sensor data exclusively from undamaged structures. The model leverages entirely passive measurements from inexpensive sensors and incorporates data compression techniques to create a "deploy-and-forget" system. Numerical simulations of gusset plates with various crack patterns are conducted, alongside experimental validation using different structural setups, such as gusset plates and beam-column structures. These setups include different anomaly conditions, such as boundary condition variations and the presence of cracks. Damage is detected through the model's continuous learning process, with the performance evaluated using statistical metrics such as accuracy, precision, recall, etc. Damage localization is presented through contour maps. Additionally, the above performance is evaluated with a variable number of sensors. Different conditions, such as noisy data sources and temperature effects, are also considered.

Chapter 3 focuses on developing a neural network framework for structural parameter identification within nonlinear dynamic systems. This framework is designed to discover unknown and hard-to-measure structural parameters governing the PDE from measured sensing data. These parameters, assumed to vary spatially, are modeled through neural networks, enabling the estimation of structural response based on the PDE by minimizing the error between the predicted dynamic response and the actual measurements. As a proof-of-concept, the chapter explores the forced vibration responses of an Euler-Bernoulli beam with spatially varying parameters. Once the unknown system parameters are estimated, the differential model is used to efficiently predict the time evolution of the structural response. The chapter also includes neural network hyper-parameter studies to evaluate the framework's performance and examine its effectiveness under limited training data conditions across various input loading scenarios. This approach replicates real-world challenges in monitoring structures with limited sensors and sampling capabilities. The framework's performance is compared with deep neural networks (DNN) and PINN to demonstrate its effectiveness.

Chapter 4 advances structural parameter identification through experimental validation using a composite beam and numerical analysis of 2D plate structures. The chapter begins with experimental work on a composite beam tested in the lab to identify unknown beam parameters. For more complex structures, such as plate vibration problems, a progressive training technique is introduced to efficiently estimate parameters. Additionally, spline penalty functions are applied during the later stages of network training to ensure smooth parameter estimations in real-life applications.

Chapter 5 further extends the previously discussed approach to broader systems modeled by general parameterized PDEs, which are prevalent in various physical, industrial, and social phenomena. These systems often have unknown or unpredictable parameters that traditional methods struggle to estimate due to real-world complexities like multiphysics interactions and limited data. In this chapter, a general approach is introduced for estimating unknown PDE parameter fields from scarce observations of the system's response. The parameters are modeled and learned as functions of space, time, or state variables through neural networks. A two-step training strategy is proposed to greatly improve the training efficiency and accuracy. The chapter also includes comparisons with PINN-based baselines and other data-driven methods to evaluate the effectiveness of the proposed approach.

Chapter 6 concludes the work performed in this dissertation, presents the main findings, outlines the timeline for the remaining proposed work of the thesis, and provides directions for future research.

CHAPTER 2

MECHANICS INFORMED AUTOENCODER ENABLES AUTOMATED DETECTION AND LOCALIZATION OF UNFORESEEN STRUCTURAL DAMAGE

2.1 Overview

SHM ensures the safety and longevity of structures like buildings and bridges. As the volume and scale of structures and the impact of their failure continue to grow, there is a dire need for SHM techniques that are scalable, inexpensive, can operate passively without human intervention, and are customized for each mechanical structure without the need for complex baseline models. We present MIDAS, a novel “deploy-and-forget" approach for automated detection and localization of damage in structures. It is a synergistic integration of entirely passive measurements from inexpensive sensors, data compression, and a mechanics-informed autoencoder. Once deployed, MIDAS continuously learns and adapts a bespoke baseline model for each structure, learning from its undamaged state’s response characteristics. After learning from just 3 hours of data, it can autonomously detect and localize different types of unforeseen damage. Results from numerical simulations and experiments indicate that incorporating the mechanical characteristics into the autoencoder allows for up to a 35% improvement in the detection and localization of minor damage over a standard autoencoder. Our approach holds significant promise for reducing human intervention and inspection costs while enabling proactive and preventive maintenance strategies. This will extend the lifespan, reliability, and sustainability of civil infrastructures.

SHM plays a vital role in monitoring and ensuring the safety and reliability of various engineering systems. Poor monitoring and maintenance can lead to severe damage or even catastrophic failures of structures. Numerous structural failures have occurred despite frequent manual inspections and the adoption of many active sensing technologies over the years. For instance, a severe crack in the I-40 Bridge in Memphis went undetected for years before being discovered in 2021 [115], resulting in long-term road closure, substantial economic losses, and significant safety concerns among the public. Similarly, in 2022, a bridge in Pittsburgh collapsed due to the corrosion

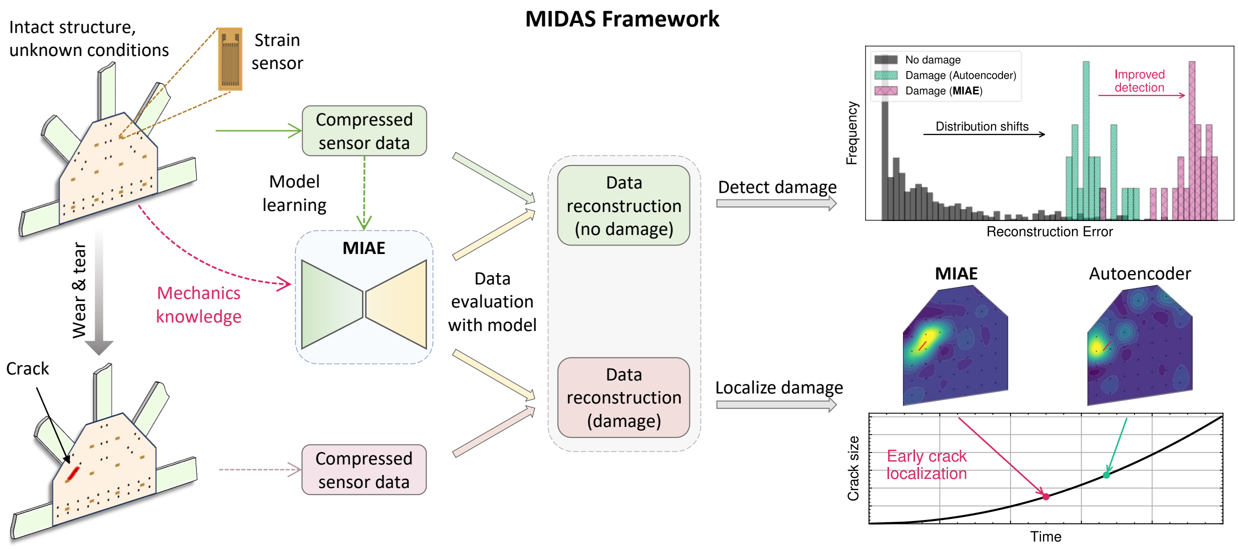


Figure 2.1 Overview of MIDAS. The automated structural damage detection and localization framework. Raw structural response data from the sensors are compressed, and MIAE is trained purely on the response from the structure's undamaged state. No additional information is leveraged besides the pairwise mechanical relations between the strain responses. Once trained, the distribution of reconstruction errors between the network's input and output on the training data serves as a reference representation of an intact structure's response. After deployment, the trained model processes data from the sensors, and resultant reconstruction errors are compared to the reference error distribution to detect and localize potential damage. An observable shift in reconstruction errors (top right) highlights the detection of damage. The incorporated mechanical knowledge notably amplifies the distribution shift, significantly enhancing damage detection at an early stage. Sensor-wise error comparisons are interpolated (heatmaps at the bottom right) to localize anomalies representing the onset of damage.

and deterioration of the bridge legs [116], damaging several vehicles and causing many injuries. Preventing such incidents as the built environment scales and ages necessitates the development of passive, inexpensive, and continuous structural monitoring techniques, with the ultimate aim of detecting, localizing, and identifying different types of damage at an early stage. Such solutions would complement existing active and costly manual inspections.

SHM systems often employ sensors to measure physical quantities such as strain, vibration, and temperature. The measurements are coupled with a numerical model to infer the structure's health condition. Real-world deployment of SHM has to contend with multiple challenges due to the complexity and diversity of structures, sensors, and damage scenarios. First, detecting and localizing damages as early as possible is critical to extend the structure's longevity. However, minor damage, hidden or distributed in the structure, may not readily manifest in the sensor data and cannot be identified by the numerical model. Second, due to the sheer diversity of structures and associated damage they may endure, SHM methods have to contend with unknown or novel damage without being able to rely on prior knowledge or annotated data. Third, multiple sensors are typically used at different locations on the structure. Seamless SHM will require a combination of inexpensive passive sensors and algorithms that can simultaneously and effectively utilize data from multiple sensors.

While many solutions have been developed for SHM, existing solutions are limited in multiple respects. They either need active measurements [117, 118, 119, 120, 121], detect but do not localize damage [120, 119, 122, 123, 124, 125, 18, 126], or employ technology that is accurate but very complex and expensive, such as guided waves [127, 117, 118] and acoustic emissions [120, 121]. Furthermore, some are based on predefined damage features or thresholds [1, 128], designed to model data from a single sensor or do not take the domain attributes of the structure and the sensor placement into account when detecting or localizing damage [36, 129], or are limited to identify known types of damage [27, 130] only.

In the broader context of structural engineering, ML methods are increasingly being relied upon for addressing many problems. For instance, PINNs [131, 105, 132], which leverage both data and knowledge of the underlying physics, and GNNs) [133, 134, 135, 136, 137] are commonly being employed for forward and inverse problems. These solutions promise significant computational gains over traditional numerical methods. However, the need for precise knowledge of the governing equations, parameters, loading, etc., limits their applicability for detecting and localizing damage in the real world, where such information is usually unavailable.

Current SHM solutions instead rely on more traditional ML such as support vector machines [19] for steel bridge structures, neural networks for buildings [138], concrete slabs [123], pavement [130], and steel frames [124], and recurrent neural networks [139], LSTM and gated recurrent units [30] to detect, localize, and quantify structural defects. Such solutions have also been proposed to detect damage in gusset plates [140, 141, 142, 143], bridges [144, 145, 16], highway sections [146], and railways [147, 148].

The primary drawback of the aforementioned body of work is their need for annotated sensor data with labels corresponding to normal or damaged operating conditions. Obtaining such annotations in large quantities and for each deployment is costly and impractical. Furthermore, models learned through explicit supervision often fail to generalize to unseen damage scenarios. A few unsupervised anomaly detection approaches have also been developed with a focus on autoencoders [149, 15, 150] and principal component analysis (PCA) [151, 152, 126, 153]. Besides detection, a limited number of approaches focused on damage localization with FEMs [154], CNNs [155, 156], and autoencoders [129]. These existing unsupervised methods [36, 129], however, are typically designed to model data from a single sensor or do not take the domain attributes of the structure and the sensor placement into account when detecting or localizing damage.

We propose Mechanics-Informed Damage Assessment of Structures (MIDAS), a near-real-time SHM framework for automated damage detection and localization. Our solution is based on the premise that sensor data collected from a structure during its regular operation represents its expected behavior, and any deviation from this behavior indicates potential damage. A structure we wish to assess for damage is instrumented with sensors, and data from its undamaged state is collected to establish the reference (baseline) for damage detection through unsupervised learning. The established reference can be employed to detect and localize damage. This solution affords adaptation to known and unknown damage across diverse structures like gusset plates and beam-columns.

The key contribution of MIDAS is the seamless integration of inexpensive sensors, data pre-processing in the form of compression, and a customized autoencoder called Mechanics-Informed Autoencoder (MIAE). From a sensor perspective, our solution is agnostic to the sensor technology and can even employ wireless sensors [157, 158, 18, 159], which are becoming cost-effective and widely used today. These sensors are easier to install and maintain and are often self-powered, rendering them very effective for long-term monitoring. From a pre-processing perspective, we leverage the on-device data compression (edge computing) [89, 158, 159] offered by modern sensors and use a highly (temporally) compressed version of the raw sensor data. Subsequently, variations due to environmental or loading fluctuations are filtered away by the compression. Therefore, any abnormal patterns in the data are indicative of damage. From the neural network perspective, we adopt an autoencoder that learns a compact representation of the data streams from multiple sensors while incorporating the mechanical relations between their strain responses. Such a design significantly enhances the detection and localization of damage in the structure.

Figure 2.1 shows an overview of MIDAS. Damage detection is achieved by comparing the reconstruction error of the instantaneous sensor data in time windows with that of the undamaged baseline. To localize the damage, we further compute the norms of reconstruction errors at each sensor and interpolate them between the sensors. This approach does not require data from damaged structures for training, which is a significant advantage of our method, given that collecting realistic damaged data on large-scale structures is practically infeasible. Other techniques that use simulated damage scenarios are often inaccurate and impractical for real-time applications due to the constant need for re-calibration. In contrast, MIDAS relies solely on reference data to establish an intact model reference and detect damage by tracking deviations from this reference. Furthermore, with the integrated mechanical knowledge, MIAE significantly improves its performance in detecting and localizing damage early when it is minor.

2.2 Methods

Finite element analysis (FEA). The gusset plate is simulated using 3D elements (C3D8R) in ABAQUS under clamped-clamped boundary conditions at the bottom edge of the plate. The Poisson ratio and Young's modulus are 0.32 and $200GPa$, respectively. To simulate traffic loading, random loading magnitudes are applied to the top left and top right edge in both $-x$ and $-y$ directions. The loading magnitudes are periodic data generated by 100 combinations of Sine and Cosine functions.

To generate enough training data, the FEA of the undamaged plate structure is repeated with different random loads for multiple iterations. The FEA model uses a fixed timestep of 0.025s. In the case of damaged structures, random cracks are introduced within the plate geometry, varying in location, length (l), width (w), and angle (α). We varied the crack width from $0.1cm$ to $0.5cm$ across different cases, with an interval of $0.1cm$. And we introduced the crack at an angle of 0, 30, 45, 60, and 90°. The mesh size is set to $0.2cm$ for the crack area and $1cm$ for all other regions. The strain responses are obtained by averaging the values across all elements within the specified sensor regions. The strain data in the y direction are recorded at every timestep, with an interval of 0.025s. To analyze the temperature effect, the expansion coefficient of the structure is set to $11\times10^{-6}\degree C^{-1}$. The default initial temperature is set to $25\degree C$. The training data is generated at different temperatures varying from 5 to $30\degree C$, with an interval of $5\degree C$.

Laboratory experiment setup. Young's modulus of the steel plate is unknown for the laboratory experiment of the gusset plate. The strain gauge type is 1-LY11-6/350 and is attached vertically to measure the strain in the vertical direction, aligning with the vertically applied loading. The clamped-clamped boundary conditions of the plate are considered due to their higher controllability in an experimental setup compared to other types of boundary conditions, such as pinned-pinned or pinned-clamped.

For the beam and column structure, the experimental setup is intended to test the behavior of a beam-column connection with a supporting prop under loading. A moment connection is established between a $W4 \times 13$ I-beam and a $W4 \times 13$ column, both made of A992 grade steel. The beam, measuring 40 inches from the column face to its end, is connected to the column using two $L4 \times 4 \times 1/2$ web cleats made of A992 grade steel. The cleats are bolted to the beam flange and column flange using two 1/2 inch diameter A325 bolts per cleat, with bolt holes positioned 1 inch from each edge. The spacing between bolts and the edge distance satisfies the minimum edge distance and spacing requirements specified by the AISC manual. The column is connected at the bottom to a circular plate of 36-inch diameter by 3/16-inch fillet welds while it is supported by a $W4 \times 7.7$ I-section prop at the top by a 1/4-inch fillet weld. The base plate is anchored to the foundation using four anchor bolts. The loading profile is similar to the gusset plate experiment and controlled with a maximum displacement of 0.23 inches, corresponding to a maximum load of approximately 2000 lbs.

We generated a continuous randomly simulated traffic effect loading profile in both experiments with a time step of $0.1s$. Displacement-control testing was performed using an MTS loading frame model, applying the loading at the top and bottom fixtures. The strain sensors were connected to a NI-9236 strain input module for strain responses monitored during the loading stage, and we collected the raw strain data through LabVIEW. For data compression, we selected seven threshold levels ranging from 30 to 175 $\mu\varepsilon$ with an increment of 24 $\mu\varepsilon$.

Data compression and dataset construction. In this study, it is assumed that N sensors have been affixed to the structure of interest at N locations. During normal operation, the structure experiences continuous loading forces of unknown magnitude. Each sensor S_i continuously measures a strain signal ε_i over time (where $i = 1, 2, ..., N$).

The data reduction approach is mainly adopted from [12, 17] to solve significant data problems typically generated from structural monitoring sensors (see Fig. 2.2a). The approach can be summarized as follows: (i) predefining several strain thresholds based on the overall strain events, (ii) calculating the cumulative times for a selected segment of strain-time responses for all levels of the threshold, (iii) fitting the cumulative time data to the Gaussian equation 2.1 for data compression, and (iv) obtaining the parameters for Gaussian cumulative density function (CDF) through equation 2.1.

$$F_{Guassian}(\varepsilon) = \frac{A}{2}\left[1 - erf\left(\frac{\varepsilon - \mu}{\sigma\sqrt{2}}\right)\right] \tag{2.1}$$

where A is the summation of all cumulative time events. μ and σ represent the mean and standard deviation of the cumulative density function, and erf denotes the Gauss error function. To determine the thresholds in Fig. 2.2a, the mean strain value ε_{mean} is computed by averaging the strain responses collected from all sensors in the undamaged structure. Then, the seven threshold values are evenly distributed between $0.5\varepsilon_{mean}$ and $3\varepsilon_{mean}$. For each sensor, every 200 data points are compressed into one set of μ and σ (see Fig. 2.2b).

Next, the compressed sensor data μ and σ (see right side of Fig. 2.2b) are utilized to construct the dataset in batches. We use a moving window with length $l = 12$ and a stride of 2 to create one

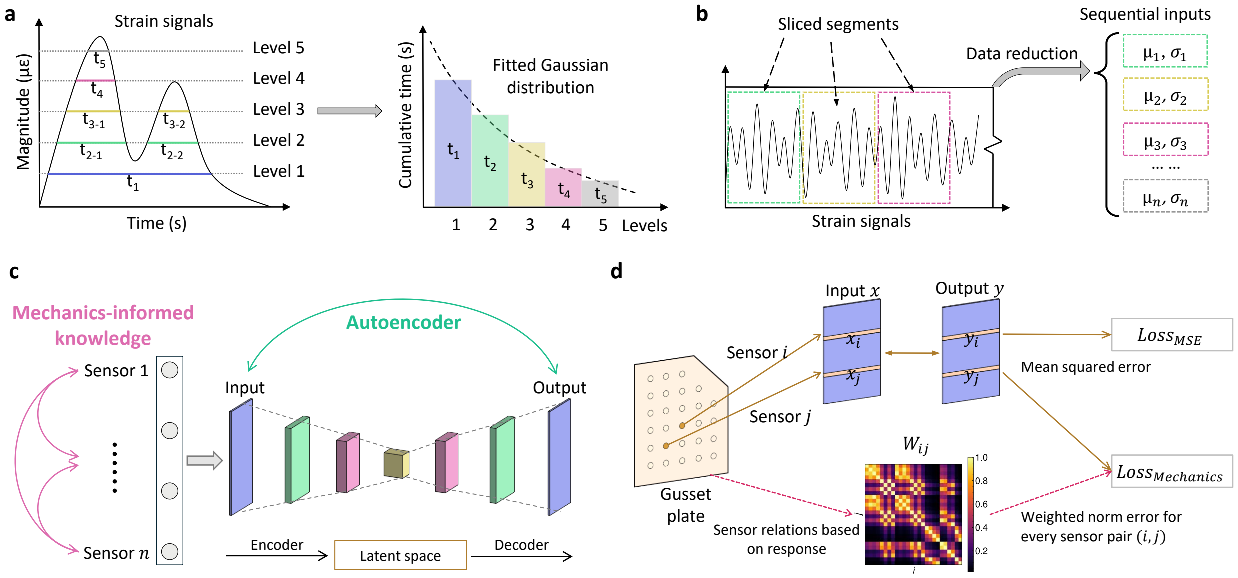


Figure 2.2 MIDAS methodology. **a**, Sensor data compression algorithm based on Gaussian distribution. Time-series sensor responses (in micro-strain) from structures are recorded in the left graph, and different threshold levels are defined based on the overall response magnitudes. Next, cumulative events above each threshold level are computed and plotted in the second graph, which is supposed to follow a Gaussian distribution. The best combination of μ and σ is obtained as the compressed sensor data through curve fitting. **b**, Sensor data processing and dataset construction. **c**, Our autoencoder architecture. **d**, The proposed loss function. The weight matrix is computed based on the strain responses from each pair of sensors. Values are shown as contours in the lower part of the graph.

batch. For example, the first training sample is taken from the 1^{st} to the 12^{th} segment, the second training sample is from the 3^{th} to the 14^{th} segment, then from the 5^{th} to the 16^{th} segment. The constructed dataset has a size of $B \times l \times 2N$ representing the number of batches, time-series data length, and the number of sensor parameters μ and σ, respectively.

Mechanica-Informed Autoencoder network. Figure 2.2c illustrates the proposed mechanics-informed autoencoder architecture with six layers. Specifically, the input and output have the same matrix size, with the output intended to reconstruct the input values. The input layer size is twice the number of sensors employed. For instance, in the case of our numerical simulation, where data from all 45 sensors is used, the input size is 90. In contrast, the size of the middle hidden layers is compacted to 32. In contrast, when using fewer sensors, such as only 4, the size of the middle hidden layer is scaled up by a factor of 8; specifically, each of the first three layers is scaled up by a factor of 2, and the last three layers are scaled down by a factor of 2. This scaling adjustment is necessary because reducing the size of the middle hidden layers beyond this point would not contribute further to model learning. The standard autoencoder has the same architecture as the MIAE for all comparisons. The first part of the loss function is computed as the mean squared error (MSE).

$$\mathcal{L}_{MSE} = \frac{1}{n} \sum (\mathbf{y} - \mathbf{x})^2 \tag{2.2}$$

where n, $\mathbf{x}$, and $\mathbf{y}$ denote the number of samples, the input, and the output from the neural network, respectively.

Most importantly, compared to standard autoencoders, MIAE utilizes mechanics-informed knowledge between sensors, leveraging a “mechanics-featured pattern”–inherent in intact structures but absent in damaged ones. This pattern is discerned by analyzing variations in data across different sensors, allowing the model to learn and recognize deviations from the baseline more effectively when damage occurs. Compared to autoencoder, the training reconstruction errors are reduced, while the reconstruction errors on other data for structures usually increased, improving MIAE’s sensitivity to subtle damage. The mechanical characteristics can be incorporated into the neural networks by considering the sensors’ mechanical responses using a weight matrix W. Specifically, the matrix has a shape of $N \times N$, and the weight elements are assigned based on corresponding sensor measurements (the largest strain values). This assignment accounts for the correlation of strain changes between two adjacent points in an undamaged structure, effectively reflecting the mechanical features such as the stress concentration effect at boundaries. When accounting for the effect of temperature, the measurements from sensors proximate to the center of the plate are scaled down to one-third before calculating the weight matrix W (as per Equation 2.5), and the corresponding λ value is reduced by half. These adjustments improve training.

Furthermore, it is essential to highlight that weights are assigned based on corresponding sensor measurements rather than relying on manual input or predefined assumptions about sensor importance. This approach properly reflects the actual mechanical features of the structure, such as stress concentrations at boundaries, whereas methods that assign weights based on geometry cannot handle them. This capability to utilize raw sensor data to automatically capture and leverage structural mechanics is a crucial aspect of our novel approach. As shown in Fig. 2.2d, the mechanical loss is evaluated for every pair of sensors i and j. The mechanics loss term $\mathcal{L}_{Mechanics}$ and the proposed loss function $\mathcal{L}$ can be calculated using the following equations.

$$\mathcal{L}_{Mechanics} = \sum_{i,j}^{N} W_{ij}(\Delta_i - \Delta_j)^2 \tag{2.3}$$

$$\Delta_i = \| \mathbf{y}_i \|_2^2 - \| \mathbf{x}_i \|_2^2 \tag{2.4}$$

$$W_{ij} = \begin{cases} max(\varepsilon_i)/max(\varepsilon_j), & \text{if } max(\varepsilon_i) < max(\varepsilon_j) \\ max(\varepsilon_j)/max(\varepsilon_i), & \text{if } max(\varepsilon_i) \geq max(\varepsilon_j) \end{cases} \tag{2.5}$$

$$\mathcal{L} = \mathcal{L}_{MSE} + \gamma \mathcal{L}_{Mechanics} \tag{2.6}$$

where Δ_i refers to the difference of norms of the input and output at sensor i, and Δ have shapes of $n \times l \times 2N$. $\mathbf{x}_i$ and $\mathbf{y}_i$ represent the corresponding input and output of the neural network from sensor i. The norm operation in equation 2.4 is computed along the temporal dimension (second dimension). W denotes the weight matrix defined based on each sensor's strain responses, with all element values less than or equal to 1. It is worth noting that W is calculated based on the original strain responses. γ is the penalty coefficient for mechanics loss term and is fine-tuned to 0.05 in this study. The proposed model, trained on equation 2.6, enhances the characteristics of structural integrity and sensitivity of model prediction. Data from the damaged structure will not follow the original mechanical features from the intact structure, resulting in poor reconstruction by the neural network and higher reconstruction errors.

Damage detection metric. After training, the model utilizes the training reconstruction errors $\hat{\Gamma}$ as a reference. It compares them to the reconstruction errors Γ at test time to identify any deviations in the samples' distribution. Assuming there are m samples from N sensors, the input, output, and reconstruction error would have $m \times N$ values. The reconstruction error Γ for each data point is calculated as:

$$\Gamma_i^j = (y_i^j - x_i^j)^2 \tag{2.7}$$

where $j = 1, 2, \ldots, m$ and $i = 1, 2, \ldots, N$.

To assess the damage detection performance, all samples Γ (size of $m \times N$) are first categorized as either anomaly (positive) or normal data (negative). This classification is accomplished by setting adaptive thresholds based on false positive rates (FPR) derived from training reconstruction errors. Next, we define a ratio q to ascertain whether a testing sample originates from a damaged structure across all m samples. Specifically, if more than $q * N$ of the N sensors were classified as anomalies, the sample is deemed to originate from a damaged structure. As a result, all m samples predict whether the structure is damaged, providing the feasibility of calculating various metrics later on. Due to limited testing data, SMOTEENN [160, 161] was employed to handle class imbalance. Subsequently, sample predictions are compared to the ground truth using binary classification metrics, including accuracy, precision, recall, F1-score, and area under the receiver operating characteristic (AUROC).

Damage localization metric. Damage can be accurately localized by comparing the obtained norm error Δ from equation 2.4 across different sensors. The objective is to summarize the damage condition at each sensor into a single scalar value, and this computation is divided into two steps. First, $\hat{\Delta}$ calculated from the undamaged data and Δ calculated from damaged data is compared with the reference, considering each sensor parameter μ or σ. This intermediate-term is denoted $\mathcal{T}$ as shown in equation 2.8, representing the relative change in reconstruction errors. Second, $\mathcal{T}$ from two types of parameters (μ and σ) are integrated as a single metric for conciseness. Therefore, the damage score p is introduced as a damage estimation metric in equation 2.9.

$$\mathcal{T} = \frac{\left|\frac{1}{m}\sum \Delta - \frac{1}{n}\sum \hat{\Delta}\right|}{\frac{1}{n}\sum \| x \|_2^2} \tag{2.8}$$

$$p = \left(\lambda \frac{\mathcal{T}^{\mu}}{max(\hat{\mathcal{T}}^{\mu})} + (1-\lambda)\frac{\mathcal{T}^{\sigma}}{max(\hat{\mathcal{T}}^{\sigma})}\right)/2 \tag{2.9}$$

where in the first equation, $\frac{1}{m}\sum \Delta$ estimates the mean value from all m testing samples, $\frac{1}{n}\sum \hat{\Delta}$ represents the mean value from all n training samples, and the denominator $\frac{1}{n}\sum \| x \|$ calculates the average of the corresponding norm. In the second equation, $\mathcal{T}^{\mu}$ and $\mathcal{T}^{\sigma}$ are vectors of relative parameter errors from parameters μ and σ based on sensors, respectively. $\hat{\mathcal{T}}^{\mu}$ and $\hat{\mathcal{T}}^{\sigma}$ are the corresponding $\hat{\mathcal{T}}$ calculated from the reference. λ is a coefficient that leverages the contribution from parameters μ and σ to damage score p. In this work, λ is set to 0.5 for all numerical simulations and experimental work. Furthermore, for damage differentiation, the evaluation for each sensor is performed separately for μ and σ, as $\frac{\mathcal{T}^{\mu}}{max(\hat{\mathcal{T}}^{\mu})}$ and $\frac{\mathcal{T}^{\sigma}}{max(\hat{\mathcal{T}}^{\sigma})}$.

The proposed estimation function can effectively differentiate the undamaged and damaged regions based on the damage scores. Specifically, a damage score p less than 1 represents the baseline or the undamaged case, while a higher score demonstrates damage around the sensor region. To some extent, the magnitude of p can indicate the damage severity at the corresponding sensor location. However, such a pattern was not consistently observed when small damage occurred. Furthermore, by incorporating the location information from all sensors and the p scores, the analysis can establish score maps for more precise structural damage localization and estimate the overall structural integrity. When using fewer sensors, the weighted centroid is computed based on the obtained p values and the corresponding sensor's position.

SPIRIT uses incremental PCA to find correlations and hidden variables that summarize the trend and signify pattern changes. The projection coefficients of the first two hidden variables (i.e., W_1 and W_2 of the PCA weight matrix W) are computed for both the training and testing datasets. The damage scores p are calculated as the element-wise Euclidean distance between the training data point ($W_{i,1}^{train}$, $W_{i,2}^{train}$) and test data point ($W_{i,1}^{test}$, $W_{i,2}^{test}$) where $i = 1, 2, \ldots, n$. The corresponding norm error Δ for SPIRIT can be calculated as the element-wise Euclidean distance using equation 2.10, while the damage score will be computed as in equation 2.9 above.

$$\Delta = \sqrt{(W_{i,1}^{train} - W_{i,1}^{test})^2 + (W_{i,2}^{train} - W_{i,2}^{test})^2} \tag{2.10}$$

2.3 Results

We evaluate the effectiveness of MIDAS in three ways: (i) numerical simulation of a gusset plate, (ii) experimental validation on a gusset plate, and (iii) experimental validation on a beam-column structure. Beyond these structures, MIDAS can be readily employed to monitor the health of other kinds of structural components.

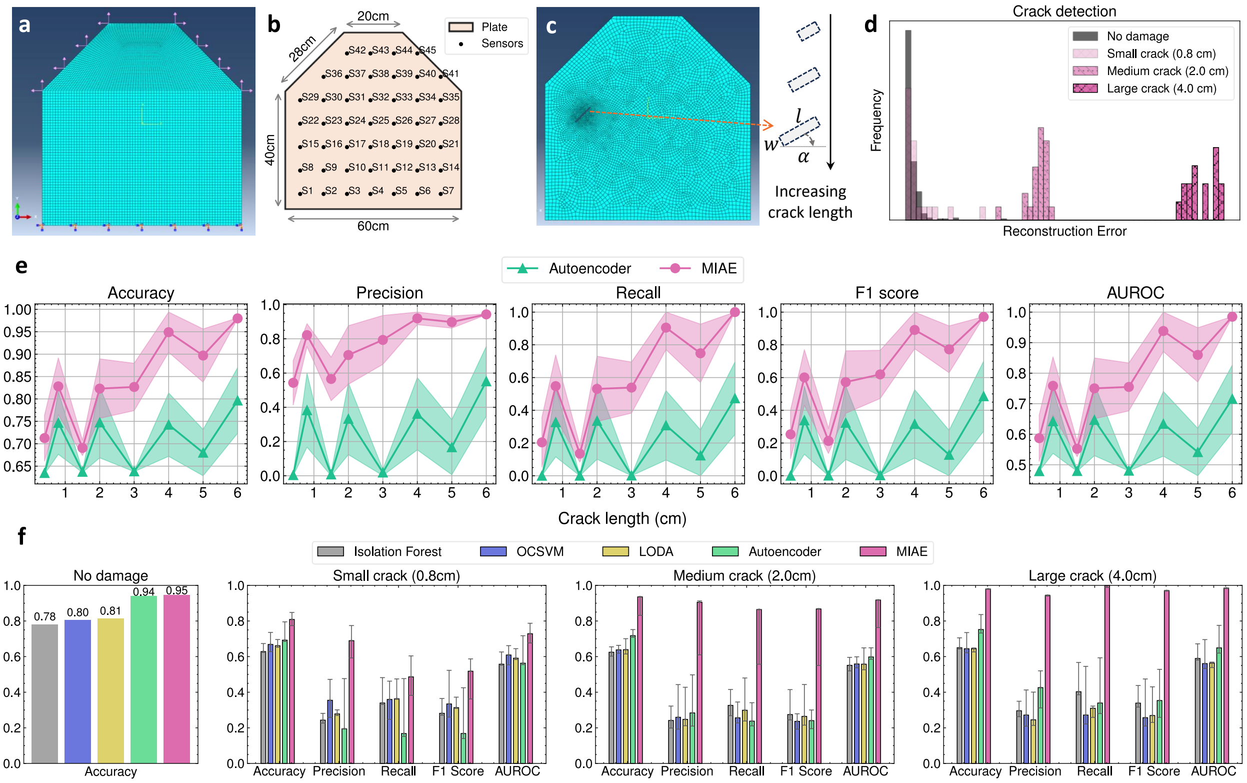


Figure 2.3 Damage detection for a cracked gusset plate. **a**. Finite element mesh of an intact plate, boundary conditions, and loading. **b**. Sensor arrangement with labels. **c**. A typical cracked plate and its meshing. Different crack lengths represent damage progression. **d**. Distributions of reconstruction errors of the structure from its undamaged reference and damaged states. As the crack progresses (three different crack lengths), the error distribution shifts to the right and becomes more distinct from the undamaged reference. **e**. Damage detection performance as the crack length increases. MIAE outperforms the baseline autoencoder in all five metrics, especially in the early stages of damage emergence. **f**. Compared to baseline anomaly detection methods, MIAE exhibits the best detection accuracy in the undamaged scenario and consistently achieves higher damage detection rates across all the evaluated metrics and crack lengths.

Numerical simulation–a gusset plate. An intact (undamaged) polygon-shaped steel plate is analyzed using finite element simulations. The mesh details are shown in Fig. 2.3a. This undamaged plate is subjected to random traffic loads to simulate the normal operations of a structural component. The detailed dimensions of the plate are shown in Fig. 2.3b (thickness is 1.2cm). Strain responses are measured at 45 points within the structure as marked in Fig. 2.3b.

Establishing reference baseline of structural behavior: Time-series data from the sensors are measured, segmented, and then compressed (more details are provided in the Methods section).

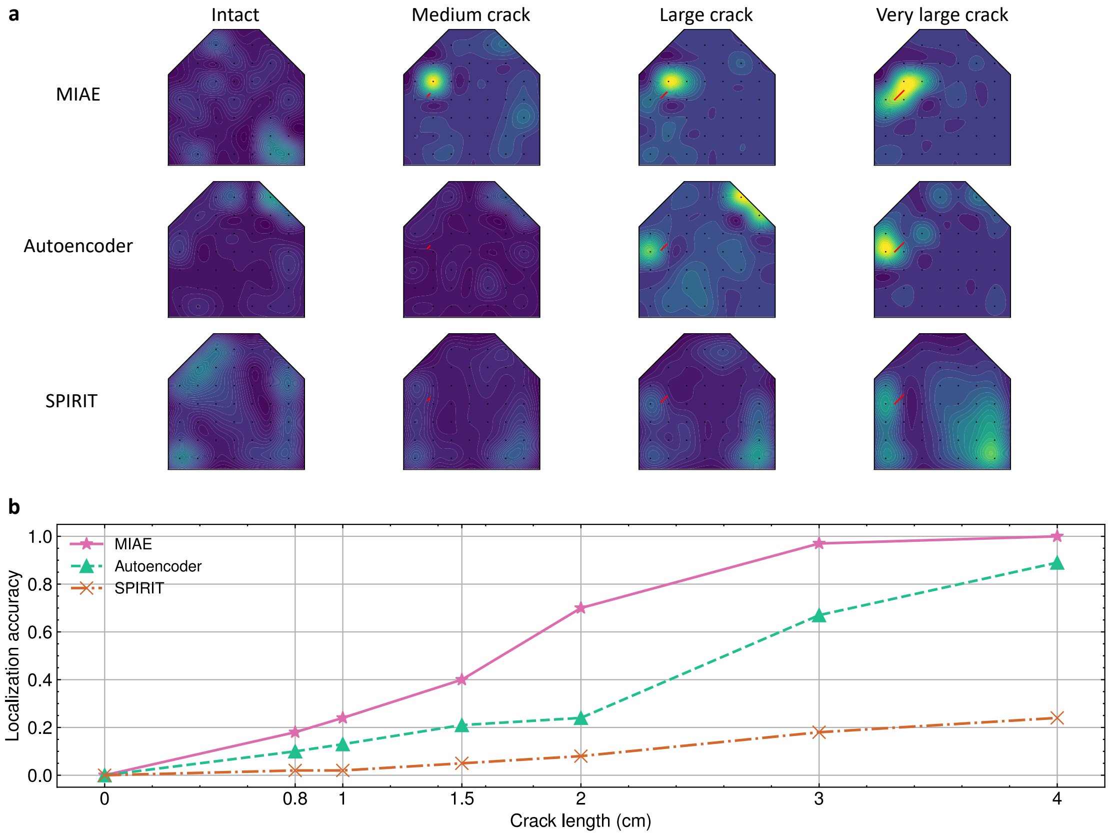


Figure 2.4 Damage localization for a cracked gusset plate. We consider different crack lengths: intact ($0cm$), medium ($2cm$), large ($4cm$), and very-large ($6cm$). MIAE localizes cracks at an earlier damage stage than prior unsupervised methods. **a**. Damage score maps for different damage scenarios. A high damage score (peak values in yellow) at one or more sensors near the crack indicates successful localization. MIAE can localize the crack earlier (at a small crack length) than SPIRIT and autoencoder. **b**. Damage localization accuracy from an extensive analysis of 37 different crack scenarios. The y-axis refers to the percentage of cases where damage was successfully localized. Compared to autoencoder and SPIRIT, MIAE has a higher localization accuracy across all crack lengths (e.g., 35% better localization for $2cm$ long cracks), demonstrating its ability to localize cracks earlier than the baseline approaches.

The compressed data consists of the running mean μ and standard deviation σ for each sensor. Subsequently, the MIAE utilizes this data for training by seeking to reconstruct the input compressed sensor data. The trained network computes a reference for reconstruction errors, which involves the MSE between the input and output for each sensor. This reference is the intact structure's baseline, representing the undamaged structural condition.

Damage detection evaluation: We randomly introduced cracks at various locations in the FEM. To simulate different cracks in each damage scenario, we increase the crack length l from $0.4cm$

to $6cm$ while keeping the crack width w and orientation (angle) α fixed. An example of the damaged plate and the corresponding mesh is illustrated in Fig. 2.3c ($w = 0.4cm$ and $\alpha = 30°$). For evaluation, compressed sensor data is now obtained from the damaged plate over small time windows and processed by the trained model to obtain their reconstruction errors. These errors are compared to the reference reconstruction errors from baseline behavior to identify damage in the structure. The distribution of the reconstruction errors can reveal how closely the response behavior of the damaged structure resembles the original undamaged system. Additionally, we limit the number of anomaly data samples to 20 or fewer to demonstrate MIDAS's rapid damage detection capabilities. Such a capability allows MIDAS to be efficient, effective, and deployable in real-world applications for near-real-time damage detection.

Figure 2.3d presents reconstruction error histograms comparing the undamaged baseline to the damaged structure with varying crack lengths. For small damage ($0.8cm$), the reconstruction errors overlap with the reference undamaged reconstruction errors with only minor separation in the distributions, suggesting a similarity in structural response behavior. As the damage grows to a crack of length $2cm$, noticeable differences emerge between the two reconstruction error distributions. These disparities indicate that the model cannot accurately reconstruct the sensor data due to the distribution shift and can thus detect the damage. Furthermore, as the crack length increases to $4cm$, the distribution of reconstruction errors for data from the damage shifts towards higher magnitudes. Therefore, damage to the structure can be easily detected in this case.

We also evaluated the proposed MIAE against a standard autoencoder w.r.t. a range of metrics, including accuracy, precision, F1-score, and AUROC [162] (detailed information on computing these metrics is provided in the Methods section). Figure 2.3e reveals that MIAE outperforms the autoencoders in all five metrics across a wide range of crack lengths. Crucially, MIAE exhibits significant improvement in detection performance when the crack is minor (before $2cm$), which is highly desirable for early detection in real applications, especially on fracture-critical structural components that typically lack a baseline model and exhibit large behavioral differences even among similarly designed components.

Detection performance comparison against other ML methods: We compare MIAE with four baseline methods: Isolation Forest [163], One-Class support vector machines (OCSVM), LODA [164], and autoencoders using sensor data from small ($0.8cm$), medium ($2cm$), and large ($4cm$) crack lengths, across 37 cases with cracks at different locations and widths. The results are shown in Fig. 2.3f. MIAE consistently surpasses all other methods in accuracy, recall, F1-score, and AUROC. Compared to standard autoencoders, the incorporated mechanical knowledge in MIAE significantly improves damage detection performance, particularly for small cracks.

Damage localization evaluation: Apart from detecting damage, another critical desideratum of SHM is localizing the damage on the structure. MIAE demonstrates robust localization ability, even when the damage is relatively small. Unlike the detection process, which involves comparing reconstruction errors from all the sensors, localization is performed by computing the norms of reconstruction errors at each sensor to obtain a damage score (see Method section for details). A high score indicates the presence of damage adjacent to that sensor. To localize the damage more precisely, we interpolate the scores between the sensors and identify the peak score location.

Figure 2.4a shows the damage localization heatmaps for different crack lengths and the exact damage location (red line). The intact structure exhibits a uniform damage score in the first column, indicating the absence of detected damage. As a crack emerges, MIAE accurately localizes a medium-size crack ($2cm$) and a large-size crack ($4cm$), as indicated by a high damage score (yellow region). The high damage score precisely overlays the cracked region as it grows to a very large crack ($6cm$).

Here, we compare against two baseline dimensionality reduction methods, (i) SPIRIT [165, 166], which performs linear dimensionality reduction through online PCA, and (ii) a standard autoencoder that performs non-linear dimensionality reduction through a DNN. Compared to the baselines (second and third row of Fig. 2.4a for SPIRIT and autoencoder, respectively), MIAE (non-linear dimensionality reduction with mechanical consistency) is capable of localizing damage at an earlier stage ($2cm$, second column) of crack propagation. Autoencoder can only localize the very-large crack ($6cm$, fourth column), while SPIRIT completely fails to localize the crack.

These results highlight the benefit of non-linear (autoencoder) over linear (SPIRIT) dimensionality reduction and the additional benefit afforded by incorporating mechanical constraints (MIAE).

We also evaluated damage localization accuracy for the same damage detection cases we considered earlier. Figure 2.4b reports the fraction of cases, out of 37, where the damage was successfully localized at different crack lengths. Compared to autoencoder and SPIRIT, MIAE has an overall higher success rate and around 35% better localization for medium-sized cracks ranging from 1.5 to $3cm$. Furthermore, MIAE can localize most of the cracks at a size of $3cm$ while the autoencoder still fails in many cases.

Damage detection and localization with a reduced number of sensors: So far, we evaluated the damage detection and localization performance of MIAE using all available sensors (45 in number). However, real-world applications seek to minimize the number of sensors and instead place a few sensors strategically. Therefore, we evaluate the damage detection and localization performance by varying the number of sensors. When the number of sensors is fewer than 10, they are strategically selected to ensure coverage over the plate. Otherwise, the sensors are placed randomly on the structure. To ensure reliability, we repeated the evaluation multiple times for a given sensor budget, each time with a different configuration. Figure 2.5a shows the damage detection performance for a crack size of $0.8cm$ as we vary the number of sensors. The performance of methods such as Isolation Forest, OCSVM, and LODA shows no appreciable improvement as we increase the number of sensors since they are designed to operate separately on data from each sensor. In contrast, autoencoder and MIAE are learned on data available from all sensors. They can better leverage the additional information available as we increase the number of sensors and thus gain performance. Importantly, MIAE leverages sensor correlations based on mechanics knowledge, achieving the best performance among all evaluated methods with only four sensors while getting more accurate as more sensors are available.

Figure 2.5b shows a configuration of four sensors (S9, S13, S30, and S34, marked as black dots within the localization map) utilized to localize damage from different scenarios. Compared to the standard autoencoder, MIAE achieves better localization accuracy (notice that the peak damage

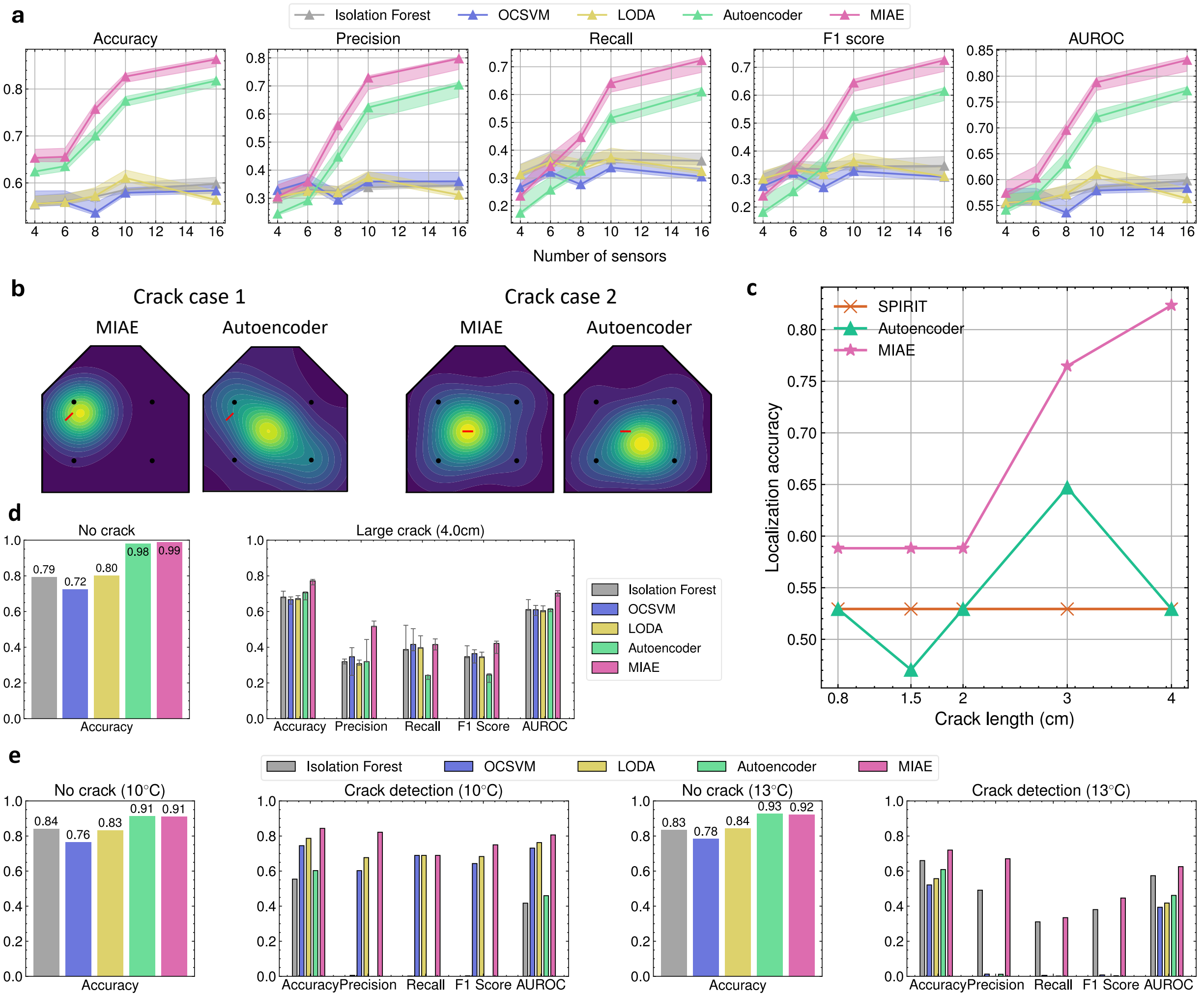


Figure 2.5 Damage detection and localization under sensor and temperature variations. **a**. Damage detection performance as the number of sensors varies. **b**. Comparison of localization accuracy between MIAE and autoencoder with four sensors for two different crack scenarios. MIAE's peak damage score is closer to the true crack location in both cases. **c**. Comparison of damage localization accuracy with four sensors as crack length increases. MIAE outperforms the baseline approaches. **d**. Damage detection performance with noisy (0.5% additive Gaussian noise) sensor data. **e**. Damage detection performance was evaluated at two different temperatures.

scores are closer to the crack). SPIRIT failed to localize damage with only four sensors, so we do not report these results. Next, we extensively analyze the localization performance as the fraction of cases correctly localized as the crack size increases. Specifically, in the 4-sensor setup, we estimate the peak damage score location as the centroid of the four sensors, which is weighted by their damage scores. In this case, we define localization as successful if the true damage is within a radius of $13cm$ (half of the sensor-to-sensor gap) around the peak location in the damage score map. As shown in Fig. 2.5c, MIAE outperforms both the autoencoder and SPIRIT, achieving around 10% to 35% better localization performance across different crack lengths. In summary, even with a limited number of sensors, MIAE exhibits excellent damage detection and localization performance.

Environmental effects consideration: Here, we explore the impact of environmental factors, such as noisy data sources and temperature variations, on structural damage assessment. Since strain sensors typically provide highly accurate measurements, a Gaussian noise level of 0.5% is introduced to the raw strain data from four sensors (S9, S13, S30, and S34). This data undergoes preprocessing (compression), and MIAE is trained on such data from the structures's undamaged state. The trained model is then evaluated using noisy sensor data under various crack scenarios. Figure 2.5d shows the testing accuracy for undamaged data and damage detection performance. Even with only four sensors, MIAE outperforms the other models when evaluated on undamaged scenarios and excels at damage detection for large cracks of length $4cm$ (can detect even smaller cracks if more sensors are used). These results underscore MIAE's robustness against noisy sensor data for detecting minor damage, i.e., at an early stage.

We analyze the temperature effect by applying different temperature environments to the structure under loading. The same sensor configuration is utilized as in the noisy data scenario. For training, data from the undamaged structure is measured at temperatures between $5°C$ and $30°C$ with intervals of $5°C$. After training, the model was evaluated at $10°C$ and $13°C$ for undamaged and damaged cases. Figure 2.5e shows damage detection performance for these configurations. Specifically, the autoencoder achieves similar performance as MIAE for undamaged cases but fails to detect damages at 10 and $13°C$. This demonstrates that incorporating mechanical strain relations between the sensors into the autoencoder increases its robustness to temperature variations.

LODA and Isolation Forest can obtain comparable recall scores during damage detection evaluation. However, their accuracy is low when evaluating undamaged samples, making these damage detection results less reliable. Overall, MIAE outperforms the other baseline methods. These numerical results provide comprehensive coverage across various scenarios, enabling the model to

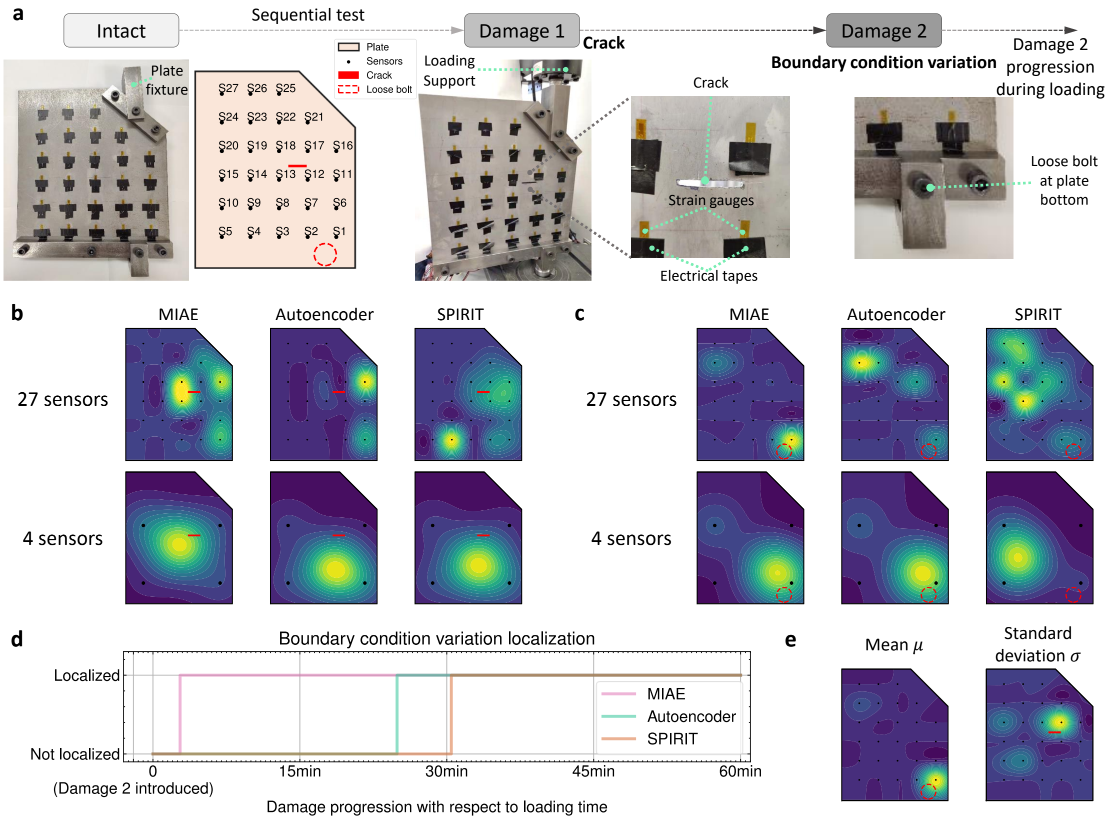


Figure 2.6 Laboratory experiment on a steel plate structure. **a**. Two types of damage are introduced sequentially (a crack and boundary condition variation). The crack is located in the middle of the plate, and the second damage was introduced by loosening the bolt connections. Under loading, the connection of the plate loosens, thus mimicking damage progression. **b**. Crack localization results with 27 and 4 sensors, respectively. When using all 27 sensors, MIAE accurately delineates the crack region with a high damage score (yellow region) around the crack tips, outperforming the autoencoder. SPIRIT fails to localize the damage in both setups. **c**. Localization for bolt loosening damage under loading. MIAE correctly localizes the damage at the bottom plate connection in the early loading stage (damage progression). **d**. Localization performance for boundary condition variation. Only MIAE can localize the damage early. As the crack size increases, both the autoencoder and SPIRIT gradually succeed in localizing it. **e**. Damage differentiation through compressed sensor data μ and σ. While μ is more sensitive to boundary condition changes, σ responds more to cracks in the structure.

distinguish actual structural damage from effects caused by unknown temperature variations, even if they are not included during training. At last, damage localization is also performed for noisy data scenarios and temperature variations, with results similar to those shown in Fig. 2.4a. We omit these results for brevity.

Experimental validation–a gusset plate. We evaluate MIDAS on a plate structure (Fig. 2.6a) to demonstrate its feasibility. The experimented steel plate measures approximately $45cm \times 36cm$. Twenty-seven (27) strain sensors were attached to the plate surface with a center-to-center gap of $6.5cm$. Random traffic-like loading is applied to the intact plate structure for 3 hours to generate enough data to train MIAE. Then, we introduced damage to the plate. To demonstrate the ability to differentiate between damage types, we sequentially evaluated two typical types of damage—cracks followed by boundary condition variations—applied on the plates during the experiment. This approach allows us to illustrate the progression of damage. Figure 2.6a shows the first damage, a crack of size $4cm \times 0.5cm$, introduced in the middle right side of the plate. The second type of damage (boundary condition variations) was subsequently introduced at the lower boundary connection of the plate. The damage was introduced by manually loosening the bolt connecting the plate to the loading frame. The bolt was loosened continuously throughout the experimental loading to mimic the progression of the boundary condition damage. In both damage states, random traffic loading was applied to the plate before and after introducing damage, and corresponding strain response data were recorded from all sensors. Data from damaged structures was evaluated similarly to the finite element simulation. Details of the sensor placement and two damage locations are shown in Fig. 2.6a.

Damage Detection and Localization: When considering the significant damage we introduced, the performance of MIAE is comparable to that of the autoencoder. However, MIAE can localize the damage more accurately than the autoencoder. Figure 2.6b presents the crack localization score maps as we vary the number of sensors. Compared to the standard autoencoder, the integrated mechanical knowledge significantly improves the damage localization accuracy. When using all available sensors, the score map exhibits a much larger peak region on both sides of the crack. This occurs because stress concentration primarily occurs at the crack's tips, and the sensors on both sides of the tip sense the structural response variations. The autoencoder score map exhibits a similar pattern, but the peak scores are much lower (faint yellow) near the right side of the crack tip, resulting in very weak localization. SPIRIT completely failed to localize the crack.

When using only four, instead of twenty-seven, sensors, MIAE had the best localization performance, with a smaller distance between the peak in the score map and the crack location than autoencoder and SPIRIT. These results suggest that our proposed method is more sensitive to minor damage, amplifying such discrepancies and improving localization over a standard autoencoder.

Figure 2.6c shows the damage score map for the boundary condition variation in the first $2min$ of loading after manually loosening the plate connections (i.e., introducing second damage). Only MIAE successfully localized the damage at the bottom right corner of the plate. This region corresponds to the actual location of the boundary variations we introduced. Autoencoder and SPIRIT exhibit worse localization performance with late localization during damage progression. Meanwhile, in the second row of Figure 2.6c, we demonstrate that accurate localization can also be achieved with fewer sensors. Figure 2.6d shows the progression of damage localization across different methods. Autoencoder and SPIRIT can only localize the damage after $20min$ of loading. This again demonstrates that MIAE exhibits higher sensitivity, enabling early damage localization after $2.5min$ of loading. Overall, our results indicate that MIAE achieves early detection and localization for different types of damage.

Damage identification: In addition to damage detection and localization, MIAE can also distinguish unseen types of damage based on the compressed sensor data features μ and σ. We independently compute the difference of norms for μ and σ for each sensor without combining them in the damage score (see Method section and Equation 2.9 for details). Figure 2.6e presents two distinct damage score maps derived from μ and σ. We observe that μ, which represents the temporal average of the strain responses, is more sensitive to boundary condition variations. This behavior is attributed to the fact that the loosening of structural connections reduces the structure's stiffness, resulting in an overall attenuation in strain magnitudes. On the other hand, the standard deviation σ is more sensitive to the cracks induced by stress concentrations. Sensors positioned near the crack tips experience elevated strain during loading, leading to a larger deviation from the baseline strain responses, i.e., increased standard deviation.

Experimental validation–a beam-column structure. Beam columns are structural elements frequently encountered in various engineering applications, such as building frames, bridges,

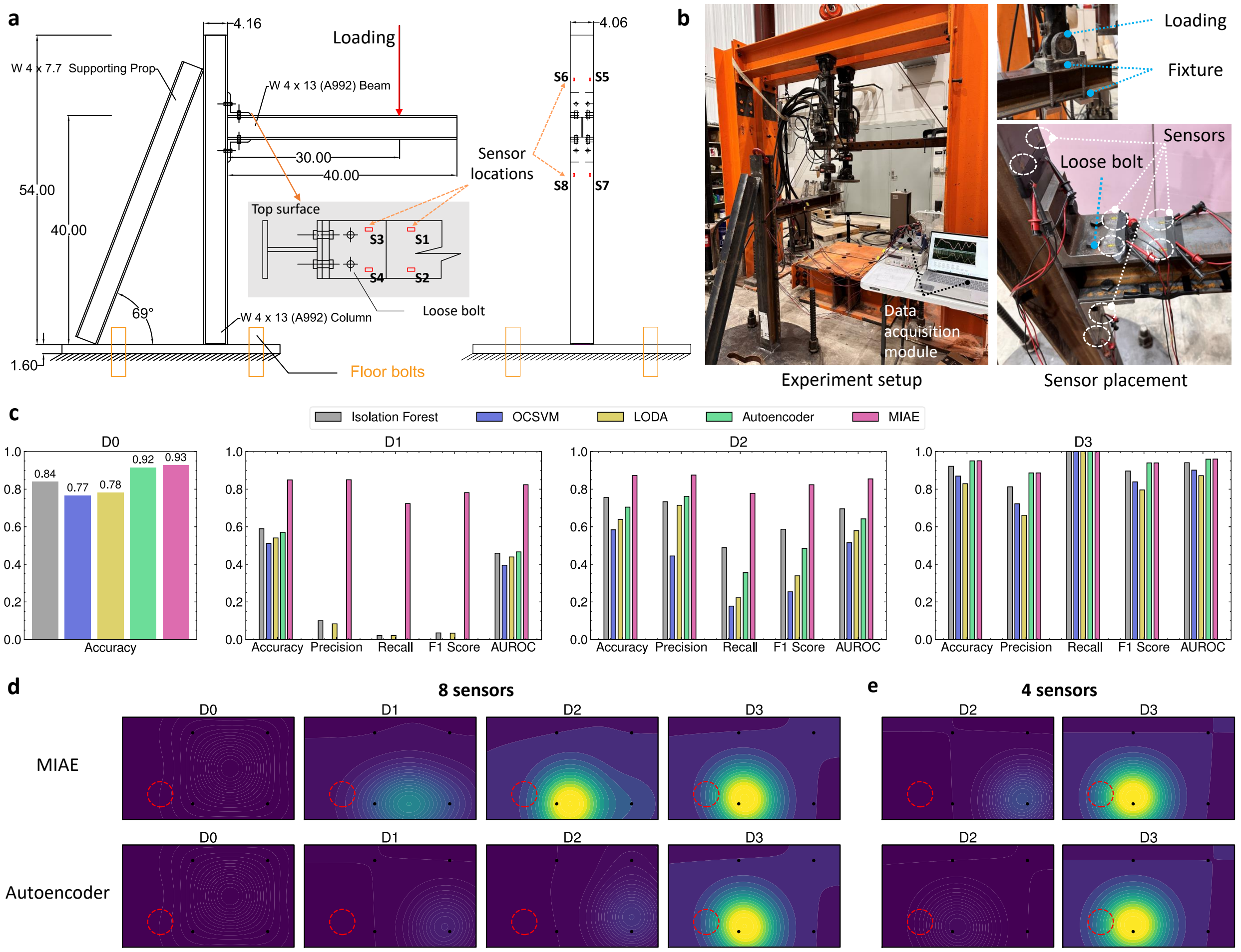


Figure 2.7 Laboratory experiment on beam-column structure. **a**. The design specifications. **b**. The experimental setup, sensor placement, and loading details. **c**. The damage detection performance at different levels of damage. **d**. Damage localization using all eight sensors' data for training. Only four sensors on the beam and support are used to generate the localization map. **e**. Damage localization with only four sensors on the support and beam for training and evaluation. Overall, MIAE can detect and localize small damage, achieving the best performance among all comparisons to the baselines.

and industrial structures. The widespread use of beam-columns in structures highlights their importance in structural engineering and the need for engineers to understand their behavior and design principles thoroughly.

We consider a structure with multiple connected components. Figure 2.7a illustrates our setup consisting of a column, beam, and other components, with units in inches. The load is applied at 3/4 of the span of the beam, 76.2*cm* (30 inches) from the column face. Strain sensors are strategically placed at the support, beam, and column flange. Figure 2.7b shows a picture of our experimental setup in the lab and the loading details.

After loading the undamaged structure (state D0), we introduced different levels of damage in the form of variations in boundary conditions (bolt loosening) during loading. The bolt near sensor four (S4) is progressively loosened from an intact state of 80 $lb \cdot ft$ to around 60 $lb \cdot ft$ for three levels of damage (D1, D2, and D3).

Damage Detection: Time-series strain data was recorded for the entire experiment and compressed for model training and evaluation. Figure 2.7c shows the detection accuracy for the intact structure. In the undamaged state D0, MIAE achieves the highest accuracy compared to other baseline methods, indicating excellent learning of the undamaged reference state. When detecting damage, MIAE demonstrates superior performance at the early stages of levels 1 and 2. At damage level 3, almost all methods can detect damage. However, methods like OCSVM and LODA achieve low accuracy when evaluating the undamaged case, making their damage detection results less reliable. Overall, the results demonstrate MIAE's ability to detect damage earlier than existing methods.

Damage Localization: We compute the damage scores at all eight sensors for different damage levels. Sensors on the beam and support (S1-S4) exhibit relatively higher damage scores compared to the other four sensors in the column, indicating damage nearby. We use only these sensors to compute damage score maps and localize the damage as they are in-plane with the beam, while the others are not. In Fig. 2.7d, when the first level of damage (D1) occurs, MIAE has its peak damage score between sensors 2 and 4, indicating potential damage. Still, it does not accurately localize the damage near sensor 4. But at damage states D2 and D3, MIAE accurately localizes the damage near sensor 4. The baseline autoencoder only localizes the damage at D3, while SPIRIT fails to localize any damage (results omitted for brevity). These results demonstrate that MIAE enhances localization for low-severity damage.

It is noteworthy that although sensors at the column are not directly used in the localization, they greatly contribute to the model training and establish the reference baseline behavior of the structure. To illustrate this, we consider a configuration using only four sensors located on the beam and the support (excluding the sensors on the column). In Fig. 2.7e, MIAE can hardly localize damage at the second damage level D2. And both MIAE and autoencoder can localize the damage at D3. Compared to successful localization at D2 when using eight sensors with MIAE, this delayed localization using fewer sensors demonstrates that additional sensors on the other structural component greatly enhance MIAE's performance.

2.4 Summary

This paper presented MIDAS for automated detection and localization of unforeseen damage, as well as the differentiation between different types of damage. MIDAS leveraged sensors positioned at various locations to gather time-series data from an intact structure, which were compressed into features at each sensor and employed for training a mechanics-informed autoencoder. The overall idea of MIDAS was to learn a reference model of strain responses from an intact structure, which aids in capturing anomalies indicative of structural damage. We demonstrated the efficacy of MIDAS through both numerical and laboratory experiments on two structures, namely, a gusset plate and a beam-column structure.

A key component of MIDAS is the Mechanics-informed autoencoder (MIAE). It leveraged the relationships between sensors based on their mechanical strain responses to enhance detection during early damage progression and enable earlier damage localization than prior methods. MIAE is sample efficient, requiring only a minimal amount of data samples for damage detection and localization. MIAE outperformed standard ML techniques like One-Class SVM, Isolation Forest, and LODA in detecting damage across different damage scenarios, achieving better accuracy, precision, recall, F-score, and AUROC. Notably, the novel loss function incorporating pairwise mechanical relations between the sensors improved the localization rate of minor damages by up to 35% over a standard autoencoder. In our laboratory experiment on a steel plate, MIDAS could also distinguish between different types of damage (boundary condition variations and cracks). Finally, the experiment on a beam and column structure demonstrated the generalization ability of MIDAS to complex structures with multiple components and different geometries.

The data compression technique used in this work has been previously developed by our research group to achieve low-cost field deployable edge computing on ultra-low-powered wireless sensors. The method has been validated in laboratory and field tests [157, 158, 18, 159]. This work's application enhances and distinctly sets our method apart from existing autoencoder-based techniques that typically process raw time-series signals directly. Incorporating data compression affords robustness to sensor noise and enables more efficient data processing, network training, and prediction, facilitating near-real-time damage detection and localization. This is extremely important for advanced wireless sensors that require efficient data storage and transmission. Overall, this technology underpins the practicality of our approach in real-world applications, contributing to an efficient and automated SHM solution.

We demonstrated the utility of MIDAS as a SHM framework for near-real-time detection and localization of structural damage. We evaluated it across various numerical and laboratory experiments, including gusset plate structures and a large-scale beam-column structure involving multiple connected components. An exciting direction of future work can focus on scaling MIDAS to even larger-scale structures (e.g., entire bridge or building). This would necessitate optimizing sensor placement, using a heterogeneous suite of sensors, and adapting the mechanics correlations for larger structures and different types of sensors.

CHAPTER 3

STRUCTURAL PARAMETER IDENTIFICATION IN NONLINEAR DYNAMICAL SYSTEMS

3.1 Overview

Structural-system identification (SI) [167, 47, 168, 84, 169, 170] refers to methods for inverse calculation of structural systems using data to calibrate a mathematical or digital model. The calibrated models are then used to either estimate or predict the future performance of structural systems and, eventually, their remaining useful life. Non-linear structural systems with spatial and temporal variations present a particular challenge for most inverse identification methods [86, 171, 172]. In dynamic analysis of civil structural systems, prior research efforts primarily focused on matching experimental data with either mechanistic models (i.e., known mechanical models) [48, 49] or with black box models with only input/output information (i.e., purely data-driven approaches), [50, 51, 52]. Examples of these approaches include eigensystem identification algorithms[53], frequency domain decomposition [54], stochastic optimization techniques [55], and sparse identification[56]. A majority of these approaches, however, fail to capture highly non-linear behaviors.

In this chapter, we propose a framework (NeuralSI) for nonlinear dynamic system identification that allows us to discover the unknown parameters of PDEs from measured sensing data. We consider the class of nonlinear structural problems with unknown spatially distributed parameters (see Figure 3.1 for an overview). The parameters correspond to geometric and material variations and energy dissipation mechanisms, which could be due to damping or other system imperfections that are not typically captured in designs.

As an instance of this problem class, we consider forced vibration responses in beams with spatially varying parameters. The primary challenges in such problems arise from the spatially variable nature of the properties and the distributed energy dissipation. This is typical for built civil structures, where energy dissipation and other hard-to-model phenomena physically drive the dynamic response behavior. In addition, it is very common to have structural systems with unknown strength distributions, which can be driven by geometric non-linearities or indiscernible/hidden material weaknesses. Finally, a typical challenge in structural systems is the rarity of measured data, especially for extreme loading cases.

The developed model performance is compared to conventional PINN methods and direct regression models. Upon estimating the unknown system parameters, we apply them to the differential model and efficiently prognosticate the time evolution of the structural response. We also investigate the performance of NeuralSI under a limited training data regime across different input beam loading conditions. This replicates the expected challenges in monitoring real structures with limited sensors and sampling capabilities. NeuralSI contributes to the fields of NeuralPDEs, structural identification, and health monitoring:

1. NeuralSI allows us to learn unknown parameters of fundamental governing dynamics of structural systems expressed in the form of PDEs.

2. We demonstrate the utility of NeuralSI by modeling the vibrations of nonlinear beams with unknown parameters. Experimental results demonstrate that NeuralSI achieves two to three orders of magnitude lower error in predicting displacement distributions in comparison to PINN-based baselines.

3. We also demonstrate the utility of NeuralSI in temporally extrapolating displacement distribution predictions well beyond the training data measurements. Experimental results demonstrate that NeuralSI achieves four to five orders of magnitude lower error compared to PINN-based baselines.

3.2 Structural problem

Many physical processes in engineering can be described as fourth-order time-dependent partial differential problems. Examples include the Cahn-Hilliard type equations in Chemical Engineering, the Boussinesq equation in geotechnical engineering, the biharmonic systems in continuum mechanics, the Kuramoto-Sivashinsky equation in diffusion systems [173] and the Euler-Bernoulli equation considered as an example case study in this chapter. The Euler-Bernoulli beam equation

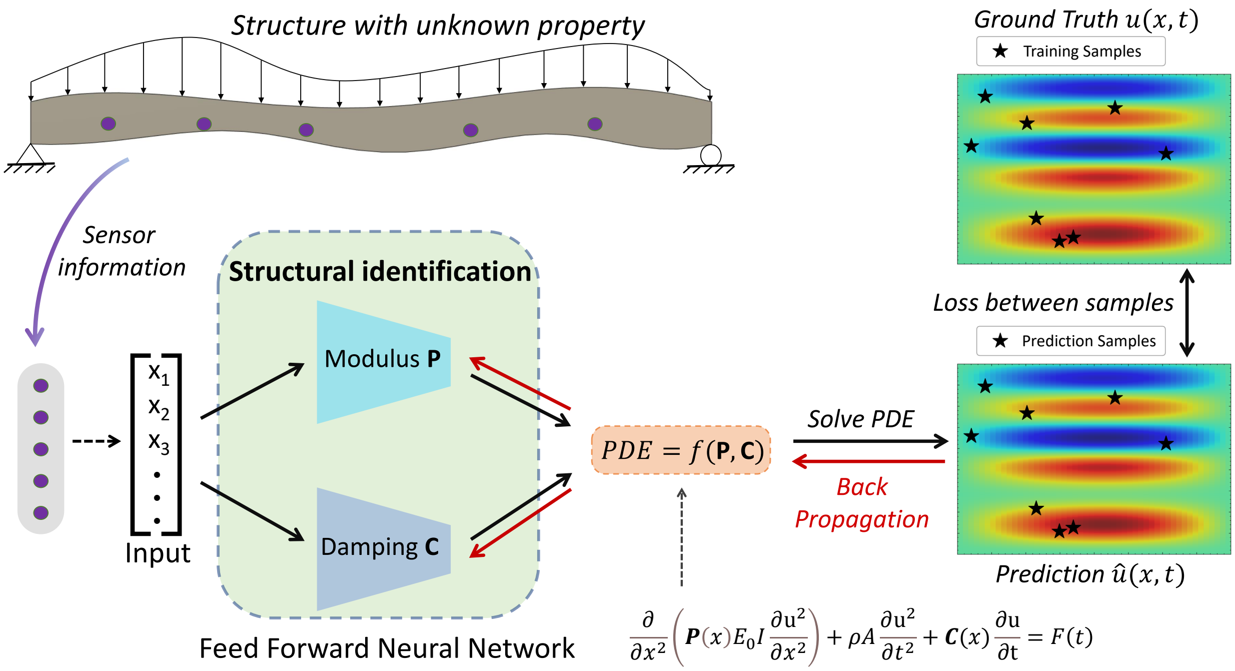

Structure with unknown property
Sensor information
Input
x_1
x_2
x_3
Structural identification
Modulus **P**
Damping **C**
Feed Forward Neural Network
$PDE = f(\mathbf{P}, \mathbf{C})$
Solve PDE
Back Propagation
Ground Truth $u(x,t)$
★ Training Samples
Loss between samples
★ Prediction Samples
Prediction $\hat{u}(x,t)$
$\frac{\partial}{\partial x^2}\left(\boldsymbol{P}(x)E_0 I \frac{\partial u^2}{\partial x^2}\right) + \rho A \frac{\partial u^2}{\partial t^2} + \boldsymbol{C}(x)\frac{\partial u}{\partial t} = F(t)$

Figure 3.1 Overview of the framework. We consider structures whose dynamics are governed by a known PDE, but with unknown parameters that potentially vary in both space and time. These unknown parameters are modeled with neural networks, which are then embedded within the PDE. In this illustration, the unknown parameters, modulus P and damping C, vary spatially. The network weights are learned by solving the PDE to obtain the structural response (deflection in this case) and propagating the error between the predicted response and the measured ground truth response through the PDE solve and the neural networks.

is widely used in civil engineering to estimate the strength and deflection of beam structures. The dynamic beam response is defined by:

$$F(t) = \frac{\partial^2}{\partial x^2}\left(P(x)E_0 I \frac{\partial^2 u}{\partial x^2}\right) + \rho A \frac{\partial^2 u}{\partial t^2} + C(x)\frac{\partial u}{\partial t} \tag{3.1}$$

where $u(x,t)$ is the displacement as a function of space and time. $P(x)$ and E_0 are the modulus coefficient and the reference modulus value of the beam, I, ρ, and A are refereed to the beam geometry and density. F is the distributed force applied to the beam. $C(x)$ represents damping, which is related to energy dissipation in the structure. In this work, we restrict ourselves only to spatial variation of the beam's properties and leave the most generalized case with variations in space and time of all variables for a future study.

The fourth-order derivative of the spatial variable and the second-order derivative of time describes the relation between the beam deflection and the load on the beam [174]. Figure 3.2.

shows an illustration of the beam problem considered here, with the deflection $u(x,t)$ as the physical response of interest. The problem can also be formulated as a function of moments, stresses, or strains. The deflection formulation presents the highest order differentiation in the PDE. This was selected to allow for the flexibility of the solution to be extended to other applications beyond structural engineering.

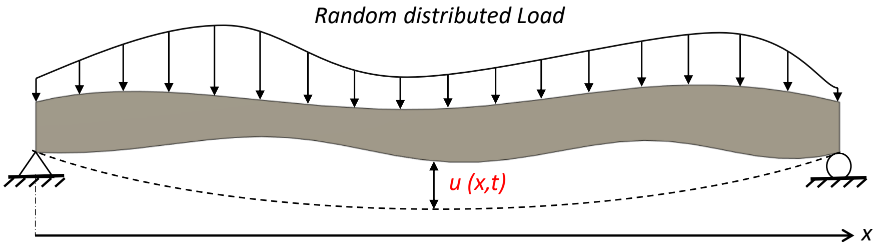


Figure 3.2 Simply supported dynamic beam bending problem. Dynamic load can be applied to the structure with its values changing in time. The geometry, modulus, and other properties of the beam can also vary spatially with x. The deflection of the beam is defined as $u(x,t)$.

To accurately represent the behavior of a structural component, its properties need to be identified. Though the beam geometry is straightforward to measure, the material property and damping coefficient are hard to estimate. The beam reference modulus E_0 is expected to have an estimated range based on the choice of material (e.g., steel, aluminum, composites, etc.) but unforeseen weaknesses in the build conditions can introduce unexpected nonlinear behavior. One of the objectives of this work is to capture this indiscernible randomness from response measurements. In addition, as discussed above, the damping is unpredictable at the design stage and is usually calculated by experiments. For the simply supported beam problem, the boundary conditions are defined as:

$$\begin{cases} u(x=0,t)=0; \qquad u(x=L,t)=0 \\ \dfrac{\partial^2 u(x=0,t)}{\partial x^2}=0; \quad \dfrac{\partial^2 u(x=L,t)}{\partial x^2}=0 \end{cases} \tag{3.2}$$

where L is the length of the beam. Initially, the beam is static and stable, so the initial conditions of the beam are:

$$\begin{cases} u(x,t=0)=0 \\ \dfrac{\partial u(x,t=0)}{\partial t}=0 \end{cases} \tag{3.3}$$

3.3 NeuralSI framework

To tackle this high-order PDE efficiently, a numerical approach based on the method of lines is employed to discretize the spatial dimensions of the PDE. Then the system is solved as a system of ODEs. The implemented discretization for the spatial derivatives of different orders is expressed as:

$$A_4^* u/\Delta x^4 = \frac{\partial^4 u}{\partial x^4}; \quad A_3^* u/\Delta x^3 = \frac{\partial^3 u}{\partial x^3}; \quad A_2^* u/\Delta x^2 = \frac{\partial^2 u}{\partial x^2} \tag{3.4}$$

where in the fourth order discretization, A_4^* is a $N \times N$ modified band matrix (based on the boundary conditions), and the size depends on the number of elements used for the space discretization, and Δx is the distance between the adjacent elements discretized in the spatial domain. A similar principle is applied to other order derivatives.

A pictorial schematic of NeuralSI is shown in Figure 3.1. The Julia differential equation package [93] allows for very efficient computation of the gradient from the ODE solver. This makes it feasible to be used for neural network backpropagation. Thus, the ODE solver can be considered as a neural network layer after defining the ODE problem with the required fields of initial conditions, time span, and any extra parameters. Inputs to this layer can either be output from the previous network layers or directly from the training data.

The network in NeuralSI for the beam problem takes as input the location of the deformation sensors installed on the structure for continuous monitoring of its response. A series of dense layers are implemented to produce the output, which are the parameters that represent the structural characteristics. The parameters are re-inserted into the pre-defined ODE to obtain the final output, i.e., the structure's dynamic response. The loss is determined by the difference between the dynamic responses predicted by NeuralSI and those measured by the sensors (ground truth).

For experimental considerations in future lab testing, we simulate in this case a beam with length, width, and thickness respectively of $40cm$, $5cm$, and $0.5cm$. The density ρ is $2700kg/m^3$ (aluminum as base material). The force $F(t)$ is defined as a nonlinear temporal function. Considering the possible cases of polynomial or harmonic material properties variations as an example [175], we integrate the beam with a nonlinear modulus $E(x)$ as a sinusoidal function. We use a range for the modulus from $70GPa$ to $140GPa$ (again using aluminum as a base reference). The damping coefficient $C(x)$ is modeled as a ramp function. The PDE can be rewritten and expressed as:

$$\begin{aligned} F(t) &= E_0 I\Big(\frac{\partial^2 P(x)}{\partial x^2}\frac{\partial^2 u}{\partial x^2} + 2\frac{\partial P(x)}{\partial x}\frac{\partial^3 u}{\partial x^3} + P(x)\frac{\partial^4 u}{\partial x^4}\Big) + \rho A\frac{\partial^2 u}{\partial t^2} + C(x)\frac{\partial u}{\partial t} \\ &= E_0 I\Big(A_2^* P(x) A_2^* u + 2A_1^* u P(x) A_3^* u + P(x) A_4^* u\Big) + \rho A\frac{\partial^2 u}{\partial t^2} + C(x)\frac{\partial u}{\partial t} \end{aligned} \tag{3.5}$$

$$F(t) = \begin{cases} 1000 & t \leq 0.02s \\ 0 & t > 0.02s \end{cases} \tag{3.6}$$

where the estimated modulus reference E_0 is $70GPa$, and $P(x)$ and $C(x)$ are modulus coefficient and damping that can vary spatially with x. The pre-defined parameters $P_0(x)$ and $C_0(x)$ are shown in Figure 3.3.

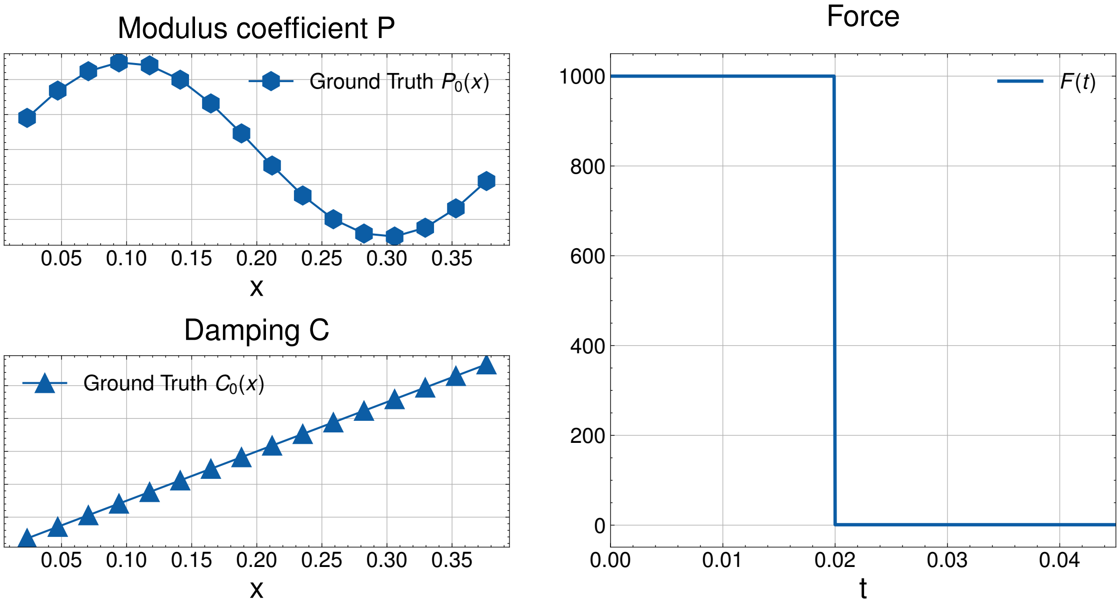


Figure 3.3 Pre-defined structural properties and resultant dynamic response. Structural parameters P and C are defined as a sinusoidal and a ramp function. Force is applied as a step function of 1000 N and reduced to zero after 0.02s.

The PDE presented in (3.5) is solved via the differential equation package in Julia. The RK4 solver method is selected for this high-order PDE. The time span was set to $0.045s$ to have 3 complete oscillations of the bending response. The number of spatial elements and time steps

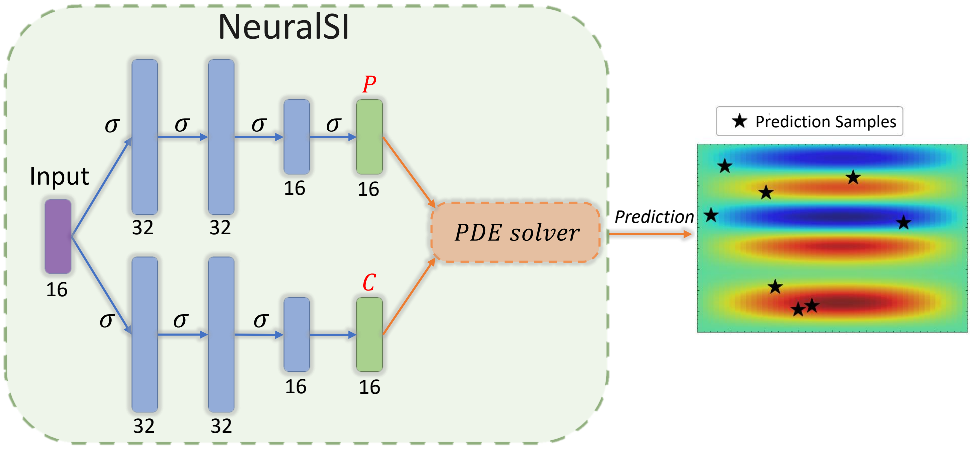


Figure 3.4 NeuralSI network architecture and training. The network has several dense layers and the output is split into P and C. Those parameters are taken to the PDE solver for structural response prediction. Samples are taken randomly from the response for training the network.

are chosen as 16 and 160 respectively for balancing the training time cost and response resolution (capture the peak deflections). The deflections $u(x,t)$ are presented as a displacement distribution of size 16 × 160, from which ground truth data is obtained for training.

The network architecture is presented as a combination of multiple dense layers and a PDE-solver layer. The input to the network is the spatial coordinates x for the measurements, and the network output is the prediction of the dynamic response $u(x,t)$. It is worth mentioning that the structural parameters P and C are produced from the multiple dense layers in separate networks, and the PDE layer takes those parameters to generate a response displacement distribution of size 16×160. The activation function for predicting the parameter P is a linear scale of the sigmoid function so that the output can be in a reasonable range. For the prediction of parameter C, the network of the same architecture is used, but the last layer does not take any activation function since the range of the damping value is unknown.

The modulus coefficient might be very high during training and lead to erroneous predictions with very high-frequency oscillations. So, we used minibatch training to escape local minima with a batch size of 16. The loss function is defined as the mean absolute error (MAE) between samples from the predicted and ground truth displacement distribution:

$$loss = \frac{1}{n}\sum_{i=1}^{n}|u - \hat{u}| \tag{3.7}$$

, where n is the number of samples for training, u and $\hat{u}$, are the values from true and prediction dynamic responses at different training points in the same minibatch.

Furthermore, inspired by the effectiveness of positional embeddings for representing spatial coordinates in transformers [176], we adopt the same as well for the spatial input to the network. It is worth noting that the temporal information in the measurements is only used as an aid for mapping and matching the predictions with the ground truth. We use ADAMW [177] as our optimizer, with a learning rate of 0.01.

3.4 Results and performance

The evaluation of NeuralSI is divided into two parts. In the first part, we evaluate predictions of the parameters P and C from the trained neural network. We assume that each structure has a unique response. To determine how well the model is predicting the parameters, Fréchet distance [178] is employed to estimate the similarity between the ground truth and predicted functions. In this case, the predicted P and C are compared to the original P_0 and, C_0 respectively.

The second part of our evaluation is the prediction of the dynamic responses, which is achieved by solving the PDE using the predicted parameters. The metric to determine the performance of the prediction is the MAE between the predicted and ground truth displacement distribution. The prediction can be extrapolated by solving the PDE for a longer time span and compared with the extrapolated ground truth. The MAE is also calculated from the extrapolated data to examine the extrapolation ability of NeuralSI. Moreover, the dynamic response can be visualized on different elements separately (i.e., separate spatial locations x) for a more fine-grained comparison of the extrapolation results.

We first trained and evaluated NeuralSI with different combinations of the number and size of dense layers, percentage of data used for training, and minibatch size. The best results were achieved by taking a minibatch size of 16, training for a total of 20 epochs, and a learning rate of 0.001 (the first 10 epochs have a learning rate of 0.01).

Figure 3.5 shows the output of modulus coefficient P and damping C from NeuralSI. For the most part, the predictions match well with the target modulus and damping, respectively. Compared

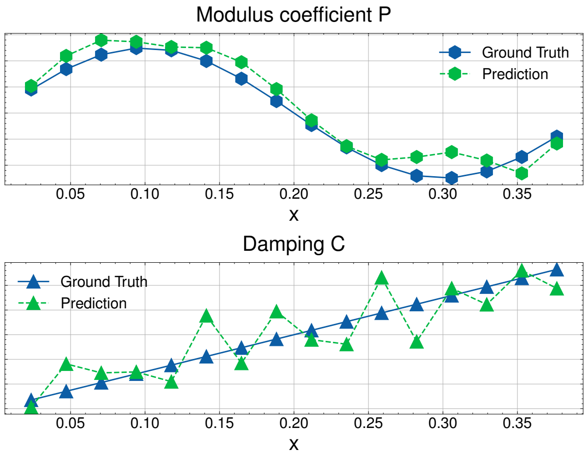


Figure 3.5 Predicted beam parameters modulus coefficient (top) and damping (bottom). Observe that the modulus coefficient P matches well with the sinusoidal ground truth, since the modulus dominates the magnitude of the response. The damping C fluctuates as it is less sensitive than P, but the outputs still present a trend of increasing damping magnitude from the left end of the beam to the right end.

to the modulus coefficient P, the predicted damping C has a larger error since it is less sensitive to the response. A small difference in damping magnitude will not affect the dynamic response as much as a change in the modulus parameter. However, the non-linearity of the modulus and damping are predicted accurately, and it is easy to identify whether the system is under-damped or over-damped based on the predicted damping parameters.

Figure 3.6 visualizes the ground truth and predicted dynamic displacement response, along with the error between the two. We observe that the maximum peak-peak value in the displacement error is only 0.3% of the ground truth. We also consider the ability of NeuralSI to extrapolate and display the dynamic response by doubling the prediction time span. It is worth mentioning that the peak error in temporal extrapolation does not increase much compared to the peak error in temporal interpolation. The extrapolation results are also examined at different elements from different locations. Figure 3.7 presents the response at the beam midspan and at quarter length. There are no observed discrepancies between the ground truth and the predicted response.

Based on the parameters chosen above, we tested the effect of a number of dense layers, training sample ratio, and minibatch size on the parameter identification and prediction of dynamic responses.

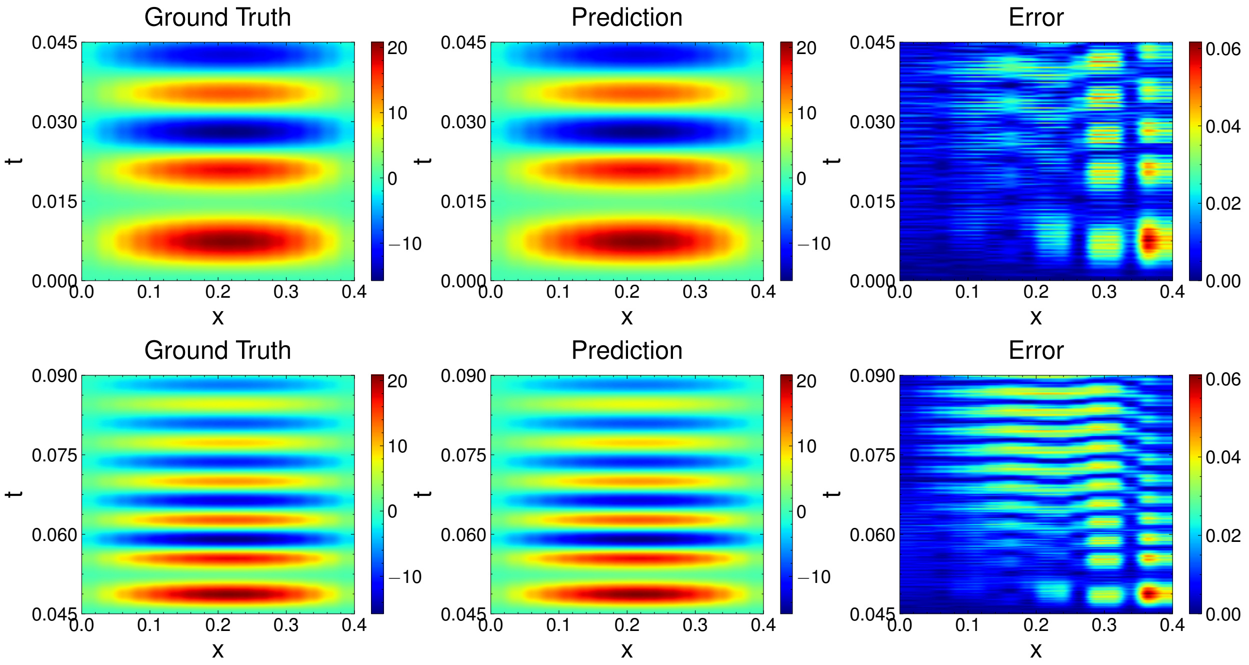


Figure 3.6 NeuralSI predictions. The interpolation results (top row) are calculated from 0 to 0.045s and temporal extrapolation results (bottom row) are from 0.045s to 0.09s. Peak error is only around 0.3% of the peak value from the ground truth, and the error magnitude remains the same for extrapolation.

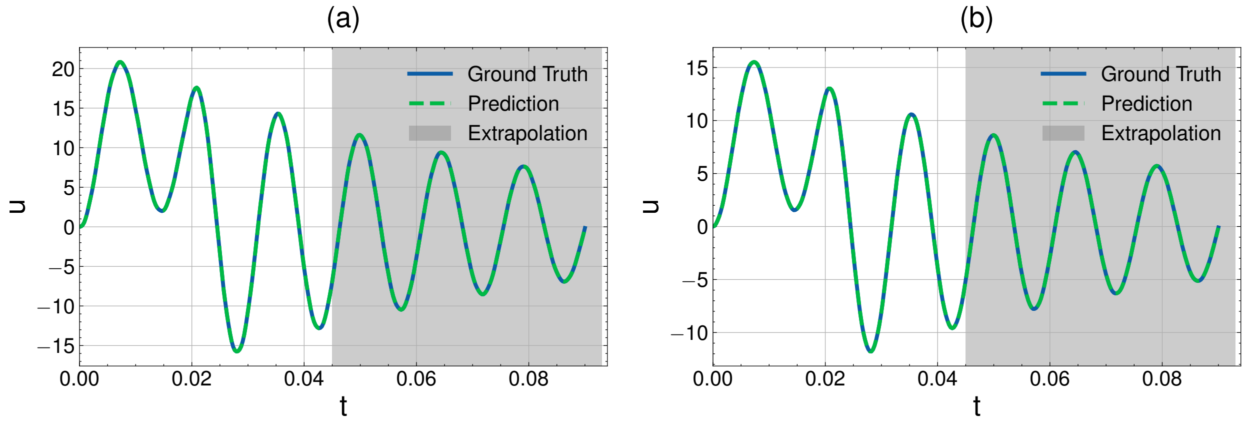
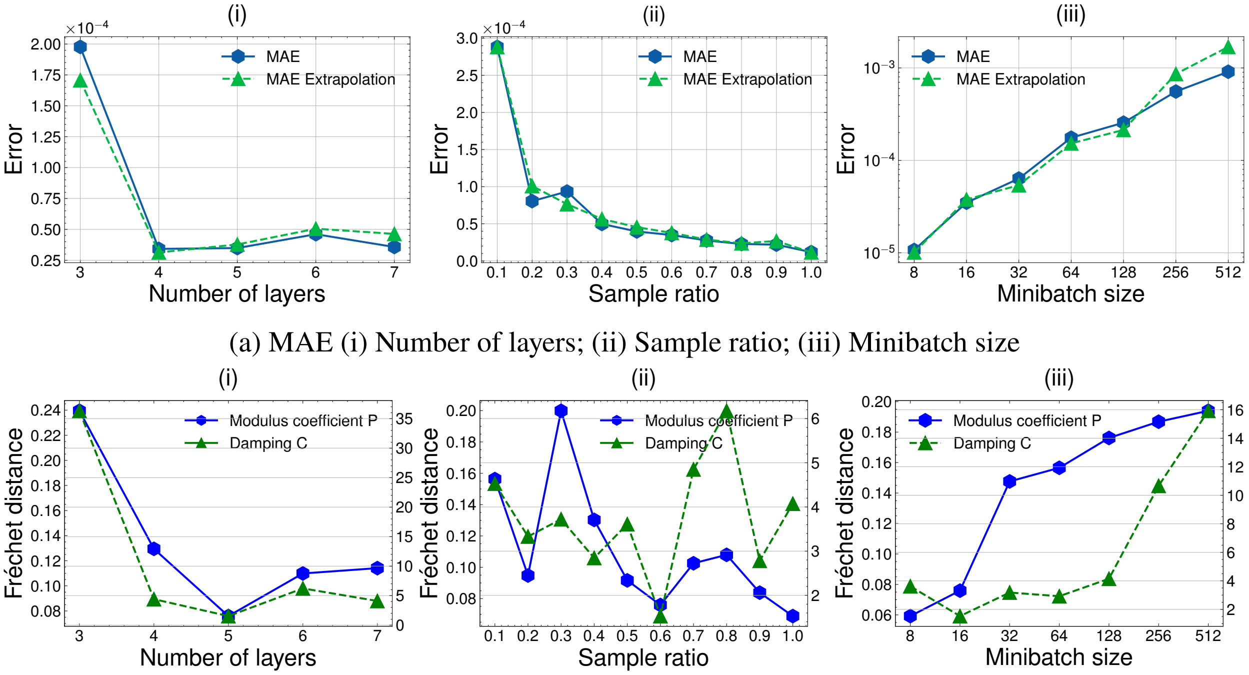


Figure 3.7 Elemental response, spatial elements from the beam are selected to examine the temporal response. The ground truth and prediction responses match perfectly. (a) element at beam midspan; (b) element at a quarter length of the beam.

The number of layers is varied by consecutively adding an extra layer with 32 hidden units right after the input. From Figure 3.8, the performance of the network is affected if the number of layers is below 4. This is explained by the fact that the network does not have sufficient capacity to precisely estimate the unknown structural parameters. It is noted that the size of the input and



(a) MAE (i) Number of layers; (ii) Sample ratio; (iii) Minibatch size

(b) Fréchet distance (i) Number of layers; (ii) Sample ratio; (iii) Minibatch size

Figure 3.8 Hyperparameter performance. A sufficient number of layers, more training samples, and a small minibatch size will produce a good combination of hyperparameters and loss MAE (top row). The Fréchet distances (bottom row) are calculated for P and C respectively. The fluctuation of Fréchet distance for different sample ratios is because the values are relatively small.

output are determined by the minibatch size and the number of elements used for discretization. A higher input or output size will automatically require a bigger network to improve prediction accuracy. Additionally, the Fréchet distance decreases as the size of the neural network increases, which demonstrates that the prediction of beam parameters is more accurate.

The number of training samples plays an important role in the model and in real in-field deployment scenarios. The number and the efficiency of sensor arrangements will be directly related to the number of samples required for accurately estimating the unknown parameters. It is expected that a reduced amount of data is sufficient to train the model given the strong domain knowledge (in the form of PDE) leveraged by NeuralSI. From Figure 3.8, when 20% of the ground truth displacement samples are used for training, the loss drops noticeably. With an increased amount of training data, the network performance can still be improved. Furthermore, observe that there is a slight effect of data overfitting when using the full amount of data for training. The

Fréchet distance of damping is not stable since our loss function optimizes for accurately predicting the dynamic deflection response, instead of directly predicting the parameters. As such, the same error could be obtained through different combinations of those parameters.

The minibatch size plays an important role in the efficiency of the training process and the performance of the estimated parameters. It is worth mentioning that a smaller minibatch size helps escape local minima and reduces errors. However, this induces a higher number of iterations for a single epoch, which is computationally expensive. From Figure 3.8 we observe that both the MAE error and the Fréchet distance are relatively low when the minibatch size is smaller than 32.

3.5 Comparison of NeuralSI with a direct response mapping DNN and a PINN

The NeuralSI framework is compared with traditional DNN and PINN methods. The tested DNN has 5 dense layers and a Tanh activation. The inputs are the spatial and temporal coordinates x and t, respectively, of the displacement response, and the output is the beam deflection $u(x,t)$ at that spatio-temporal position. The optimizer is LBFGS and the learning rate is 1.0. With a random choice of 20% samples, the loss stabilizes after 500 epochs.

The PINN method is defined with a similar strategy to existing solutions [82, 105]. The Neural network consists of 5 dense layers with Tanh activation function. The loss is defined as a weighted aggregate of the boundary condition loss (second derivative of input x at the boundaries), governing equation loss (fourth-order derivative of x and second-order derivative of the t), and loss between the prediction and ground truth displacement response. We used LBFGS as the optimizer with a learning rate of 1.0. The training was executed for 3700 epochs.

The prediction of the dynamic deformation responses for the two baseline methods and NeuralSI and the corresponding displacement distribution errors are shown in Figure 3.9. In NeuralSI, we used the ImplicitEulerExtrapolation solver for a 4× faster inference. We further optimized the PDE function with ModelingToolkit [179], which provides another 10× speedup, for a total of 40× speedup over the RK4 solver used for training. Due to a limited amount of data for training, the DNN fails to predict the response. With extra information from the boundary conditions and equation, the PINN method results in an MAE loss of 0.344, and the prediction fits the true displacement

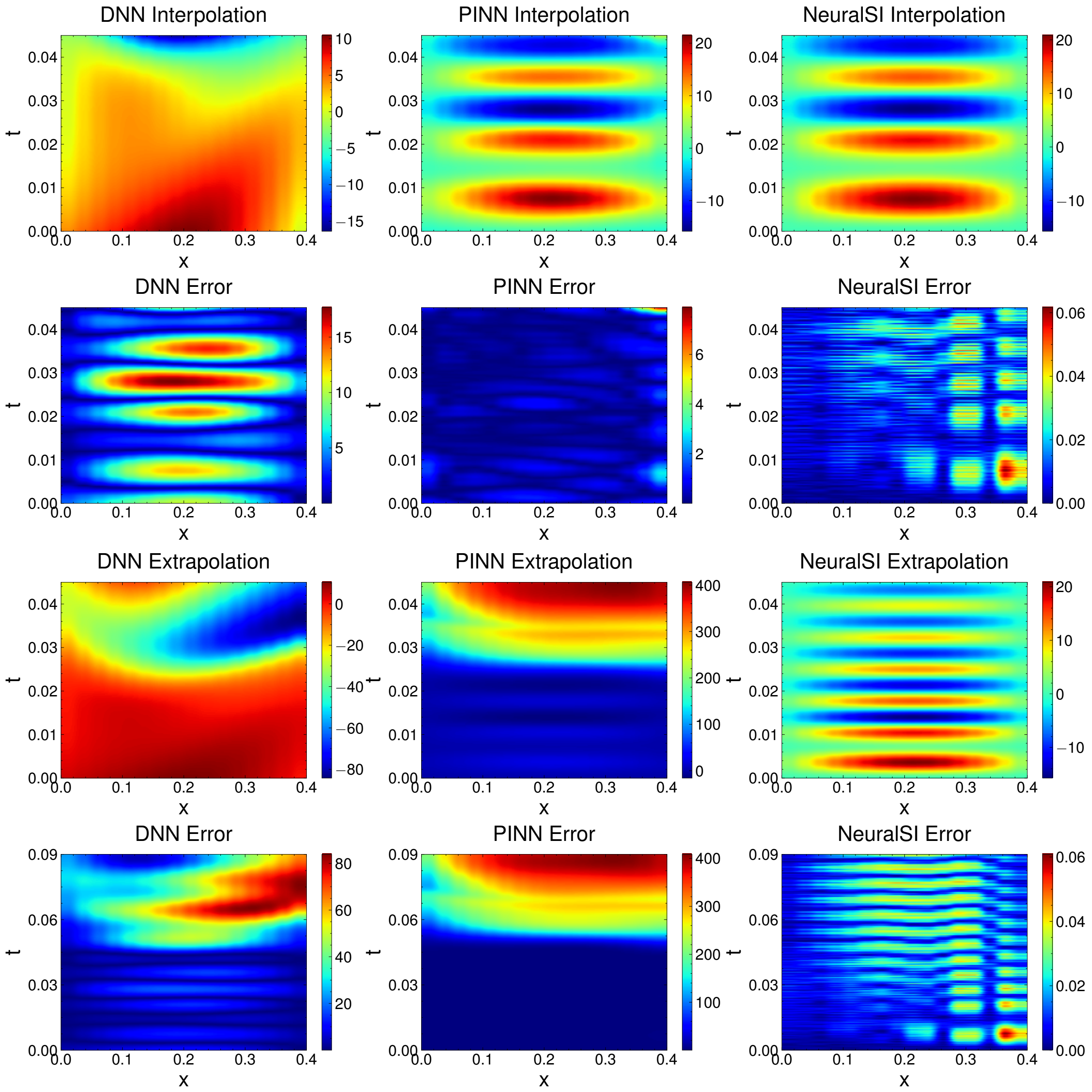


Figure 3.9 Spatio-temporal displacement distribution predictions and comparisons between DNN, PINN and NeuralSI for both interpolation (top) and extrapolation (bottom). The DNN method fails to learn the interpolation response, while the PINN can predict most of the responses correctly, with only a few errors at the corners of the displacement response. Predictions from NeuralSI have two orders of magnitude lower error in comparison to PINN. With the learned structural parameters, NeuralSI maintains the same magnitude of error in extrapolation results. Both DNN and PINN completely fail at extrapolation and lead to considerable errors.

distribution well. Most of the values in the displacement distribution error are small, except for some corners. However, both methods fail to extrapolate the structural behavior temporally. The

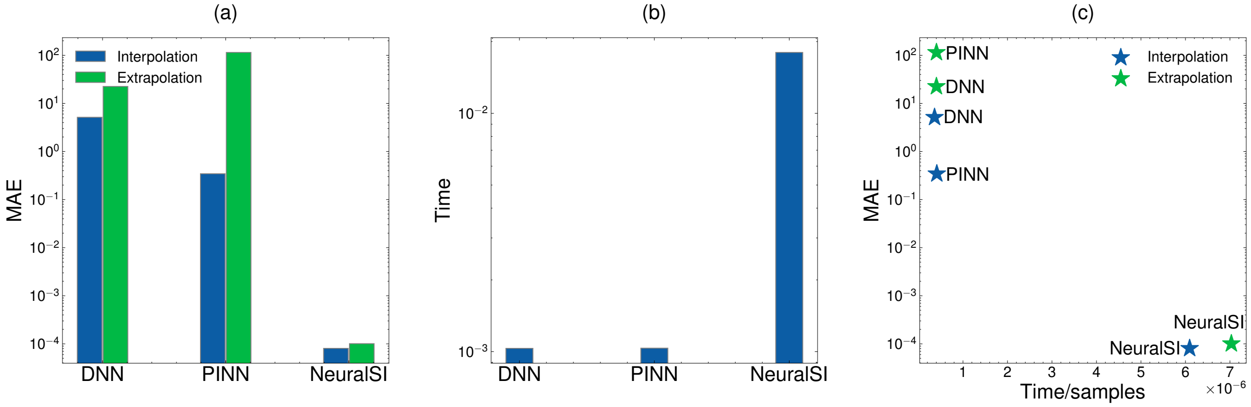


Figure 3.10 Performance comparison between DNN, PINN, and NeuralSI for both interpolation and extrapolation (a) MAE, (b) Inference time, and (c) Trade-off between MAE and inference time. NeuralSI offers significantly lower error while being as expensive as solving the original PDE, thus offering a more accurate solution when the computational cost is affordable. NeuralSI obtains the extrapolation results by solving the whole time domain starting from $t = 0$, while DNN and PINN methods directly take the spatio-temporal information and solve for extrapolation.

extrapolation of DNN predictions produces large discrepancies compared to the ground truth. Similarly, the PINN method fails to match the NeuralSI performance, while fairing much better than the predictions from the DNN, as expected due to the added domain knowledge. The MAE errors were computed and compared with the proposed method trained with 20% data as shown in Figure 3.10.

3.6 Summary

In this chapter, we proposed NeuralSI, a framework that can be employed for structural parameter identification in nonlinear dynamic systems. Our solution models the unknown parameters via a learnable neural network and embeds it within a PDE. The network is trained by minimizing the errors between predicted dynamic responses and ground truth measurement data. A major advantage of the method is its versatility and flexibility; thus, it can be successfully extended to any PDEs with high-order derivatives and nonlinear characteristics. The trained model can be used to either explore structural behavior under different initial conditions and loading scenarios, which is vital for structural modeling or to determine high-accuracy extrapolation, also essential in systems' response prognosis. An example beam vibration study case was analyzed to demonstrate the capabilities of the framework. The estimated structural parameters and the dynamic response variations match well with the ground truth (MAE of 10^{-4}). The performance of NeuralSI is also shown to outperform direct regression significantly through DNN and PINN methods by three to five orders of magnitude.

CHAPTER 4

ADVANCED STRUCTURAL PARAMETER IDENTIFICATION

4.1 Overview

Following the identification process [21] [180] [86] [181], the anticipated modal behavior should fall within an acceptable tolerance [182]. In SI, while some model information may be partially known, modal parameters need to be estimated through the calculation and processing of monitoring data. The estimated information will aid in determining or recalibrating the structural properties, predicting structural responses, and establishing a new baseline model for unknown structures [143]. However, civil structures are constructed under diverse conditions, where complex factors such as friction and other hard-to-model phenomena physically influence dynamic response behavior. Additionally, spatial variations of properties present typical challenges for built structures [86, 65, 183]. Such difficulties create a gap between theories, lab experiments, and real-life cases. Moreover, structural problems often manifest as high-order differential equations, rendering the problem stiff, hard to solve, or requiring a significant amount of time to converge.

Prior research has primarily focused on mainly two approaches. One is the model-driven technique based on assumptions of prior knowledge. Research has been implemented in model updating using finite element method [154], orthogonal diagonalization [184], and Bayesian updating [180, 185], Many other research also emphasized purely data-driven methods using sparse identification [56], eigensystem identification algorithms [186], and neural networks [78, 187]. However, a majority of these approaches fail to capture highly non-linear behaviors.

Recently, a new approach driven by physics models has emerged in the field of SI, which is based on recovering or approximating governing equations or equation parameters [86]. Existing research mainly focuses on the field of ordinary differential equations (ODEs) [94, 95], but limited research has been conducted on PDEs that are commonly used for civil infrastructures.

Traditional methods for SI typically involve mapping map external excitation to the corresponding structural response using state-space models [57, 58, 59] and sparse component analysis [60, 61, 62]. Besides, many model updating approaches [46] such as Bayesian updating [63, 64, 65] and finite element model updating [66, 67, 68, 69] has been applied to SI. Recently, data-driven methods and machine learning approaches have significantly enhanced the discovery and approximation of governing equations [70, 71, 47, 65, 72, 73]. Specifically, machine learning approaches have been widely utilized for structural system modeling and capturing nonlinear characteristics [74, 43]. Different network architectures are also implemented, such as long short-term memory (LSTM) [75, 76] and convolutional neural networks [77, 78, 79]. Additionally, research has focused on modeling with physics-informed neural networks (PINNs) [78, 80, 81], which incorporate augmented knowledge of constitutive equations (ODEs or PDEs), boundary and initial conditions [82, 83]. These conditions act as penalizing terms to constrain the solution space and provide more precise and acceptable solutions.

The emerging approach of neural differential equation [90] has gained significant traction in recent years due to its capacity to learn and capture dynamic behaviors. It has been widely used in various problems, such as in the fields of hydrology [94], fluids [95], climate models [96], chemistry [97], causal inference [92], and structures [86]. Compared to direct fitting using traditional machine learning methods, Neural ordinary differential equations (ODEs) establish a novel perspective by creating a connection between input and output variables. In addition, few studies have explored Neural PDEs using lie point symmetry data augmentation [98], PINNs [99], message passing [101], and graph neural networks [102, 103].

In addition to estimating modal parameters or model identification, research also focuses on directly utilizing PINN for structural modeling. Some studies concentrate on structural applications such as predicting responses in gusset plates [106, 107], wind turbines [108], seismic response [109], and glass structure material [110]. PINNs are also widely applied in other fields such as climate modeling [111], transportation [112], fluid mechanics [113], electromagnetic analysis [114].

This chapter focuses on parameter identification case studies of NeuralSI, especially for experimental validation of a Beam and analysis of 2D Plate Structures. Despite NeuralSI's ability to learn parameters directly through neural networks, we improve the efficiency of the parameter identification framework by using an adaptive resolution (a suitable mesh size) based on the training progress. Three important plate parameters are estimated.

4.2 Method

We consider a 2D spatiotemporal system whose governing equations can be described by a nonlinear parameterized PDE in the general form:

$$\mathcal{F}\left[u, u^2, ..., \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, ..., \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial y^2}, ..., \frac{\partial^2 u}{\partial x \partial y}, \frac{\partial^3 u}{\partial x^2 \partial y}, ...; \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial t^2}; P\right] = Q \tag{4.1}$$

where $\mathcal{F}$ is the governing PDE, P is the set of parameters that needs to be estimated in the structural problems, Q is the source term, and it is usually forced in structural problems. To efficiently solve the high-degree derivative PDEs in structural problems, a numerical approach based on the finite difference method (FDM) is employed for the discretization in either the 1D or 2D space domain. The discretized form of the equation becomes a system of ODEs. The number of equations depends on the number of points selected for discretization. The discretization approximation strategy is shown as:

$$\frac{\partial^p u}{\partial x^p} = A_p^x \mathbf{u}; \qquad \frac{\partial^q u}{\partial y^q} = \mathbf{u} A_q^y; \qquad \frac{\partial^{p+q} u}{\partial x^p \partial y^q} = A_p^x \mathbf{u} A_q^y \tag{4.2}$$

Here, A_p^x and A_q^y are $N_x \times N_y$ modified band matrices for the p^{th} and q^{th} derivatives of the discretization, where $p, q = 1, 2, 3, 4.$ and N_x and N_y are the number of discretization points in the x and y directions. On the left-hand side of those equations, u is the spatially continuous dynamic response that is taking the partial derivatives. While the $\mathbf{u}$ on the right side is the response at all selected discretization points in the space domain.

Derived from Equation 4.2, the discretized form of a generic PDE can be expressed as:

$$\mathcal{F}\left[u, u^2, ..., A_1^x u, A_2^x u, ..., A_1^y u, A_2^y u, ..., A_1^x u A_1^y, A_2^y u A_1^y, ...; \frac{\partial u}{\partial t}, \frac{\partial^2 u}{\partial t^2}; P\right] = Q \tag{4.3}$$

The PDE from Equation 4.2 is now discretized into a system of ODEs and can be solved by many ODE solvers such as Runge–Kutta methods [188] given the initial and boundary conditions.

$$u = \mathcal{R}(P) = \mathcal{R}(P_1, P_2, ...) \tag{4.4}$$

here, u is the solved dynamic response, $\mathcal{R}$ denotes the PDE solver. In many real-life cases, the solver may take many parameters $P_1, P_2, ...$ instead of a single structural parameter P.

The unknown structural parameters may not be just one type or a constant value. We consider the time-independent nonlinear properties in this study, so the structural parameters will only vary with respect to the space domain. The parameters we aim to identify can be denoted as $P(x, y)$ as in the 2D space case. We choose a feed-forward coordinate neural network that only takes one coordinate sample at a time. The input is scaled structural coordinates and outputs are the target structural properties P.

To solve for the PDE, the output of structural properties P is taken as the input parameters for the PDE solver in Equation 4.4. It is worth mentioning that each parameter p_i in the equation needs to be estimated by a separate coordinate network $\mathcal{N}_i$. The solved structural response is compared to the ground truth response for an error metric. The MAE is used for the loss function $\mathcal{L}$ in this framework.

$$\mathcal{L} = \left|\hat{u} - u\right| = \left|\mathcal{R}(P) - u\right| \tag{4.5}$$

where u and $\hat{u}$ are the ground truth and predicted responses. Moreover, instead of comparing the whole response data for errors, we utilize sparse observations by taking minibatch samples during training. This technique largely avoids local minima and improves parameter prediction in some cases. The optimizer is ADAMW for faster convergence.

The training is through back-propagating the gradient from the loss to the neural network parameters. We use the differential equation package from Julia [93] for fast forward and backward computations. In this approach, the PDE solver can be considered as a neural network layer without any parameters. The only requirement for this layer is the information on initial conditions, time span, and any extra parameters that are already determined or given.

The discretization method introduces a huge amount of computation costs, as the complexity (number of the unknown equation that needs to be solved) is $O(2N^n_D)$, where N is the number of discretization points, n_D is the spatial dimensions, and the number 2 is because both acceleration and velocity need to be solved.

For large structure components, the mesh size should be increased correspondingly. This will require much more time and larger memory for network training and backpropagation. Due to flexibility in the number of discretization points and the feature of the coordinate network, N can be relatively smaller. By training the network to learn a coarse mesh of the parameters, the parameter values in the fine mesh can be queried by feeding interpolated coordinates to the trained network. Such a process is iterative as the mesh size can be gradually increased while training the same networks until the predicted response can fit the ground truth response.

A schematic diagram is shown in Figure 4.1 and the detailed strategies are applied in Chapter 4.3. However. It is also worth mentioning that the choice of N should be cautious because of the increased numerical errors in solving the discretized PDEs. A good mesh size is more likely to be a case-dependent value rather than a constant.

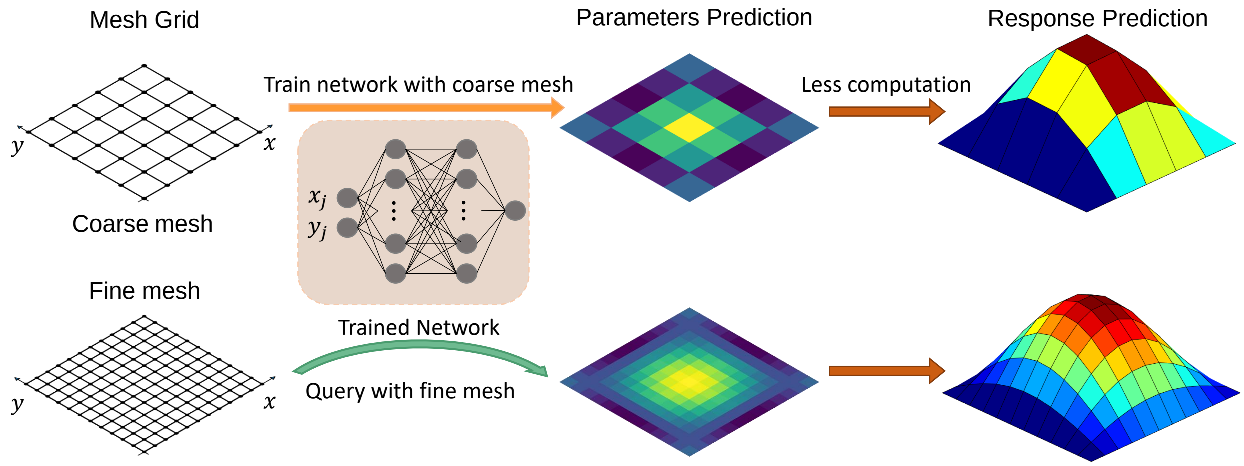


Figure 4.1 Instead of direct training to estimate parameters in a high resolution, we start the process with a coarse mesh to save on computational costs. Even though the coarse response would be less accurate, the difference is negligible and the parameter estimation errors are still acceptable. Also, based on the learned model, the mesh size can increase progressively to reach a more accurate parameter estimation.

4.3 Plate vibration problem

Plates are also one of the most important components for buildings and bridges that provide foundational support in civil infrastructures. Based on Kirchhoff's hypothesis and by placing the plate in Cartesian coordinates, we write the governing equation for the plate vibration preliminarily as the following [189]:

$$\begin{aligned} Q(x,y,t) = D_1\frac{\partial^2 u}{\partial x^4} + 2\frac{\partial D_1}{\partial x}\frac{\partial^3 u}{\partial x^3} + \big(\frac{\partial^2 D_1}{\partial x^2} + v_x\frac{\partial^2 D_2}{\partial y^2}\big)\frac{\partial^2 u}{\partial x^2} + D_2\frac{\partial^4 u}{\partial y^4} \\ +2\frac{\partial D_2}{\partial y}\frac{\partial^3 u}{\partial y^3} + \big(\frac{\partial^2 D_2}{\partial y^2} + v_y\frac{\partial^2 D_1}{\partial x^2}\big)\frac{\partial^2 u}{\partial y^2} + D_3\frac{\partial^4 u}{\partial x^2\partial y^2} \\ +2\big(v_y\frac{\partial D_1}{\partial x} + \frac{\partial D_{66}}{\partial x}\big)\frac{\partial^3 u}{\partial x\partial y^2} + 2\big(v_x\frac{\partial D_2}{\partial y} + \frac{\partial D_{66}}{\partial y}\big)\frac{\partial^3 u}{\partial x^2\partial y} \\ +\frac{\partial^2 D_{66}}{\partial x\partial y}\frac{\partial^2 u}{\partial x\partial y} + \rho h\big(\frac{\partial^2 u}{\partial t^2}\big)^2 \end{aligned} \tag{4.6}$$

where v_x and v_y are the the poison's ratio in the x and y direction. D_1, D_2 and D_{66} are orthotropic bending stiffness. h is the plate thickness and ρ is the material density. In detail, the bending stiffness D_1 and D_2 can be described by fundamental material properties respectively:

$$D_1 = \frac{E_x h^3}{12(1 - v_x v_y)} \tag{4.7}$$

$$D_2 = \frac{E_y h^3}{12(1 - v_x v_y)} \tag{4.8}$$

$$D_{66} = \frac{G_{xy} h^3}{12} \tag{4.9}$$

$$D_3 = v_x D_x + 2D_{66} = v_y D_1 + 2D_{66} \tag{4.10}$$

where E_x and E_y are the young's modulus in the x and y direction, G_{xy} is the shear modulus.

Here, the parameters we plan to estimate are the bending stiffness D_1 and D_2 and plate thickness h. This is because those parameters are the dominant factors to control the vibration of the plate, and they can vary easily in real life due to corrosion, rust, abrasion, welding, etc. Those spatial varying parameters can be characterized by the following equations:

$$D_1(x,y) = D_{10} f_{D1}(x,y) = D_{10}\big(1 + \mathcal{N}_1(x,y)\big) \tag{4.11a}$$

$$D_2(x,y) = D_{20} f_{D2}(x,y) = D_{20}\big(1 + \mathcal{N}_2(x,y)\big) \tag{4.11b}$$

$$h(x,y) = h_0 f_h(x,y) = h_0\big(1 + \mathcal{N}_3(x,y)\big) \tag{4.11c}$$

where D_{10}, D_{20}, and h_0 are constants and act as references of the target plate. f_{D1}, f_{D2}, and f_h are coefficients that measures the target parameters. $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$ are three neural network branches that take the coordinate input and output the relative parameter variations with respect to the corresponding D_{10}, D_{20}, and h_0.

It is worth mentioning that we expect to recover bending stiffness D_1 and D_2 at the high level even though they are derived from Young's modulus E_x and E_y. In this way, the potential local minima can be avoided because D_1 and D_2 are directly from the governing differential equation (equation number). The variation of some parameters is ignored because they rarely change in real life (i.e., the poison's ratio v_x and v_y in the x and y direction) or their variations won't affect the vertical deflection much (i.e., shear modulus G_{xy}, density ρ). The discretized form of the governing equation is shown as follows:

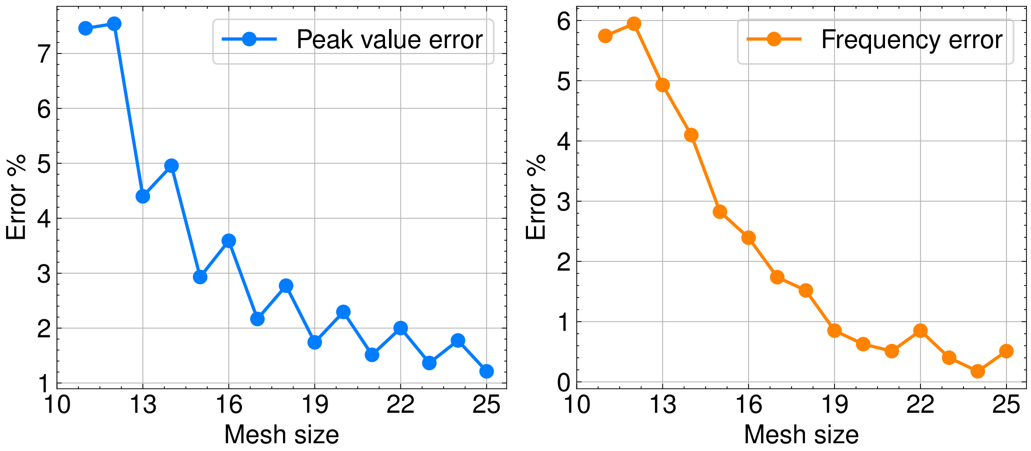
$$\begin{aligned} Q = D1 * A_4^x u + 2A_1^x D_1 * A_3^x u + (A_2^x D_1 + v_x * D_1 A_2^y) * A_2^x u + D2 * u A_4^y \\ +2D_2 A_1^y * u A_3^y + (D_2 A_2^y + v_y * A_2^x D_1) * u A_2^y + A_2^x D_3 A_2^y \\ +2(v_y * A_1^x D_1 * A_1^x D_{66}) * A_1^x u A_2^y + 2(v_x * D_2 A_1^y * D_{66} A_1^y) * A_2^x u A_1^y \\ +A_1^x D_{66} A_1^y * A_1^x u A_1^y + \rho h (\frac{\partial^2 u}{\partial t^2})^2 \end{aligned} \tag{4.12}$$

Before generating data for training, our PDE solver is first validated with the analytical solution from a homogeneous plate with the simply supported boundary condition. For a forced vibration with constant distributed load Q, the deflection at the plate center can be expressed as:

$$z_{true}(t) = A\big(1 - cos(\omega t)\big) \tag{4.13}$$

$$A = \frac{0.142 Q {L_x}^4}{E h^3 \big(2.21 + (L_y/L_x)^3\big)} \tag{4.14}$$

$$\omega = \sqrt{\frac{D}{\rho h}\bigg((\frac{\pi}{L_x})^2 + (\frac{\pi}{L_y})^2 \bigg)} \tag{4.15}$$



(a) Peak value error

(b) Frequency error

Figure 4.2 The response accuracy is examined in terms of the peak values and frequency. The calculated peak values and frequencies from different mesh sizes are compared with terms A and ω for percentage errors. When the mesh size is greater than 17, both errors are relatively small and within the 2% range.

where L_x and L_y are the side lengths of rectangle and $L_x \geq L_y$, ω is the natural frequency ([190]), and A is the response magnitude. We solved this homogeneous plate problem with the proposed PDE solver at different mesh sizes N_x and N_y and evaluated the errors with respect to the peak values and frequency. For simplicity, we set $N_x = N_y$ and $L_x = L_y$ in this verification. The error curves are shown in Figure 4.2.

Based on the 1D nonlinear properties (functions of sine, exponential, polynomial, etc.) of FGM in Figure 4.3, we expanded those 1D properties in 2D space to mimic similar properties for the plate. The 2D nonlinear property can be expressed as:

$$\psi(x, y) = \phi(x)\phi(y) \tag{4.16}$$

where $\phi(x)$ and $\phi(y)$ are 1D nonlinear functions in the x and y directions, and $\phi(x) \in \{sin(ax)$, $cos^2(ax), e^{a_1 x + a_2}, a_3x^3 + a_2x^2 + a_1x + a_0\}$, and same choices apply to $\phi(y)$. We did not consider functions such as logarithms because the curves are very similar to some of the expressions given. The example distributions used in this study are shown in the left column of Figure 4.9a and Figure 4.9b.

The plate dimension is $80 \times 80 \times 0.1cm$. The poison's ratio v_x and v_y are set to be 0.31. The

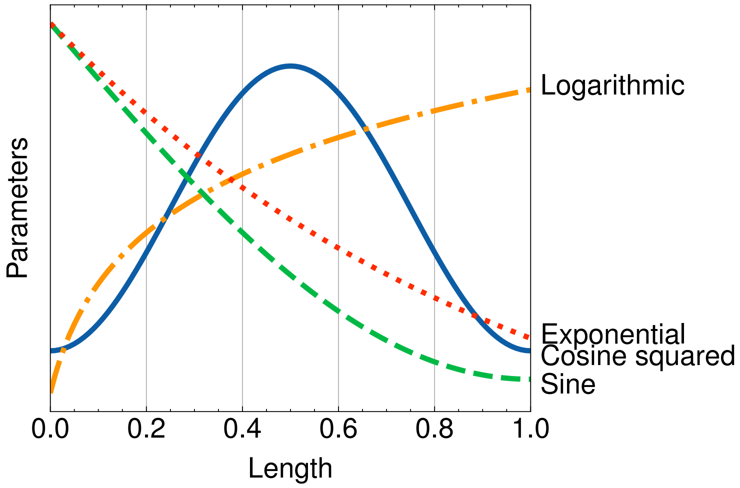


Figure 4.3 1D spatially distributed nonlinear shapes for Young's modulus. We refer to the functionally graded materials and display the possible choices for those non-linear distributions.

boundary condition is simply supported, which is implemented in the discretization. The load is distributed over the plate with a magnitude of $Q = 20N/m^2$. The initial condition of the plate is static, where the deflections and velocities are 0 at all points. Finally, the dynamic response is analyzed for 0.3s. The number of timesteps N_t is set to be 101, so $\Delta t = 0.003s$. This time resolution is tested to be enough to capture accurate peak values in the response. For ground truth generation, the mesh size is set to be 17 × 17, which has enough accuracy according to Figure 4.2.

For generating the ground truth data, we randomly pick nonlinear functions from Figure 4.3 to construct plate parameters D_1, D_2, and h as described in 4.3. Different cases are formulated and the ground truth dynamic response is calculated by solving the PDE.

The plate vibration problem is much more computation-intensive than the beam case. Direct training of the network with a fine mesh would take a few days, which makes it inefficient in real-life applications. As we tested, a smaller mesh size will largely reduce the computational costs during training, especially on the backpropagation of the PDE solver (Figure 4.4).

We try to train the network progressively starting from a coarse mesh of the plate as described in Chapter 4.2. After training, we improve the resolution of the prediction by increasing the mesh size until the dynamic response error reaches a tolerable level.

In detail, we discretize the plate into a $N_x \times N_y$ size of the mesh, where $N_x = N_y = 17$. This

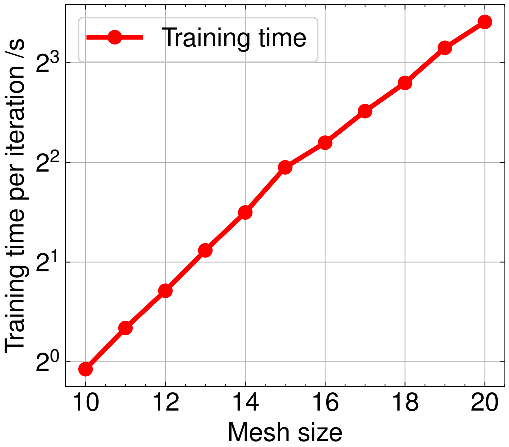


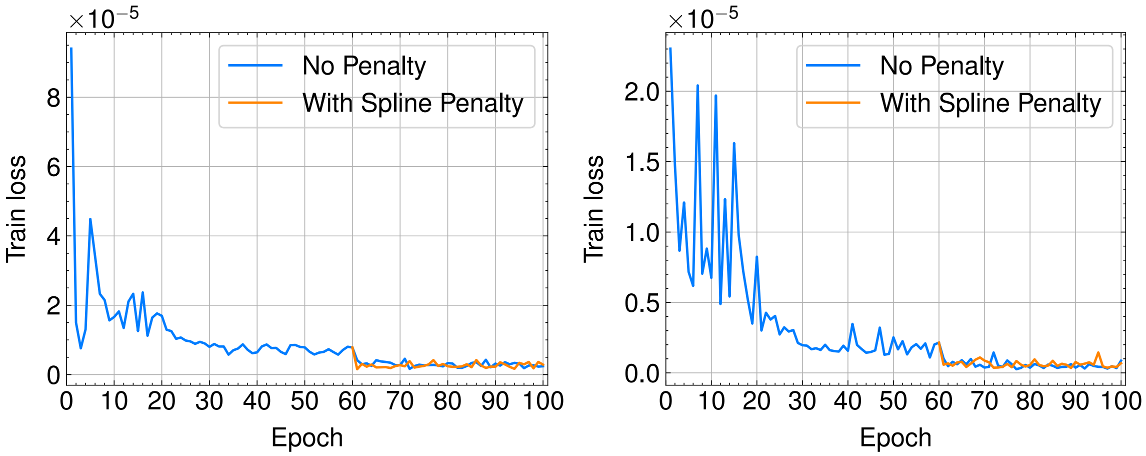
Figure 4.4 The training time per iteration with respect to the mesh size. The training cost increased greatly as the mesh size increased. This makes training with large meshes impossible for large-scale structures. Thus, it is necessary to reduce the mesh size and progressively train the network.

size gives a good trade-off between a high-resolution distribution of the nonlinear properties and the computational cost. We start to recover the parameters from a coarse mesh of 11 × 11, where the numerical errors from the dynamic responses are within an acceptable range (Figure 4.2).

To account for the inconsistency of mesh size in loss computation (i.e., by comparing to the predicted dynamic responses for prediction errors), the ground truth dynamic responses calculated in 17 × 17 will be interpolated ([191]) into the size of 11 × 11. Since the responses at the boundaries and at the initial time step are always 0, only a data size of $(N_x - 2) \times (N_y - 2) \times (N_t - 1)$ is used for error (loss) computation.

The neural network has 3 branches $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$ that are identical in the architecture. Respectively, They predict parameters D_1, D_2, and h. It is worth noting that we used a 2D mask for the loss calculation at different locations of the plate. This is because the response magnitude is much higher in the center than at the boundaries. The mask is customized to balance the MAE contribution from different areas with the expression: $10 - 9sin(x/\pi)sin(y/\pi)$, where x and y here represent coordinates of the measured points re-scaled between 0 and 1.

Another finding is that the minibatch size needs to be small to achieve an accurate recovery, regardless of the mesh size used for training. The minibatch size or the number of points selected to compute for errors is 32. And only 80% of the data are used in training.



(a) Case 1

(b) Case 2

Figure 4.5 Loss curve for different cases. The first 20 epochs use a small mesh size of 11 and take a larger learning rate to roughly learn the parameter distribution, the next 40 epochs refine the distribution with a smaller learning rate. At epoch 61, we re-scale the mesh size from 11 to 13 and continue training to reach optimal distributions.

The training is separated into 3 stages: the first 20 epochs use a learning rate of 2E-4 to roughly learn a shape of the 2D parameter value distribution, and the next stage uses a small learning rate of 4E-5 for refining the shape. This training strategy is proven to avoid local minima issues and effectively fit many different property distributions as we tested. For cases we tested, we also progressively increased the mesh size to 13×13 and continued training for another 20 epochs.

It is worth mentioning that for comparison, we also performed direct training with the 17×17 fine mesh, and the parameter prediction results obtained are very similar to the progressive train result in Figure 4.9. By checking the training loss curve by epoch in Figure 4.5, there is little improvement during the later stage of the training. In this case, it is not necessary to continue to increase the mesh size for further training.

4.4 Results

Experimental validation of a beam. A composite beam is 3D printed using an Ultimaker S5 for experimental validation. The length L_x, width, and thickness are $30.5cm$, $4.5cm$, and $1.0cm$, respectively. It consists of two identical wedges printed with PLA (red) and ABS (black) filament, as depicted in Figure 4.6a. During printing, a 100% infill density with a cubic infill pattern is used to ensure good structural integrity. The printing resolution is set to $0.06mm$. To enhance the bonding of the two different materials, the top/bottom surface layers are reduced from 4 (default) to 0, the top/bottom layer thickness is reduced to 0.6 mm, and the wall count is set to 0.

As shown in Figure 4.6b, an MTS loading frame module is employed to conduct a four-point bending test in displacement-control mode. A continuous cyclic loading is applied with a frequency of $2Hz$ for $3s$. The loading profile is shown in Figure 4.7. The loading is symmetrically applied at $12.7cm$ and $17.78cm$ ($\frac{5}{12}L_x$ and $\frac{7}{12}L_x$) from one end of the beam. The force sensor records the applied force with a high sampling rate of $6144Hz$, serving as the input for the differential equation. The beam is subjected to simply supported boundary conditions and initially remains in a static state with no deformation.

For dynamic response monitoring and collection, an image-based method is utilized to accurately capture the movement of the beam edge. In Figure 4.6c, the edge of the beam is coated with blue fluorescent paint and illuminated with black lighting to visually distinguish the fluorescent light from its ambient background [192]. Thirteen reference points are marked along the transverse direction of the beam, evenly spaced at a distance of $2.54cm$. Images are captured at a rate of 30 frames per second (fps) and processed using the Canny edge detection method to track the movement of the beam edge at the reference points along the longitudinal direction. The final dynamic response is measured and displayed in Figure 4.7b.

A constant density value is assumed for the beam because the dynamic response is primarily influenced by nonlinear modulus and damping. The beam weight measures $0.1734kg$, yielding a calculated density of $1215kg/m^3$ based on the beam dimensions. Young's modulus reference values are set to $E_0 = 2.4GPa$, derived from the modulus of PLA and ABS filament, while the damping reference is $c_0 = 10s/Nm$. The 13 reference points marked on the beam serve as mesh points for spatial discretization. Neural networks are constructed to predict the modulus and damping at these 13 discrete points. Predicted dynamic responses are then calculated using the PDE solver and compared to the ground truth dynamic responses for loss assessment. Notably, only the response data from the first 2 seconds is utilized for network training.

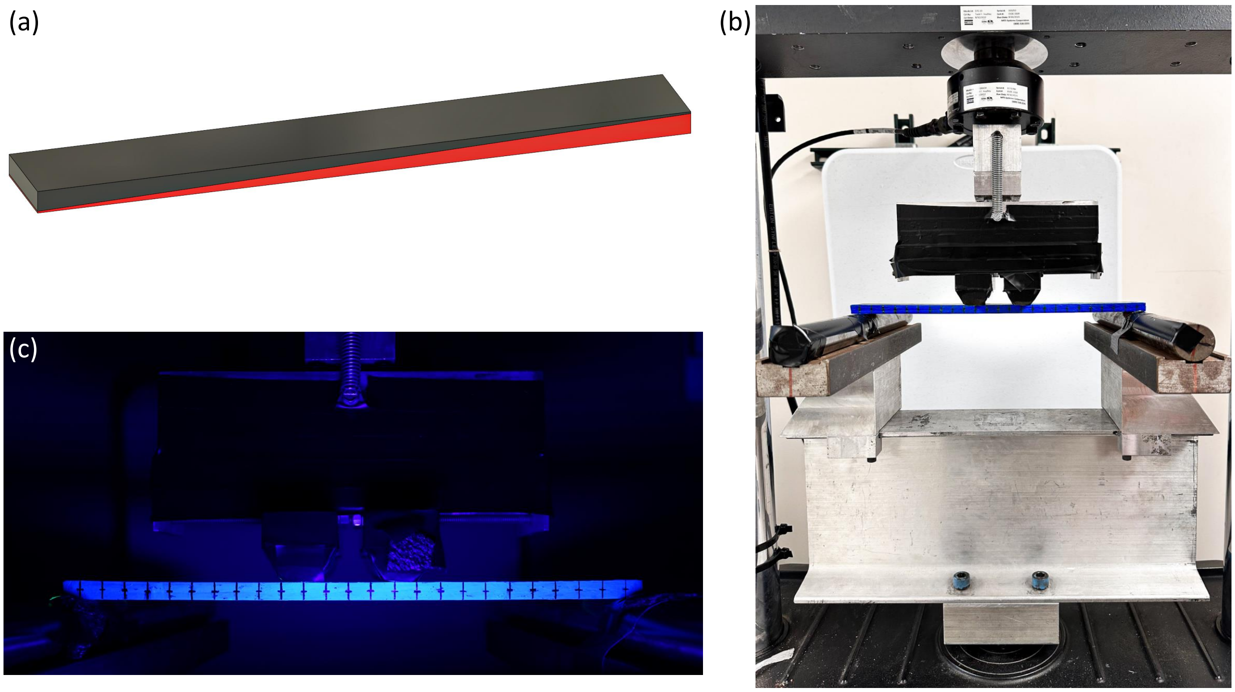


Figure 4.6 Beam vibration experiment. (a) A demonstration of 3D printed composite beam. (b) Experimental setup. The printed beam is under the MTS loading frame for dynamic loading. (c) The beam is placed under fluorescent light to measure displacement using edge detection techniques.

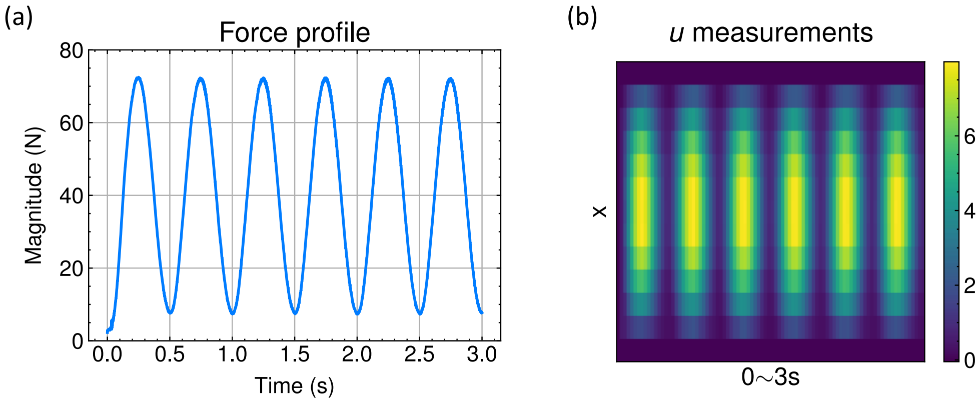


Figure 4.7 Experimental loading and collected beam response. (a) Loading profile. The load is applied at two symmetric points on the beam. (b) The measured beam spatial-temporal response using image-based method. The magnitude unit is mm.

4.4.0.1 Results

After network training, the graph in Figure 4.8a illustrates the predicted Young's modulus and damping. Notably, the projected beam modulus evidently showcases a linear decrease, which corresponds with the observed linear variation in PLA and ABS materials along the longitudinal axis. This serves as a reference point for comparing against the actual values. However, determining

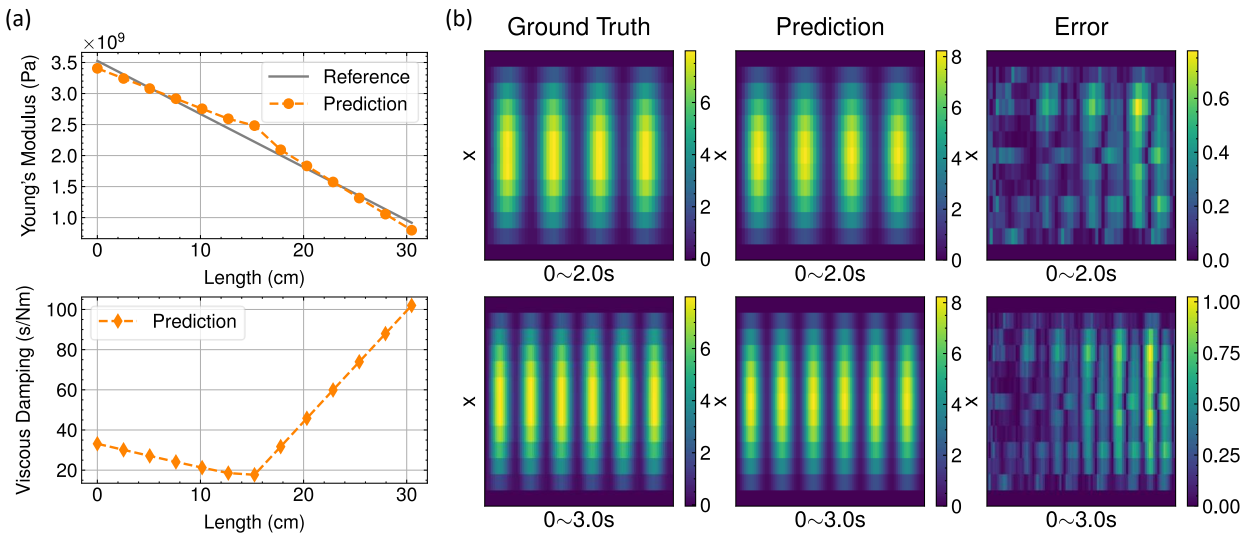
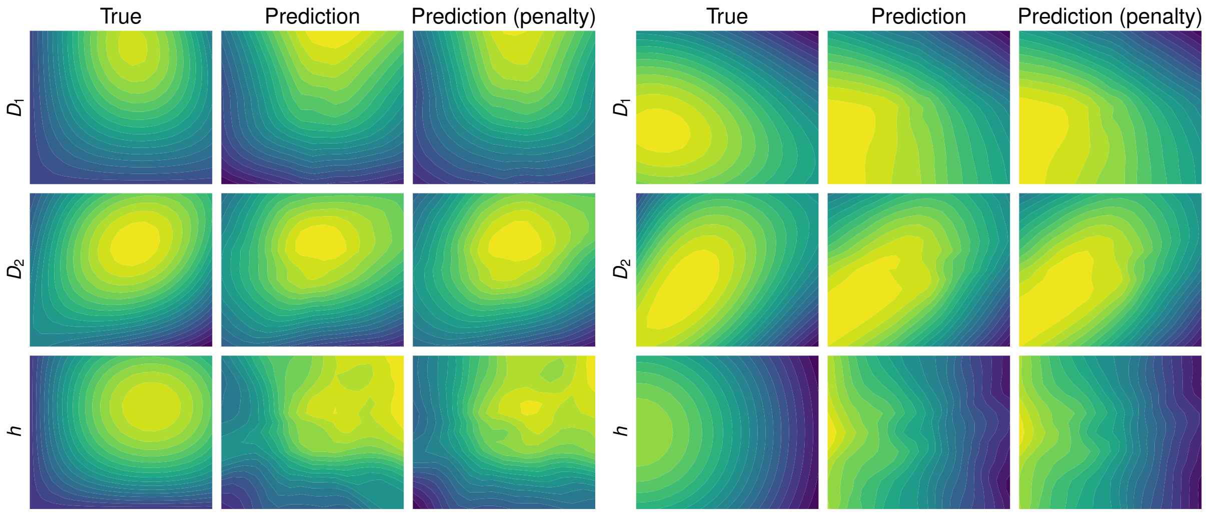


Figure 4.8 Parameter estimation results and beam vibration predictions. (a) The estimated young's modulus and damping fields. (b) The prediction dynamic responses. The top row shows the predictions for the training region. The bottom row displays the temporal extrapolation.

the precise ground truth modulus proves challenging due to the intricacies of the printing processes and conditions. Moreover, directly measuring damping values presents difficulties, hence only the model's predicted damping field is depicted in Figure 4.8b. Despite encountering notable errors during data collection and processing, the predicted dynamics depicted in Figure 4.8b notably align well with the collected response data. Specifically, dynamic responses are evaluated with MAE. Even with experimental data noise, satisfactory results are achieved with errors of $0.18mm$ for the training region and comparable error levels of $0.24mm$ for temporal extrapolation.

2D Plate vibration. The trained network performance can first be evaluated based on the prediction errors of the target parameters. Instead of estimating them at the coarse mesh, we query the parameters prediction in a high resolution by taking points from the mesh. Moreover, based on the FGM property from Equation 4.16, we can perform post-processing by fitting the learned parameters with 2D splines through Dierckx (a Julia package).

In Figure 4.9, The parameters learned from the neural network and from the 2D spline fitting are compared with the original parameters (ground truth) in fine mesh. Compared to results directly obtained from the neural network, the distributions from the fitted 2D spline curve present much smoother contours, and some areas match better with the distribution of the target parameters.



(a) Case 1

(b) Case 2

Figure 4.9 The parameter estimation results. The network prediction results (second column) have some mismatches with the ground truth, which can be further improved with 2D spline fitting (third column). Overall, both approaches are accurate, and the results are within a tolerable error.

For quantitative analysis, we use MAPE to evaluate the parameter estimation accuracy. The network prediction of three parameters is accurate and well below 6% for both cases tested. The fitted spline curve further reduces the MAPE for most distributions. However, it is worth noting that the spline fitting method is based on the assumption of a known parameter distribution from Equation 4.16, which might not be valid for some real-life cases. But overall, both types of results present tolerable errors that are much below the threshold for general civil infrastructure applications [193].

The plate dynamic responses are calculated based on both network prediction and spline fitting. Although the spline fit method provides a better match in terms of the parameters, the calculated dynamic responses have higher errors. It is possible that good solutions are not unique so that different combinations of parameter distributions can provide similar plate behavior.

4.5 Summary

In this chapter, we expanded the framework of NeuralSI for parameter identification in complex nonlinear dynamic systems. While NeuralSI can model the unknown parameters with neural networks, the computational cost grows greatly from the embedded PDE. The proposed progressive training technique is utilized for efficiently estimating parameters for the plate vibration problem. The training is divided into several different stages starting from a coarse mesh for focusing on learning the parameter distribution, followed by switching to a finer mesh to ensure accurate parameter estimation. A penalty loss function is implemented for a later stage of the training to further refine the smoothness of the predicted parameter distribution. The final stage will be determined by the training progress and the performance of dynamic response prediction. Different combinations of nonlinear parameter distributions are studied. The final estimation of parameters is all within a maximum MAPE of 3%, and the estimated structural behaviors are within 1% in terms of MAPE compared to the true responses, which are well below the tolerance of civil engineering applications. This well demonstrates the effectiveness of the proposed structural parameter estimation approach. Future endeavors will concentrate on expanding this framework by integrating the finite element method to accommodate more generalized 3D structures.

CHAPTER 5

ESTIMATING PARAMETER FIELDS IN MULTIPHYSICS PDES FROM SCARCE MEASUREMENTS

5.1 Overview

Physical phenomena are often governed by differential equations and evolve with time. Such variations primarily arise from wear and tear, aging, or other environmental effects, with the governed differential equations remaining unchanged while the underlying physical parameters change spatially over time. Variations in parameters across many scenarios can potentially lead to drastic changes in physical responses and have serious consequences. For instance, changes in cardiac tissue properties within human bodies can cause arrhythmias or atrial fibrillation [194, 195]. Similarly, changes in flow properties in porous media can lead to health and safety risks or mismanagement of groundwater resources, such as water contamination or depletion of aquifers. Consequently, there is a dire need for customized models to frequently monitor and estimate these parameters for each unique physical phenomenon.

Additionally, physical phenomena are often complex for various reasons. Firstly, the system parameters may not be constant scalars but rather spatially varying field quantities. Furthermore, the governing equations themselves can exhibit complexity arising from multiphysics or coupled phenomena. Research efforts have primarily concentrated on simplified single-physics problems with constant scalar coefficients to estimate parameters and understand the underlying physics. Various methods are employed including finite element updating [196, 197], Bayesian neural networks [198], least squares estimation [199, 200], Kalman filter [201, 202], Gaussian process [203, 204, 205], and sparse identification [56, 72]. However, applying those methods for non-constant field parameter estimation can present challenges in computation or assumptions. In approaches like Bayesian methods, limitations arise from the assumption that unknown parameter values adhere to a prior distribution, which might be impractical when dealing with unknown field variables. Sparse identification, as one of the popular numerical approaches for recovering system parameters or coefficients, is often limited to scalar parameters. In addition, the considerable non-linearity in the system, particularly in multiphysics problems, poses difficulties for inverse estimation when employing statistical approaches for parameter estimation.

Recently, deep learning powered by knowledge of physics has transformed how complex physical phenomena are learned. This fusion enables deep learning networks to grasp domain-specific knowledge, such as governing differential equations, thereby enhancing the comprehension of diverse physical phenomena and engineering responses. Specifically, PINNs [206, 72] have received a lot of focus. This machine learning-based approach is known for its ability to integrate domain knowledge into a black-box model and handle various unknown physical phenomena efficiently. PINNs are widely used in solving forward problems [207, 132], but many inverse problems such as parameter estimations are also being addressed nowadays [194, 208, 209], Particularly, PINNs can also be applied to model field parameters in scenarios involving complex multiphysics or coupling effects [194, 208, 210, 211, 212] due to the flexibility of neural network and automatically differentiation. In those scenarios, the field parameters and state variables are often predicted simultaneously with different neural networks. The objective function can hinge on the error of the governing equation, which accounts for both the parameters and state variables. Another method addressing multiphysics problems and estimating field parameters involves leveraging Karhunen-Loève expansions, which is widely utilized across numerous research domains [213, 214, 108]. However, most of the statistical expansions limit their applications on spatial-dependent PDEs without considering the temporal domain [213, 214, 208], or rely on prior assumptions of the parameter distribution [108], which restricts their applicability to other domains. Furthermore, few approaches for parameter field estimations in multiphysics problems are often rooted in specific domains, with applications in geotechnics employing back analysis [215, 216] or chemical processes utilizing Aspen Custom Modeler [217].

Despite various methods proposed for estimating parameters across diverse setups and scenarios, modern computational methods and their applications pose several distinctive challenges in real-world physics problems. *First*, a predominant focus of parameter estimation methods revolves around black-box modeling and concurrent estimation of physics responses, such as PINNs. This approach may fall short in offering practical utility, particularly concerning state variable estimation in scenarios like time-domain extrapolation and changes in boundary conditions, initial conditions, or other variations. *Second*, the omission of field parameter assumptions within highly nonlinear systems, particularly in scenarios involving multiphysics or coupled problems, is prone to fail and produce large discrepancies in engineering responses. *Third*, the scarcity of real-world measurements poses a challenge that data-driven methods or the application of PINNs may encounter difficulties accurately estimating parameters or capturing the underlying physics effectively [218].

This chapter tackles the aforementioned challenges by introducing NeuroFieldID, utilizing neural networks to estimate parameters that characterize various nonlinear PDE systems directly. We assume that the observed physics can be represented through PDEs, which consist of unidentified field parameters capable of characterizing the physical phenomena within the computational domain. To address this, we initialize DNNs aimed at modeling these unidentified physics parameters that depend upon spatial information or state variables. The parameter distributions vary significantly with distinct patterns across the three problems. The input to these networks is adjusted according to the dependencies of these parameters in different applications. A scalar parameter can also be modeled by a single neuron. Subsequently, we proceed with spatial discretization utilizing FDM and transform the coupled PDEs into systems of ODE. In this form, the equations are solely dependent on time, and the spatial-temporal predictions can be computed at any timestep including irregular timesteps. During neural network training, the predictions are compared to the measured data in batches, and the resultant errors are minimized, utilizing the adjoint method [90, 219]. Following that, the optimized neural network model can accurately estimate the target parameters and effectively predict the physics behavior even beyond the training region (extrapolation). In the following section, we will show applications of parameter estimations, respectively in the field of the flow phenomena in porous media and cardiac electrophysiology.

5.2 Method

FDM and the PDE solver NeuroFieldID directly solves the PDE system using FDM for space discretization of PDEs and a differential equation solver for inference. The spatial derivatives within the PDEs can be approximated by central difference [220]. A second derivative central difference in the x dimension at a point $x = x_n$ is demonstrated in equation 5.1.

$$\frac{\partial^2 u(x = x_n, t)}{\partial x^2} \approx \frac{u_{n-1} - 2u_n + u_{n+1}}{\Delta x^2} \tag{5.1}$$

where Δx is the mesh size of the finite difference, and u_n is the discrete values at x_n for the state variable u. Particularly, to efficiently address this high-dimensionality in the space domain, the space dimensions of the state variables are organized in the form of a 1D vector. Assume the mesh size for 3D space is $n = n_x \times n_y \times n_z$, and the state variable u and the discretization matrices will have a size of $N \times 1$ and $N \times N$, respectively. The spatial derivatives of various orders are explicitly formulated for each derivative term in equation 5.2.

$$\frac{\partial^2 u}{\partial x^2} \approx A_2^x u, \qquad \frac{\partial^2 u}{\partial y^2} \approx A_2^y u, \qquad \frac{\partial^2 u}{\partial z^2} \approx A_2^z u \tag{5.2}$$

$$\frac{\partial u}{\partial x} \approx A_1^x u, \qquad \frac{\partial u}{\partial y} \approx A_1^y u, \qquad \frac{\partial u}{\partial z} \approx A_1^z u \tag{5.3}$$

where the matrices multiplication term A_1 and A_2 approximate the spatial derivatives numerically.

Moreover, the boundary condition needs to be carefully handled separately during the discretization. Two distinct types of boundary conditions (Dirichlet and Neumann) are handled for the problems addressed in this paper. The expressions are displayed in the following equation, respectively.

$$u(x) = \gamma \tag{5.4}$$

$$\frac{\partial u(x)}{\partial x} = \beta \tag{5.5}$$

The implementation of these boundary conditions involves modifying the discretized matrix described in equation 5.2. Details of handling the boundary conditions can be also referred to in the previous studies [221, 218].

Subsequently, the PDE system is addressed as a set of ODEs and solved by the Runge–Kutta family of differential equation solvers [222]. The ODE system is solved, and dynamics are extracted at specific timesteps corresponding to the observations. It is worth highlighting that periodic activation functions (i.e., Sine) [223] are employed as activation functions to tackle the high-variations in spatially-varying field parameters (see Fig. 5.2).

Specifically, the space-dependent unknown parameter p_1 and state-dependent variable p_2 can be modeled by different networks. The neural network parameters are $\boldsymbol{\theta}$. To further extend on this, the network input x, y, and z are discretized in the form of discrete points to model the space-dependent field parameters at the corresponding locations. For state-dependent field parameters, the inputs are temperature values at discrete locations within the spatial domain. Additionally, the input values are scaled from -1 to 1 for better training performance.

Network training and adjoint backpropagation This work utilizes the Julia programming language, employing adjoint backpropagation within the emerging field of neural ODEs. However, traditional Neural ODEs often assume unknown or partially known differential equations, modeling these components with black-box neural networks. In contrast, our approach diverges by leveraging known parametric expressions of PDEs with unknown field parameters, which are subsequently modeled using neural networks.

Furthermore, unlike conventional ML methods that typically rely on explicit formulations, neural ODEs solve differential equations implicitly. This implicit approach directly integrates fundamental patterns from physics, enabling the representation of characteristics in the form of differential equations within neural network architectures. Consider an ODE system with a state variable u, where the corresponding dynamic response u_{t+1} can be calculated from the last timestep u_t. The loss, represented by MAE, can be expressed in the following equation.

$$u_{t+1} = u_t + \int_t^{t+1} f(u_t, p)dt \tag{5.6}$$

$$\begin{aligned} \mathcal{L} = |\mathbf{u} - \mathbf{u}_{true}| &= \left|\text{Solve}\left[f(t, p)\right] - \mathbf{u}_{true}\right| \\ &= \left|\text{Solve}\left[f(t, \mathcal{N}(\boldsymbol{\theta}))\right] - \mathbf{u}_{true}\right| \end{aligned} \tag{5.7}$$

In a standard neural ODE system, we utilize the adjoint sensitivity method for efficiently calculating the gradients. Let $a(t)$ denote the adjoint state [90].

$$\boldsymbol{a}(t) = \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}(t)} \tag{5.8}$$

Subsequently, Julia programming language leverages various adjoint sensitivity methods and reverse-mode automatic differentiation techniques [93, 219] for various differential equation solves, enabling the derivation of gradients with respect to the neural network model parameters. Utilizing the chain rule, these gradients are used to update the model parameters and minimize the loss function.

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial p} \frac{\partial p}{\partial \boldsymbol{\theta}} = \int_t^{t+1} \boldsymbol{a}(t)^\intercal \frac{\partial f(u(t), p)}{\partial p} \frac{\partial p}{\partial \boldsymbol{\theta}} dt \tag{5.9}$$

where term $a(t)^\intercal \frac{\partial f(u(t),p)}{\partial p}$ can be efficiently calculated by automatic differentiation. Notably, if the parameter p is only space dependent, term $\frac{\partial p}{\partial \boldsymbol{\theta}}$ can be calculated separately for more efficient computation.

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{\partial p}{\partial \boldsymbol{\theta}} \int_t^{t+1} \boldsymbol{a}(t)^\intercal \frac{\partial f(u(t), p)}{\partial p} dt \tag{5.10}$$

5.3 Results

Flow in porous media. Recent attention has been also drawn to critical issues such as contaminant transport and attenuation in water resources, along with the greenhouse effect resulting from carbon emissions. Such environmental concerns have prompted increased efforts to tackle these challenges. As observed across various natural systems [224, 225, 226, 227, 228, 229, 230], this phenomenon involves the movement of solutes in porous media, encompassing diverse processes such as subsurface fluid flow or groundwater flow through soil or aquifers, as well as carbon sequestration and storage. Despite similarities in the underlying governing equations, each problem involving the movement of solutes through porous media exhibits unique characteristics due to variations in environmental conditions, human activities, or geological processes. Understanding these phenomena and accurately estimating relevant parameters are crucial for effectively addressing those pressing issues. As an illustrative example, we focus on the subsurface transport problem, a

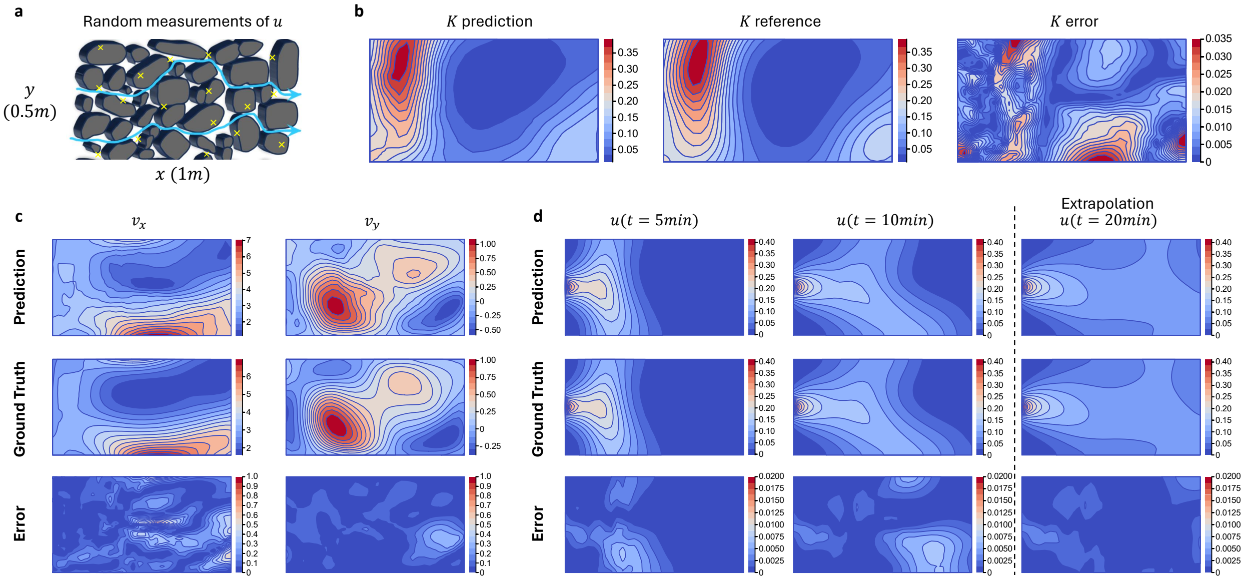


Figure 5.1 Field parameter estimation for flow in porous media. **a**. A demonstration of flow phenomena and random measurements are taken as the ground truth for model training. **b**. The estimated hydraulic conductivity K compared to the reference K. **c**. The computed flow velocity compared to the reference. **d**. The particle concentration predictions at different times. The predictions align well with the ground truth, even for extrapolation at $t = 20min$.

widely studied issue highlighted in recent literature [231, 208, 211]. This problem exemplifies the complexities of characterizing porous media and understanding transport phenomena within them.

The transport exhibits time-dependent behaviors and is described by two PDEs, the time-dependent advection-dispersion equation (ADE) coupled with Darcy's law. The ADE describes the concentration of particles based on the flow velocity **v**, and Darcy flow characterizes the fluid movement through porous media and establishes the relationship between hydraulic conductivity K and hydraulic head h. The ADE, Darcy's law, and the expression for the velocity term can be written as:

$$\frac{\partial u}{\partial t} + \nabla \cdot [\mathbf{v}u] = \nabla \cdot [\mathbf{D}\nabla u] \tag{5.11}$$

$$\nabla \cdot [K\nabla h] = 0 \tag{5.12}$$

$$\mathbf{v} = -K\nabla h/\phi \tag{5.13}$$

where state variable u represents the particle concentration field, **v** is the average pore velocity in the x and y direction in the 2D space, porosity ϕ is 0.317. Dispersion coefficient **D** is given as $\mathbf{D} = D_w \tau \mathbf{I} + \alpha \|\mathbf{v}\|_2$, with diffusion coefficient $D_w = 0.09 m^2/hr$, the tortuosity of the medium $\tau = 0.681$. Dispersivity α is a diagonal matrix with the principal components $\alpha_L = 0.01m$, and $\alpha_T = 0.001m$.

In this problem, the 2D space domain is defined as $L_x = 1m$ and $L_y = 0.5m$, as shown in Fig.5.1a. Random spatial-temporal measurements of the concentration u are taken within the domain. The initial condition for ADE is defined as:

$$u = \begin{cases} exp\big(-1600(x - L_x/2)^2\big), & \text{if } y = 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.14}$$

and Neumann boundary conditions of u are applied to all boundaries. For the steady state Darcy flow equation, the boundary conditions are:

$$h(x, y = 0, t) = 0 \tag{5.15}$$

$$-K\partial h(x, y = L_y, t)/\partial t = 1 \tag{5.16}$$

$$-K\partial h(x = 0\, or\, L_x, y, t)/\partial t = 0 \tag{5.17}$$

The field parameter of interest is the spatially varying hydraulic conductivity $K(x, y)$, which is popularly studied in previous studies with experimental work and PINN models [232, 208, 211, 233]. During the training process for parameter estimation, the neural network model takes 2D spatial coordinates as input. The model utilizes the parameter K to predict the flow phenomena of particle concentration within the domain. This predicted concentration is then compared with the ground truth concentration to assess errors. The ground truth data is measured roughly every $16s$. It is important to note that we take 10 direct parameter measurements because the system cannot be solely determined from the state variables measurements. In this case, the modeled parameter field K from the neural network is compared to the known measurement, and the parameter errors are integrated with the errors calculated between the state variable predictions and observations,

Figure 5.1b displays the estimated parameter K after training. Compared to the reference, the error contour displays relatively low discrepancies with an MAE of 0.011. As illustrated in Fig.5.1c, the flow velocities in both directions are also accurately prediction with relatively low errors compared to the reference. Figure 5.1d showcases the evolution of the state variable u over time. It is noteworthy that the training data only consists of the initial 16 minutes of spatial-temporal data. Despite this, the estimated field parameter K can accurately predict the concentration u beyond the training region. The contour results extrapolated at $20min$ exhibit low error and align well with the ground truth.

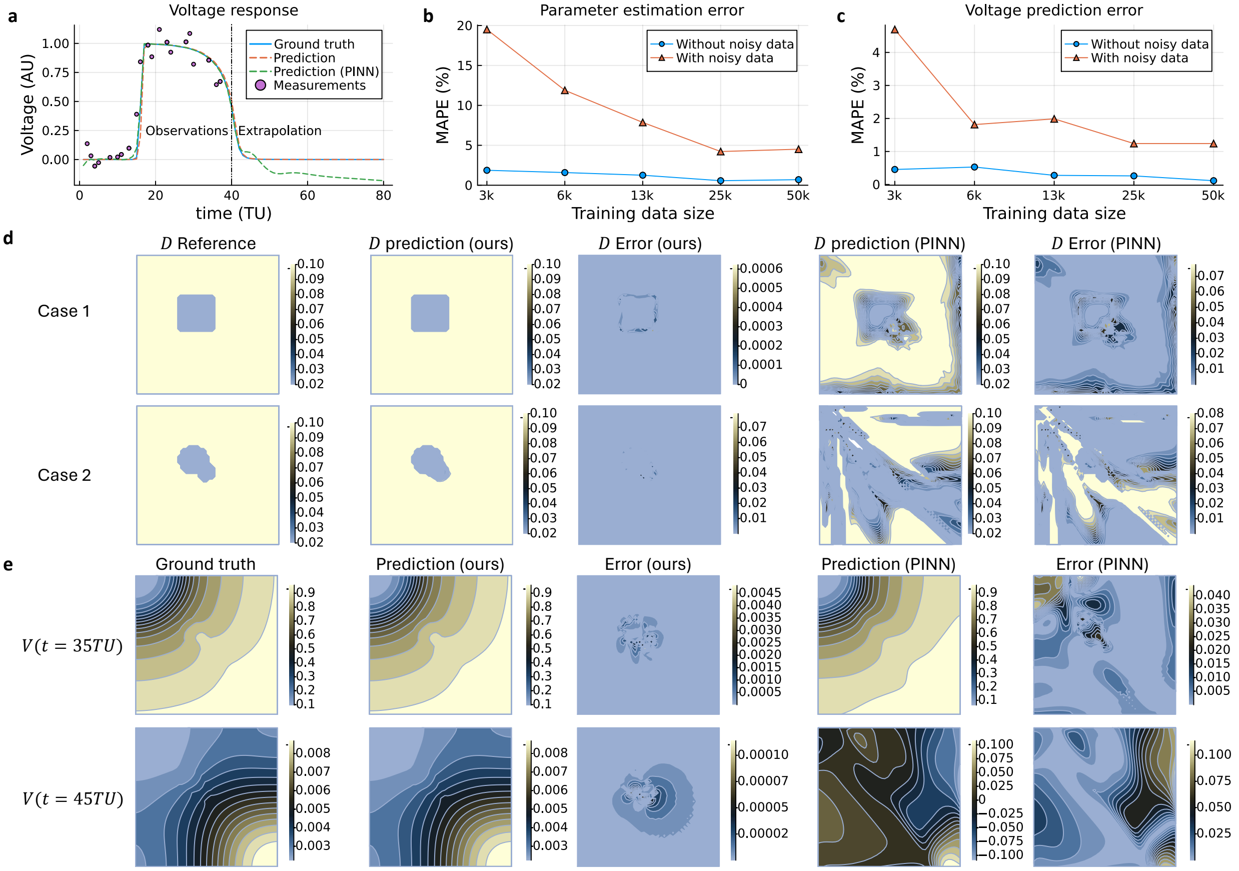


Figure 5.2 Parameter estimation in a cardiac electrophysiology application. **a**.The training region with Gaussian noise during measurement, and the estimated forward inference of V. Compared to PINN, NeuroFieldID achieves much higher accuracy in both the training and extrapolation regions. **b**. The parameter estimation error under scarce data. (PINN exhibits 40% MAPE with 25k data). **c**. The forward estimation error of V under scarce data. **d**. The parameter estimation results are presented for two representative cases with different field parameter D distributions. **e**. The estimated response V is compared to the ground truth for case 2. When performing temporal extrapolation, PINN predicts values that do not adhere to physics laws, while NeuroFieldID maintains good prediction accuracy.

Cardiac electrophysiology application and comparison with a PINN. Cardiac electrophysiology (EP) stands out as a popular field in multiphysics, emphasizing the crucial coupling between cardiac tissue properties and the generation as well as propagation of electrical signals within the heart. Numerous research efforts are dedicated to exploring this intricate relationship, focusing on physics-based models [234] and PINN models [235, 236, 194]. A subset of these studies delves into estimating cardiac tissue electrical conductivity using in silico data [237, 195, 194], specifically identifying heterogeneities. This aspect holds significant potential for clinically relevant applications, particularly in the detection of fibrosis and other localized pathologies associated with arrhythmias or atrial fibrillation [194, 195]. The canine ventricular Aliev-Panfilov model is utilized in this section, with the coupled equations [194] defined in the following.

$$\frac{\partial V}{\partial t} = \nabla(D\nabla V) - k_0V(V-a)(V-1) - VW \tag{5.18}$$

$$\frac{\partial W}{\partial t} = \big(\epsilon + \frac{\mu_1 W}{V+\mu_2}\big)\big(-W - k_0V(V-b-1)\big) \tag{5.19}$$

where the diffusion tensor D determines the propagation speed and is proportional to the electrical conductivity of the tissue σ. a and b are known scalar related to the tissue excitation threshold and refractoriness [194]. The state variable, transmembrane potential V, represents the membrane voltage and is accessible in experimental measurements. W denotes an unknown state variable. $k_0 = 8, \mu_1 = 0.2, \mu_2 = 0.3, a = 0.01, b = 0.15$, and $\epsilon = 0.002$.

We employ a 2D slab of cardiac tissue ($1cm \times 1cm$) as the spatial domain and aim to recover heterogeneous diffusion tensor $D(x, y)$. Healthy tissue is represented by $D = 0.1mm^2/TU$, while fibrosis tissue is represented by $D = 0.02mm^2/TU$. $1TU$ is roughly $13ms$ [194]. Neumann boundary condition is applied with $\frac{\partial V}{\partial x} = 0$ and $\frac{\partial V}{\partial y} = 0$.

During training, Gaussian noise is added to the training data (transmembrane potential V measurements) to mimic real-world situations. Voltage response data from a random spatial point is shown in Fig.5.2a, and the training data spans the first $40TU$ measured every $1TU$. Examining the predicted Voltage response V, it is evident that our NeuroFieldID strictly adheres to the physics and showcases robustness in predictions extending beyond the training region, with precise temporal extrapolations after $40TU$, while PINN predicts negative V values in many timesteps.

In Fig.5.2b and Fig.5.2c, we thoroughly examine the challenges posed by training data scarcity, illustrating parameter estimation errors for D and forward inference errors for V, respectively. It is shown that NeuroFieldID demonstrates robust and accurate field parameter estimation performance, even in scenarios involving noisy data and a limited number (only a few thousand) of training points (i.e., the parameter distributions can be referred to the second row of Fig.5.2e). When employing 25,000 data points (50% data of all V responses) for training, NeuroFieldID achieves a parameter estimation MAPE of less than 5%, whereas PINN exhibits errors exceeding 40%, thereby resulting in significantly inaccurate voltage predictions. In Fig.5.2d, two cases featuring distinct parameter distributions are depicted for comparison in parameter estimation with PINN. The results illustrate that NeuroFieldID effectively recovers parameters in both cases. However, the PINN approach fails to estimate the random distribution D in case 2. Even in case 1, where both methods accurately capture the region with blue colors, the MAPE for the PINN approach is 13%, while NeuroFieldID achieves an error that is two orders of magnitude more accurate, with a MAPE of less than 0.1%. Subsequently, utilizing case 1, the forward V predictions are compared in Fig.5.2e. Within the training region (the first row), PINN achieves good predictions but still has one order magnitude higher peak error than NeuroFieldID. However, PINN exhibits much higher errors for temporal extrapolation (the second row) due to inaccurate estimation of parameter D.

5.4 Summary

In this paper, we presented NeuroFieldID for parameter field estimation for various multiphysics PDE problems with scarce data. We thoroughly examined our approach in three emerging fields for estimating parameters that are usually unknown and important for identifying the physics phenomena. For the flow of porous media problem, we studied the hydraulic conductivity that depends on the space domain. The estimated conductivity can help model the flow phenomena. In the third problem, the diffusion property representing the tissue's electrical conductivity is estimated, aiming at detecting fibrosis and other localized pathologies associated with arrhythmias or atrial fibrillation. Among all three case studies, the parameter distributions vary significantly with distinct patterns, and the estimation results show satisfactory agreements with the reference parameter fields. Although we only list examples in the multiphysics domain, the method is also very efficient when dealing with single physics problems or with scalar parameters (such as diffusion equation, burgers equation, heat transfer, and advection problems).

Another advantage of this method is its capability to facilitate multiple parameter estimations. This is achieved by modeling various parameters through separate network branches, allowing each branch to possess flexible sizes and architectures. Based on the different scenarios, parameters can be learned simultaneously or separately in a sequential manner. Furthermore, the neural network output is a continuous function, which allows for high-resolution parameter evaluation at any point within the domain.

Notably, compared to PINN, the proposed approach requires even less data for learning the physics or estimating the parameters. This could be crucial in scenarios like temperature measurement, where only surface measurements are accessible, or in flow problems, where too many measurements are costly and time-consuming. In addition, the output from NeuroFieldID strictly obeys physics, even under the variations of boundary conditions or initial conditions. This is because NeuroFieldID incorporates that information into the solver and can still predict the physics behavior correctly. However, a PINN approach may often suffer from those conditions changes and necessitate a model retrain. Nevertheless, it is still worth mentioning that, unlike PINN, NeuroFieldID is still challenged by dimensionality and scalability. A relative increase in the number of mesh sizes would result in an exponential increase in calculation time. One potential approach to mitigate this challenge is by initially training on a coarse mesh to reduce the computation and subsequently refining the results on a finer mesh.

The finite difference approach enables rapid and precise forward solutions, significantly enhancing the efficiency of training through backpropagation using the adjoint method for gradient computation. One drawback of FDM is the constraint of geometries. The space domain are usually rectangular or cylindrical and can be easily discretized using Cartesian or polar coordinates systems. Regardless of many approaches have been trying to handle complex geometries in specific domains such as in mechanics [238], heat transfer problem [239], and cardiology [240], such approaches often cannot be generalized for efficient solving for the forward solutions. Nevertheless, most physics phenomena are often studied in a confined domain where the geometry can be well-managed for FDM. A more generalized approach can be based on the FEM and follow the same framework proposed in this article. It is important to note that employing the FDM approach also introduces numerical errors in forward solutions. However, these errors largely depend on the chosen mesh size, representing a trade-off between computational cost and accuracy. Our tests reveal that using a relatively smaller mesh size, despite yielding larger numerical errors, has minimal impact on field parameter estimation. Furthermore, caution should be exercised when employing FDM in certain domains, as the accumulation of numerical errors may impede convergence, especially in complex dynamic systems. However, such scenarios are rare in our work, even within the multiphysics domain.

Finally, it is worth mentioning that a correct starting point for estimating parameters is crucial. Incorrectly guessing parameters can result in a violation of physical principles, leading to the failure to solve PDE systems and nonsensical training loss. Moreover, a proper parameter guess can also help avoid local minima. Tasks like parameter estimation from observation data are generally treated as inverse problems. Such problems are often ill-posed and prone to getting stuck at local minima, especially when the parameter field is complex or exhibits high variations, or when dealing with multiple parameter estimations. Having access to prior knowledge can greatly help in addressing this issue. However, in this article, we focus on showcasing a generalized approach for parameter estimation based on governing PDEs, which can be further enhanced for domain-specific applications. For example in the flow problem, if parameters can be directly measured in a few locations, a penalty can be directly applied to the objective function by minimizing the errors between the estimated parameters and the measured parameters. Additionally, if the parameters are continuous and smooth, a spline fitting penalty can also be applied.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter presents several overarching conclusions from the previous chapters' work and outlines future research directions.

6.1 Conducted work and research contributions

The research presented in this thesis significantly advances the assessment and condition prognosis of structural systems and greatly enhances the modeling of general dynamic systems. Specifically, this thesis makes substantial contributions to two major areas: anomaly detection and inverse modeling (parameter estimation).

In the domain of anomaly detection, the thesis introduces a novel approach known as the Mechanics-Informed Autoencoder, designed for the automated detection and localization of damage within structural systems. This method represents a significant advancement by combining ML with a mechanics-informed loss function, allowing it to accurately localize structural damage. Additionally, the approach integrates data compression techniques with inexpensive sensor devices, creating a "deploy-and-forget" system that enables automated SHM without the need for human intervention. The performance of this method is noteworthy, with results showing up to a 35% improvement in accuracy for damage detection and localization compared to standard autoencoder models. Moreover, the approach's generalizability has been demonstrated through extensive experimental and numerical studies conducted under various conditions, making it a robust solution for real-world applications.

In the domain of inverse modeling, this thesis contributes by developing a comprehensive framework for estimating unknown parameter fields within differential equations. Within the field of civil engineering, this framework makes a significant contribution by enabling precise estimation of unknown and hard-to-measure structural parameters, as well as accurate prediction of structural responses. This enhances the assessment and condition prognosis of structural systems while effectively addressing mismatches between design, testing, and actual built parameters, ensuring more accurate modeling of real-world structures. The developed inverse modeling framework outperforms other deep learning methods for identifying structural behaviors and provides an accurate structural response evaluation.

Furthermore, the framework's applicability extends beyond civil engineering to the broader field of engineering and dynamic systems. The extended framework is capable of handling nonlinear high-order parametric PDEs and can be applied to both forward and inverse problems in dynamic systems. It is particularly effective in scenarios where the PDE parameters are unknown or vary with complex spatiotemporal dynamics. Additionally, the framework's generalization potential enables its application to ODEs and PDEs across various scientific fields. Its versatility, coupled with the ability to significantly reduce the required training data by 10 to 100 times while decreasing prediction errors, makes it a robust approach for capturing complex physical behaviors and a valuable resource for dynamic system modeling and prognosis.

6.2 Future research

Building upon the contributions of this thesis, future research can focus on several key areas to further advance the field. First, there is an emphasis on scientific discovery for engineering systems, particularly in identifying and solving unknown equations and parameters within interdisciplinary fields. This will be crucial for advancing our understanding and capability to accurately model complex dynamic systems.

Second, the research will explore probabilistic modeling and Bayesian inference, with a focus on uncertainty analysis and risk anticipation. This approach will enhance the robustness of predictions and provide a deeper understanding of the potential risks associated with dynamic systems.

Lastly, scalable AI/ML techniques will be a significant area of future work. This includes the development and application of foundation models and generative AI, which are expected to play a crucial role in advancing the scalability and applicability of AI/ML in engineering systems and beyond.

BIBLIOGRAPHY

[1] Hoon Sohn, Charles R Farrar, Francois M Hemez, Devin D Shunk, Daniel W Stinemates, Brett R Nadler, and Jerry J Czarnecki. A review of structural health monitoring literature: 1996–2001. *Los Alamos National Laboratory, USA*, 1:16, 2003.

[2] Chunwei Zhang, Asma A Mousavi, Sami F Masri, Gholamreza Gholipour, Kai Yan, and Xiuling Li. Vibration feature extraction using signal processing techniques for structural health monitoring: A review. *Mechanical Systems and Signal Processing*, 177:109175, 2022.

[3] Krishna Chintalapudi, Jeongyeup Paek, Omprakash Gnawali, Tat S Fu, Karthik Dantu, John Caffrey, Ramesh Govindan, Erik Johnson, and Sami Masri. Structural damage detection and localization using netshm. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 475–482, 2006.

[4] D Goyal and BS Pabla. The vibration monitoring methods and signal processing techniques for structural health monitoring: a review. *Archives of Computational Methods in Engineering*, 23:585–594, 2016.

[5] Muhammad Ali Akbar, Uvais Qidwai, and Mohammad R Jahanshahi. An evaluation of image-based structural health monitoring using integrated unmanned aerial vehicle platform. *Structural Control and Health Monitoring*, 26(1):e2276, 2019.

[6] John Mark Go Payawal and Dong-Keon Kim. Image-based structural health monitoring: A systematic review. *Applied Sciences*, 13(2):968, 2023.

[7] Junfang Wang and Jian-Fu Lin. Structural health monitoring of periodic infrastructure: A review and discussion. *Data Mining in Structural Dynamic Analysis: A Signal Processing Perspective*, pages 25–40, 2019.

[8] Jeong-Seok Lee, Gyuhae Park, Chun-Gon Kim, and Charles R Farrar. Use of relative baseline features of guided waves for in situ structural health monitoring. *Journal of intelligent material systems and structures*, 22(2):175–189, 2011.

[9] Sandeep Sony, Shea Laventure, and Ayan Sadhu. A literature review of next-generation smart sensing technology in structural health monitoring. *Structural Control and Health Monitoring*, 26(3):e2321, 2019.

[10] Muhammad Habib ur Rehman, Chee Sun Liew, Assad Abbas, Prem Prakash Jayaraman, Teh Ying Wah, and Samee U Khan. Big data reduction methods: a survey. *Data Science and Engineering*, 1:265–284, 2016.

[11] Mou Wu, Liansheng Tan, and Naixue Xiong. Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications. *Information*

Sciences, 329:800–818, 2016.

[12] Hamed Bolandi, Nizar Lajnef, Pengcheng Jiao, Kaveh Barri, Hassene Hasni, and Amir H Alavi. A novel data reduction approach for structural health monitoring systems. *Sensors*, 19(22):4823, 2019.

[13] Bin He and Yonggang Li. Big data reduction and optimization in sensor monitoring network. *Journal of Applied Mathematics*, 2014, 2014.

[14] Ping Jiang, Jonathan Winkley, Can Zhao, Robert Munnoch, Geyong Min, and Laurence T Yang. An intelligent information forwarder for healthcare big data systems with distributed wearable sensors. *IEEE systems journal*, 10(3):1147–1159, 2014.

[15] FuTao Ni, Jian Zhang, and Mohammad N Noori. Deep learning for data anomaly detection and data compression of a long-span suspension bridge. *Computer-Aided Civil and Infrastructure Engineering*, 35(7):685–700, 2020.

[16] Gwanghee Heo, Chunggil Kim, Seunggon Jeon, and Joonryong Jeon. An experimental study of a data compression technology-based intelligent data acquisition (idaq) system for structural health monitoring of a long-span bridge. *Applied Sciences*, 8(3):361, 2018.

[17] Hassene Hasni, Pengcheng Jiao, Nizar Lajnef, and Amir H Alavi. Damage localization and quantification in gusset plates: A battery-free sensing approach. *Structural Control and Health Monitoring*, 25(6):e2158, 2018.

[18] Hassene Hasni, Amir H Alavi, Pengcheng Jiao, Nizar Lajnef, Karim Chatti, Kenji Aono, and Shantanu Chakrabartty. A new approach for damage detection in asphalt concrete pavements using battery-free wireless sensors with non-constant injection rates. *Measurement*, 110:217–229, 2017.

[19] Hassene Hasni, Amir H Alavi, Pengcheng Jiao, and Nizar Lajnef. Detection of fatigue cracking in steel bridge girders: A support vector machine approach. *Archives of Civil and Mechanical Engineering*, 17:609–622, 2017.

[20] Ahmed Ibrahim, Ahmed Eltawil, Yunsu Na, and Sherif El-Tawil. A machine learning approach for structural health monitoring using noisy data sets. *IEEE Transactions on Automation Science and Engineering*, 17(2):900–908, 2019.

[21] Rih-Teng Wu and Mohammad Reza Jahanshahi. Data fusion approaches for structural health monitoring and system identification: past, present, and future. *Structural Health Monitoring*, 19(2):552–586, 2020.

[22] Mohsen Azimi and Gokhan Pekcan. Structural health monitoring using extremely compressed data through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, 35(6):597–614, 2020.

[23] XW Ye, T Jin, and CB Yun. A review on deep learning-based structural health monitoring of civil infrastructures. *Smart Struct Syst*, 24(5):567–585, 2019.

[24] Chuan-Zhi Dong and F Necati Catbas. A review of computer vision–based structural health monitoring at local and global levels. *Structural Health Monitoring*, 20(2):692–743, 2021.

[25] Mohsen Azimi, Armin Dadras Eslamlou, and Gokhan Pekcan. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. *Sensors*, 20(10):2778, 2020.

[26] Gao Fan, Jun Li, and Hong Hao. Vibration signal denoising for structural health monitoring by residual convolutional neural networks. *Measurement*, 157:107651, 2020.

[27] Osama Abdeljaber, Onur Avci, Serkan Kiranyaz, Moncef Gabbouj, and Daniel J Inman. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388:154–170, 2017.

[28] Hamid Khodabandehlou, Gökhan Pekcan, and M Sami Fadali. Vibration-based structural condition assessment using convolution neural networks. *Structural Control and Health Monitoring*, 26(2):e2308, 2019.

[29] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, and Daniel Inman. Structural damage detection in real time: implementation of 1d convolutional neural networks for shm applications. In *Structural Health Monitoring & Damage Detection, Volume 7: Proceedings of the 35th IMAC, A Conference and Exposition on Structural Dynamics 2017*, pages 49–54. Springer, 2017.

[30] Do-Eun Choe, Hyoung-Chul Kim, and Moo-Hyun Kim. Sequence-based modeling of deep learning with LSTM and GRU networks for structural damage detection of floating offshore wind turbine blades. *Renewable Energy*, 174:218–235, 2021.

[31] Yi Zeng, Peng Pan, Zhizhou He, and Zhouyang Shen. An innovative method for axial pressure evaluation in smart rubber bearing based on bidirectional long-short term memory neural network. *Measurement*, 182:109653, 2021.

[32] Xingxian Bao, Zhichao Wang, and Gregorio Iglesias. Damage detection for offshore structures using long and short-term memory networks and random decrement technique. *Ocean Engineering*, 235:109388, 2021.

[33] Gang Liu, Lili Li, Liangliang Zhang, Qing Li, and SS Law. Sensor faults classification for shm systems using deep learning-based method with tsfresh features. *Smart Materials and Structures*, 29(7):075005, 2020.

[34] Lili Li, Gang Liu, Liangliang Zhang, and Qing Li. Fs-lstm-based sensor fault and structural damage isolation in shm. *IEEE Sensors Journal*, 21(3):3250–3259, 2020.

[35] Jun Kang Chow, Zhaoyu Su, Jimmy Wu, Pin Siang Tan, Xin Mao, and Yu-Hsing Wang. Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Advanced Engineering Informatics*, 45:101105, 2020.

[36] Zilong Wang and Young-Jin Cha. Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage. *Structural Health Monitoring*, 20(1):406–425, 2021.

[37] Madhuka Jayawardhana, Xinqun Zhu, Ranjith Liyanapathirana, and Upul Gunawardana. Compressive sensing for efficient health monitoring and effective damage detection of structures. *Mechanical Systems and Signal Processing*, 84:414–430, 2017.

[38] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Zero shot detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(4):998–1010, 2019.

[39] Adín Ramírez Rivera, Adil Khan, Imad Eddine Ibrahim Bekkouch, and Taimoor Shakeel Sheikh. Anomaly detection based on zero-shot outlier synthesis and hierarchical feature distillation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(1):281–291, 2020.

[40] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[41] Dongming Feng and Maria Q Feng. Computer vision for shm of civil infrastructure: From dynamic response measurement to damage detection–a review. *Engineering Structures*, 156:105–117, 2018.

[42] GF Sirca Jr and H Adeli. System identification in structural engineering. *Scientia Iranica*, 19(6):1355–1364, 2012.

[43] Jean-Philippe Noël and Gaëtan Kerschen. Nonlinear system identification in structural dynamics: 10 more years of progress. *Mechanical Systems and Signal Processing*, 83:2–35, 2017.

[44] Xuyang Li, Hamed Bolandi, Talal Salem, Nizar Lajnef, and Vishnu Naresh Boddeti. Neuralsi: Structural parameter identification in nonlinear dynamical systems. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 332–348. Springer, 2023.

[45] Masoud Mirtaheri, Mojtaba Salkhordeh, and Masoud Mohammadgholiha. A system identification-based damage-detection method for gravity dams. *Shock and Vibration*, 2021:1–15, 2021.

[46] XG Hua, YQ Ni, ZQ Chen, and JM Ko. Structural damage detection of cable-stayed bridges using changes in cable forces and model updating. *Journal of structural engineering*, 135(9):1093–1106, 2009.

[47] Zhilu Lai and Satish Nagarajaiah. Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. *Mechanical Systems and Signal Processing*, 117:813–842, 2019.

[48] Mohammad Rezaiee-Pajand, Alireza Entezami, and Hassan Sarmadi. A sensitivity-based finite element model updating based on unconstrained optimization problem and regularized solution methods. *Structural Control and Health Monitoring*, 27(5):e2481, 2020.

[49] Tao Yin, Qing-Hui Jiang, and Ka-Veng Yuen. Vibration-based damage detection for structural connections using incomplete modal data by bayesian approach and model reduction technique. *Engineering Structures*, 132:260–277, 2017.

[50] Hassan Sarmadi, Alireza Entezami, and Mohammadhassan Daneshvar Khorram. Energy-based damage localization under ambient vibration and non-stationary signals by ensemble empirical mode decomposition and mahalanobis-squared distance. *Journal of Vibration and Control*, 26(11-12):1012–1027, 2020.

[51] Alireza Entezami, Hassan Sarmadi, Behshid Behkamal, and Stefano Mariani. Big data analytics and structural health monitoring: a statistical pattern recognition-based approach. *Sensors*, 20(8):2328, 2020.

[52] Alberto Diez, Nguyen Lu Dang Khoa, Mehrisadat Makki Alamdari, Yang Wang, Fang Chen, and Peter Runcie. A clustering approach for structural health monitoring on bridges. *Journal of Civil Structural Health Monitoring*, 6(3):429–445, 2016.

[53] Xiao-Mei Yang, Ting-Hua Yi, Chun-Xu Qu, Hong-Nan Li, and Hua Liu. Automated eigensystem realization algorithm for operational modal identification of bridge structures. *Journal of Aerospace Engineering*, 32(2):04018148, 2019.

[54] Rune Brincker, Lingmi Zhang, and Palle Andersen. Modal identification of output-only systems using frequency domain decomposition. *Smart Materials and Structures*, 10(3):441, 2001.

[55] Edwin Reynders and Guido De Roeck. Reference-based combined deterministic–stochastic subspace identification for experimental and operational modal analysis. *Mechanical Systems and Signal Processing*, 22(3):617–637, 2008.

[56] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[57] Kenneth F Alvin and KC Park. Second-order structural identification procedure via state-space-based system identification. *AIAA journal*, 32(2):397–406, 1994.

[58] Andrea Belleri, Babak Moaveni, and José I Restrepo. Damage assessment through structural identification of a three-story large-scale precast concrete structure. *Earthquake engineering & structural dynamics*, 43(1):61–76, 2014.

[59] Per Sjövall and Thomas Abrahamsson. Component system identification and state-space model synthesis. *Mechanical Systems and Signal Processing*, 21(7):2697–2714, 2007.

[60] Ting-Hua Yi, Xiao-Jun Yao, Chun-Xu Qu, and Hong-Nan Li. Clustering number determination for sparse component analysis during output-only modal identification. *Journal of Engineering Mechanics*, 145(1):04018122, 2019.

[61] Kaveh Karami, Pejman Fatehi, and Azad Yazdani. On-line system identification of structures using wavelet-hilbert transform and sparse component analysis. *Computer-Aided Civil and Infrastructure Engineering*, 35(8):870–886, 2020.

[62] Yongchao Yang and Satish Nagarajaiah. Output-only modal identification with limited sensors using sparse component analysis. *Journal of Sound and Vibration*, 332(19):4741–4765, 2013.

[63] Meiliang Wu and Andrew W Smyth. Application of the unscented kalman filter for real-time nonlinear structural system identification. *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures*, 14(7):971–990, 2007.

[64] Zongbo Xie and Jiuchao Feng. Real-time nonlinear structural system identification via iterated unscented kalman filter. *Mechanical systems and signal processing*, 28:309–322, 2012.

[65] Ying Lei, Dandan Xia, Kalil Erazo, and Satish Nagarajaiah. A novel unscented kalman filter for recursive state-input-system identification of nonlinear systems. *Mechanical Systems and Signal Processing*, 127:120–135, 2019.

[66] Wei-Xin Ren and Hua-Bing Chen. Finite element model updating in structural dynamics by using the response surface method. *Engineering structures*, 32(8):2455–2465, 2010.

[67] Nizar Faisal Alkayem, Maosen Cao, Yufeng Zhang, Mahmoud Bayat, and Zhongqing Su. Structural damage detection using finite element model updating with evolutionary algorithms: a survey. *Neural Computing and Applications*, 30:389–411, 2018.

[68] Suzana Ereiz, Ivan Duvnjak, and Javier Fernando Jiménez-Alonso. Review of finite element model updating methods for structural applications. In *Structures*, volume 41, pages 684–723. Elsevier, 2022.

[69] Nader M Okasha, Dan M Frangopol, and Andre D Orcesi. Automated finite element updating using strain data for the lifetime reliability assessment of bridges. *Reliability Engineering & System Safety*, 99:139–150, 2012.

[70] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[71] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[72] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature communications*, 12(1):6136, 2021.

[73] Tong Qin, Kailiang Wu, and Dongbin Xiu. Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395:620–635, 2019.

[74] Hojjat Adeli and Xiaomo Jiang. Dynamic fuzzy wavelet neural network model for structural system identification. *Journal of structural engineering*, 132(1):102–111, 2006.

[75] Ruiyang Zhang, Zhao Chen, Su Chen, Jingwei Zheng, Oral Büyüköztürk, and Hao Sun. Deep long short-term memory networks for nonlinear structural seismic response prediction. *Computers & Structures*, 220:55–68, 2019.

[76] Yu Wang. A new concept using lstm neural networks for dynamic system identification. In *2017 American control conference (ACC)*, pages 5324–5329. IEEE, 2017.

[77] Rih-Teng Wu and Mohammad R Jahanshahi. Deep convolutional neural network for structural dynamic response estimation and system identification. *Journal of Engineering Mechanics*, 145(1):04018125, 2019.

[78] Ruiyang Zhang, Yang Liu, and Hao Sun. Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling. *Engineering Structures*, 215:110704, 2020.

[79] Youqi Zhang, Yasunori Miyamori, Shuichi Mikami, and Takehiko Saito. Vibration-based structural state identification by a 1-dimensional convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, 34(9):822–839, 2019.

[80] Manuel A Roehrl, Thomas A Runkler, Veronika Brandtstetter, Michel Tokic, and Stefan Obermayer. Modeling system dynamics with physics-informed neural networks based on lagrangian mechanics. *IFAC-PapersOnLine*, 53(2):9195–9200, 2020.

[81] Marco Forgione and Dario Piga. Continuous-time system identification with neural networks: Model structures and fitting criteria. *European Journal of Control*, 59:69–81, 2021.

[82] Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560, 2021.

[83] N Sukumar and Ankit Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, 389:114333, 2022.

[84] Esmaeil Ghorbani, Oral Buyukozturk, and Young-Jin Cha. Hybrid output-only structural system identification using random decrement and kalman filter. *Mechanical Systems and Signal Processing*, 144:106977, 2020.

[85] Deepak Maurya, Sivadurgaprasad Chinta, Abhishek Sivaram, and Raghunathan Rengaswamy. Incorporating prior knowledge about structural constraints in model identification. *arXiv preprint arXiv:2007.04030*, 2020.

[86] Zhilu Lai, Charilaos Mylonas, Satish Nagarajaiah, and Eleni Chatzi. Structural identification with physics-informed neural ordinary differential equations. *Journal of Sound and Vibration*, 508:116196, 2021.

[87] Hassene Hasni, Amir H Alavi, Nizar Lajnef, Mohamed Abdelbarr, Sami F Masri, and Shantanu Chakrabartty. Self-powered piezo-floating-gate sensors for health monitoring of steel plates. *Engineering Structures*, 148:584–601, 2017.

[88] Marat Konkanov, Talal Salem, Pengcheng Jiao, Rimma Niyazbekova, and Nizar Lajnef. Environment-friendly, self-sensing concrete blended with byproduct wastes. *Sensors*, 20(7):1925, 2020.

[89] Hadi Salehi, Rigoberto Burgueño, Shantanu Chakrabartty, Nizar Lajnef, and Amir H Alavi. A comprehensive review of self-powered sensors in civil infrastructure: State-of-the-art and future research trends. *Engineering Structures*, 234:111963, 2021.

[90] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[91] Tianjun Zhang, Zhewei Yao, Amir Gholami, Joseph E Gonzalez, Kurt Keutzer, Michael W Mahoney, and George Biros. ANODEV2: A coupled neural ODE framework. *Advances in Neural Information Processing Systems*, 32, 2019.

[92] Hananeh Aliee, Fabian J Theis, and Niki Kilbertus. Beyond predictions in neural odes: Identification and interventions. *arXiv preprint arXiv:2106.12430*, 2021.

[93] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

[94] Marvin Höge, Andreas Scheidegger, Marco Baity-Jesi, Carlo Albert, and Fabrizio Fenicia. Improving hydrologic models for predictions and process understanding using neural odes. *Hydrology and Earth System Sciences*, 26(19):5085–5102, 2022.

[95] Carlos JG Rojas, Andreas Dengel, and Mateus Dias Ribeiro. Reduced-order model for fluid flows via neural ordinary differential equations. *arXiv preprint arXiv:2102.02248*, 2021.

[96] Jeehyun Hwang, Jeongwhan Choi, Hwangyong Choi, Kookjin Lee, Dongeun Lee, and Noseong Park. Climate modeling with neural diffusion equations. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 230–239. IEEE, 2021.

[97] Opeoluwa Owoyele and Pinaki Pal. Chemnode: A neural ordinary differential equations framework for efficient chemical kinetic solvers. *Energy and AI*, 7:100118, 2022.

[98] Johannes Brandstetter, Max Welling, and Daniel E Worrall. Lie point symmetry data augmentation for neural pde solvers. *arXiv preprint arXiv:2202.07643*, 2022.

[99] Kirill Zubov, Zoe McCarthy, Yingbo Ma, Francesco Calisto, Valerio Pagliarino, Simone Azeglio, Luca Bottero, Emmanuel Luján, Valentin Sulzer, Ashutosh Bharambe, et al. Neuralpde: Automating physics-informed neural networks (pinns) with error approximations. *arXiv preprint arXiv:2107.09443*, 2021.

[100] Andrzej Dulny, Andreas Hotho, and Anna Krause. Neuralpde: Modelling dynamical systems from data. *arXiv preprint arXiv:2111.07671*, 2021.

[101] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

[102] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

[103] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR, 2020.

[104] Masanobu Horie and Naoto Mitsume. Physics-embedded neural networks: E (n)-equivariant graph neural pde solvers. *arXiv preprint arXiv:2205.11912*, 2022.

[105] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[106] Hamed Bolandi, Gautam Sreekumar, Xuyang Li, Nizar Lajnef, and Vishnu Naresh Boddeti. Physics informed neural network for dynamic stress prediction. *arXiv preprint*

arXiv:2211.16190, 2022.

[107] Hamed Bolandi, Gautam Sreekumar, Xuyang Li, Nizar Lajnef, and Vishnu Naresh Boddeti. Neuro-dynastress: Predicting dynamic stress distributions in structural components. *arXiv preprint arXiv:2301.02580*, 2022.

[108] Xuan Li and Wei Zhang. Physics-informed deep learning model in wind turbine response prediction. *Renewable Energy*, 185:932–944, 2022.

[109] Peng Ni, Limin Sun, Jipeng Yang, and Yixian Li. Multi-end physics-informed deep learning for seismic response estimation. *Sensors*, 22(10):3697, 2022.

[110] Mikkel L Bødker, Mathieu Bauchy, Tao Du, John C Mauro, and Morten M Smedskjaer. Predicting glass structure by physics-informed machine learning. *npj Computational Materials*, 8(1):192, 2022.

[111] Karthik Kashinath, M Mustafa, Adrian Albert, JL Wu, C Jiang, Soheil Esmaeilzadeh, Kamyar Azizzadenesheli, R Wang, A Chattopadhyay, A Singh, et al. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194):20200093, 2021.

[112] Zhaobin Mo, Rongye Shi, and Xuan Di. A physics-informed deep learning paradigm for car-following models. *Transportation research part C: emerging technologies*, 130:103240, 2021.

[113] Hamidreza Eivazi and Ricardo Vinuesa. Physics-informed deep-learning applications to experimental fluid mechanics. *arXiv preprint arXiv:2203.15402*, 2022.

[114] Oameed Noakoasteen, Shu Wang, Zhen Peng, and Christos Christodoulou. Physics-informed deep neural networks for transient electromagnetic analysis. *IEEE Open Journal of Antennas and Propagation*, 1:404–412, 2020.

[115] Shahram Pezeshk, Charles V Camp, Ali Kashani, Mohsen Akhani, et al. Data analyses from seismic instrumentation installed on the I-40 bridge. *Tennessee. Department of Transportation*, 2021.

[116] Haley Carnahan. Pittsburgh bridge collapse emphasizes need for bridge repairs. *Journal of Protective Coatings & Linings*, 39(7):6–7, 2022.

[117] Lorenzo Capineri and Andrea Bulletti. Ultrasonic guided-waves sensors and integrated structural health monitoring systems for impact detection and localization: A review. *Sensors*, 21(9):2929, 2021.

[118] ZhiFeng Tang, XiaoDong Sui, YuanFeng Duan, Pengfei Zhang, and Chung Bang Yun. Guided wave-based cable damage detection using wave energy transmission and reflection.

Structural Control and Health Monitoring, 28(5):e2688, 2021.

[119] Cliff J Lissenden, Yang Liu, and Joseph L Rose. Use of non-linear ultrasonic guided waves for early damage detection. *Insight-Non-Destructive Testing and Condition Monitoring*, 57(4):206–211, 2015.

[120] Neha Chandarana, Daniel Martinez Sanchez, Constantinos Soutis, and Matthieu Gresil. Early damage detection in composites during fabrication and mechanical testing. *Materials*, 10(7):685, 2017.

[121] Shuzhi Song, Xin Zhang, Yongqi Chang, and Yi Shen. An improved structural health monitoring method utilizing sparse representation for acoustic emission signals in rails. *IEEE Transactions on Instrumentation and Measurement*, 72:1–11, 2022.

[122] Mohammad Hassan Daneshvar, Alireza Gharighoran, Seyed Alireza Zareei, and Abbas Karamodin. Early damage detection under massive data via innovative hybrid methods: application to a large-scale cable-stayed bridge. *Structure and Infrastructure Engineering*, 17(7):902–920, 2021.

[123] Norhisham Bakhary, Hong Hao, and Andrew J Deeks. Substructuring technique for damage detection using statistical multi-stage artificial neural network. *Advances in Structural Engineering*, 13(4):619–639, 2010.

[124] Michele Betti, Luca Facchini, and Paolo Biagini. Damage detection on a three-storey steel frame using artificial neural networks and genetic algorithms. *Meccanica*, 50:875–886, 2015.

[125] Zhenkun Li, Weiwei Lin, and Youqi Zhang. Real-time drive-by bridge damage detection using deep auto-encoder. In *Structures*, volume 47, pages 1167–1181, 2023.

[126] Akbar Esfandiari, Mansureh-Sadat Nabiyan, and Fayaz R Rofooei. Structural damage detection using principal component analysis of frequency response function data. *Structural Control and Health Monitoring*, 27(7):e2550, 2020.

[127] Shruti Sawant, Amit Sethi, Sauvik Banerjee, and Siddharth Tallur. Unsupervised learning framework for temperature compensated damage identification and localization in ultrasonic guided wave SHM with transfer learning. *Ultrasonics*, page 106931, 2023.

[128] Sakib Mahmud Khan, Sez Atamturktur, Mashrur Chowdhury, and Mizanur Rahman. Integration of structural health monitoring and intelligent transportation systems for bridge condition assessment: Current status and future direction. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2107–2122, 2016.

[129] Kejie Jiang, Qiang Han, Xiuli Du, and Pinghe Ni. A decentralized unsupervised structural condition diagnosis approach using deep auto-encoders. *Computer-Aided Civil and*

Infrastructure Engineering, 36(6):711–732, 2021.

[130] Duo Ma, Hongyuan Fang, Niannian Wang, Binghan Xue, Jiaxiu Dong, and Fu Wang. A real-time crack detection algorithm for pavement based on cnn with multiple feature layers. *Road Materials and Pavement Design*, 23(9):2115–2131, 2022.

[131] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.

[132] Hamed Bolandi, Gautam Sreekumar, Xuyang Li, Nizar Lajnef, and Vishnu Naresh Boddeti. Physics informed neural network for dynamic stress prediction. *Applied Intelligence*, 53(22):26313–26328, 2023.

[133] Fabio Parisi, Sergio Ruggieri, Ruggiero Lovreglio, Maria Pia Fanti, and Giuseppina Uva. On the use of mechanics-informed models to structural engineering systems: Application of graph neural networks for structural analysis. In *Structures*, volume 59, page 105712, 2024.

[134] Ling-Han Song, Chen Wang, Jian-Sheng Fan, and Hong-Ming Lu. Elastic structural analysis based on graph neural network without labeled data. *Computer-Aided Civil and Infrastructure Engineering*, 38(10):1307–1323, 2023.

[135] Yuan-Tung Chou, Wei-Tze Chang, Jimmy G Jean, Kai-Hung Chang, Yin-Nan Huang, and Chuin-Shan Chen. Structgnn: An efficient graph neural network framework for static structural analysis. *Computers & Structures*, 299:107385, 2024.

[136] Stefan Bloemheuvel, Jurgen van den Hoogen, and Martin Atzmueller. A computational framework for modeling complex sensor network data using graph signal processing and graph neural networks in structural health monitoring. *Applied Network Science*, 6(1):97, 2021.

[137] Pengming Zhan, Xianrong Qin, Qing Zhang, and Yuantao Sun. A novel structural damage detection method via multisensor spatial–temporal graph-based features and deep graph convolutional network. *IEEE Transactions on Instrumentation and Measurement*, 72:1–14, 2023.

[138] María P González and José L Zapico. Seismic damage identification in buildings using neural networks and modal data. *Computers & structures*, 86(3-5):416–426, 2008.

[139] Mahindra Rautela and S Gopalakrishnan. Ultrasonic guided wave based structural damage detection and localization using model assisted convolutional and recurrent neural networks. *Expert Systems with Applications*, 167:114189, 2021.

[140] You-Lin Xu, Jian-Fu Lin, Sheng Zhan, and Feng-Yang Wang. Multistage damage detection of a transmission tower: Numerical investigation and experimental validation. *Structural*

Control and Health Monitoring, 26(8):e2366, 2019.

[141] Nur Sila Gulgec, Martin Takáč, and Shamim N Pakzad. Convolutional neural network approach for robust structural damage detection and localization. *Journal of computing in civil engineering*, 33(3):04019005, 2019.

[142] Samim Mustafa, Hidehiko Sekiya, and Shuichi Hirano. Evaluation of fatigue damage in steel girder bridges using displacement influence lines. In *Structures*, volume 53, pages 1160–1171. Elsevier, 2023.

[143] Xuyang Li, Talal Salem, Hamed Bolandi, Vishnu Boddeti, and Nizar Lajnef. Methods for the rapid detection of boundary condition variations in structural systems. In *Smart Materials, Adaptive Structures and Intelligent Systems*, volume 86274, page V001T05A004, 2022.

[144] Bitao Wu, Gang Wu, Caiqian Yang, and Yi He. Damage identification method for continuous girder bridges based on spatially-distributed long-gauge strain sensing under moving loads. *Mechanical Systems and Signal Processing*, 104:415–435, 2018.

[145] Tangqing Li, Zheng Wang, Siying Liu, and Wen-Yan Lin. Deep unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3636–3645, 2021.

[146] Shi-Zhi Chen, Gang Wu, and De-Cheng Feng. Damage detection of highway bridges based on long-gauge strain response under stochastic traffic flow. *Mechanical Systems and Signal Processing*, 127:551–572, 2019.

[147] Md Riasat Azim and Mustafa Gül. Data-driven damage identification technique for steel truss railroad bridges utilizing principal component analysis of strain response. *Structure and Infrastructure Engineering*, 17(8):1019–1035, 2021.

[148] Md Riasat Azim and Mustafa Gül. Development of a novel damage detection framework for truss railway bridges using operational acceleration and strain response. *Vibration*, 4(2):422–443, 2021.

[149] Zahra Rastin, Gholamreza Ghodrati Amiri, and Ehsan Darvishan. Unsupervised structural damage detection technique based on a deep convolutional autoencoder. *Shock and Vibration*, 2021:1–11, 2021.

[150] Valentina Giglioni, Ilaria Venanzi, Valentina Poggioni, Alfredo Milani, and Filippo Ubertini. Autoencoders for unsupervised real-time bridge health assessment. *Computer-Aided Civil and Infrastructure Engineering*, 38(8):959–974, 2023.

[151] Faramarz Khoshnoudian, Saeid Talaei, and Milad Fallahian. Structural damage detection using FRF data, 2D-PCA, artificial neural networks and imperialist competitive algorithm simultaneously. *International Journal of Structural Stability and Dynamics*, 17(07):1750073,

2017.

[152] Shancheng Cao, Huajiang Ouyang, and Li Cheng. Baseline-free adaptive damage localization of plate-type structures by using robust pca and gaussian smoothing. *Mechanical Systems and Signal Processing*, 122:232–246, 2019.

[153] Debarshi Sen, Kalil Erazo, Wei Zhang, Satish Nagarajaiah, and Limin Sun. On the effectiveness of principal component analysis for decoupling structural damage and environmental effects in bridge structures. *Journal of Sound and Vibration*, 457:280–298, 2019.

[154] Zhiming Zhang and Chao Sun. Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating. *Structural Health Monitoring*, 20(4):1675–1688, 2021.

[155] Shengyuan Zhang, Chun Min Li, and Wenjing Ye. Damage localization in plate-like structures using time-varying feature and one-dimensional convolutional neural network. *Mechanical Systems and Signal Processing*, 147:107107, 2021.

[156] Sergio Cofre-Martel, Philip Kobrich, Enrique Lopez Droguett, and Viviana Meruane. Deep convolutional neural network-based structural damage localization and quantification using transmissibility data. *Shock and Vibration*, 2019, 2019.

[157] Amir H Alavi, Hassene Hasni, Nizar Lajnef, Karim Chatti, and Fred Faridazar. An intelligent structural damage detection approach based on self-powered wireless sensor data. *Automation in Construction*, 62:24–44, 2016.

[158] Amir H Alavi, Hassene Hasni, Nizar Lajnef, Karim Chatti, and Fred Faridazar. Damage detection using self-powered wireless sensor data: An evolutionary approach. *Measurement*, 82:254–283, 2016.

[159] H Hasni, AH Alavi, K Chatti, and N Lajnef. Continuous health monitoring of asphalt concrete pavements using surface-mounted battery-free wireless sensors. In *Bearing Capacity of Roads, Railways and Airfields*, pages 637–643. CRC Press, 2017.

[160] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[161] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972.

[162] Viv Bewick, Liz Cheek, and Jonathan Ball. Statistics review 13: receiver operating characteristic curves. *Critical Care*, 8(6):1–5, 2004.

[163] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *IEEE International*

Conference on Data Mining, pages 413–422, 2008.

[164] Tomáš Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102:275–304, 2016.

[165] Spiros Papadimitriou, Jimeng Sun, and Christos Faloutsos. Streaming pattern discovery in multiple time-series. Carnegie Mellon University, 2005.

[166] Ilari Shafer, Kai Ren, Vishnu Naresh Boddeti, Yoshihisa Abe, Gregory R Ganger, and Christos Faloutsos. Rainmon: An integrated approach to mining bursty timeseries monitoring data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1158–1166, 2012.

[167] Ka-Veng Yuen, Siu Kui Au, and James L Beck. Two-stage structural health monitoring approach for phase i benchmark studies. *Journal of Engineering Mechanics*, 130(1):16–33, 2004.

[168] Abdollah Bagheri, Osman E Ozbulut, and Devin K Harris. Structural system identification based on variational mode decomposition. *Journal of Sound and Vibration*, 417:182–197, 2018.

[169] Sertaç Tuhta and Furkan Günday. Multi input multi output system identification of concrete pavement using n4sid. *International Journal of Interdisciplinary Innovative Research Development*, 4(1):41–47, 2019.

[170] Xinyuan Zhou, Wei He, Yaoxiang Zeng, and Yahui Zhang. A semi-analytical method for moving force identification of bridge structures based on the discrete cosine transform and fem. *Mechanical Systems and Signal Processing*, 180:109444, 2022.

[171] B Banerjee, D Roy, and RM Vasu. Self-regularized pseudo time-marching schemes for structural system identification with static measurements. *International Journal for Numerical Methods in Engineering*, 82(7):896–916, 2010.

[172] Alireza Entezami, Hashem Shariatmadar, and Hassan Sarmadi. Structural damage detection by a new iterative regularization method and an improved sensitivity function. *Journal of Sound and Vibration*, 399:285–307, 2017.

[173] MI Modebei, RB Adeniyi, and SN JATOR. Numerical approximations of fourth-order pdes using block unification method. *Journal of the Nigerian Mathematical Society*, 39(1):47–68, 2020.

[174] Folake Oyedigba Akinpelu. The response of viscously damped euler-bernoulli beam to uniform partially distributed moving loads. *Applied Mathematics*, 3(3):199–204, 2012.

[175] Talal Salem, Pengcheng Jiao, Imen Zaabar, Xuyang Li, Ronghua Zhu, and Nizar Lajnef.

Functionally graded materials beams subjected to bilateral constraints: Structural instability and material topology. *International Journal of Mechanical Sciences*, 194:106218, 2021.

[176] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[177] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, 2018.

[178] Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. 1994.

[179] Yingbo Ma, Shashi Gowda, Ranjan Anantharaman, Chris Laughman, Viral Shah, and Chris Rackauckas. ModelingToolkit: A composable graph transformation system for equation-based modeling. *arXiv preprint arXiv:2103.05244*, 2021.

[180] Yong Huang, Changsong Shao, Biao Wu, James L Beck, and Hui Li. State-of-the-art review on bayesian inference in structural system identification and damage assessment. *Advances in Structural Engineering*, 22(6):1329–1351, 2019.

[181] Keith Worden. *Nonlinearity in structural dynamics: detection, identification and modelling*. CRC Press, 2019.

[182] Roger Ghanem and Masanobu Shinozuka. Structural-system identification. i: Theory. *Journal of Engineering Mechanics*, 121(2):255–264, 1995.

[183] Johan Schoukens and Lennart Ljung. Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, 39(6):28–99, 2019.

[184] H Tran-Ngoc, Leqia He, Edwin Reynders, Samir Khatir, T Le-Xuan, Guido De Roeck, T Bui-Tien, and M Abdel Wahab. An efficient approach to model updating for a multispan railway bridge using orthogonal diagonalization combined with improved particle swarm optimization. *Journal of Sound and Vibration*, 476:115315, 2020.

[185] Eleni N Chatzi and Andrew W Smyth. The unscented kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing. *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures*, 16(1):99–123, 2009.

[186] Xiao-Jun Yao, Ting-Hua Yi, Shao-Wei Zhao, Chun-Xu Qu, and Hua Liu. Fully automated operational modal identification using continuously monitoring data of bridge structures. *Journal of Performance of Constructed Facilities*, 35(5):04021041, 2021.

[187] Han Gao, Luning Sun, and Jian-Xun Wang. Phygeonet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular do-

main. *Journal of Computational Physics*, 428:110079, 2021.

[188] Uri M Ascher, Steven J Ruuth, and Raymond J Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2-3):151–167, 1997.

[189] Yuyu Song, Qiuhong Li, and Kai Xue. An analytical method for vibration analysis of arbitrarily shaped non-homogeneous orthotropic plates of variable thickness resting on winkler-pasternak foundation. *Composite Structures*, 296:115885, 2022.

[190] Ganesh Naik Guguloth, Baij Nath Singh, and Vinayak Ranjan. Free vibration analysis of simply supported rectangular plates. *Vibroengineering Procedia*, 29:270–273, 2019.

[191] Paul Dierckx. *Curve and surface fitting with splines*. Oxford University Press, 1995.

[192] Pengcheng Jiao, Wassim Borchani, Amir H Alavi, Hassene Hasni, and Nizar Lajnef. An energy harvesting and damage sensing solution based on postbuckling response of nonuniform cross-section beams. *Structural Control and Health Monitoring*, 25(1):e2052, 2018.

[193] A Kh Baiburin. Errors, defects and safety control at construction stage. *Procedia engineering*, 206:807–813, 2017.

[194] Clara Herrero Martin, Alon Oved, Rasheda A Chowdhury, Elisabeth Ullmann, Nicholas S Peters, Anil A Bharath, and Marta Varela. Ep-pinns: Cardiac electrophysiology characterisation using physics-informed neural networks. *Frontiers in Cardiovascular Medicine*, 8:768419, 2022.

[195] Konstantinos Ntagiantas, Eduardo Pignatelli, Nicholas S Peters, Chris D Cantwell, Rasheda A Chowdhury, and Anil A Bharath. Estimation of fibre architecture and scar in myocardial tissue using electrograms: An in-silico study. *Biomedical Signal Processing and Control*, 89:105746, 2024.

[196] Gunther Steenackers and Patrick Guillaume. Finite element model updating taking into account the uncertainty on the modal parameters estimates. *Journal of Sound and vibration*, 296(4-5):919–934, 2006.

[197] Hamed Ebrahimian, Rodrigo Astroza, Joel P Conte, and Raymond A de Callafon. Nonlinear finite element model updating for damage identification of civil structures using batch bayesian estimation. *Mechanical Systems and Signal Processing*, 84:194–222, 2017.

[198] Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

[199] Yan Ji, Xiaokun Jiang, and Lijuan Wan. Hierarchical least squares parameter estimation

algorithm for two-input hammerstein finite impulse response systems. *Journal of the Franklin Institute*, 357(8):5019–5032, 2020.

[200] Meihang Li and Ximei Liu. Maximum likelihood least squares based iterative estimation for a class of bilinear systems using the data filtering technique. *International Journal of Control, Automation and Systems*, 18(6):1581–1592, 2020.

[201] Devyani Varshney, Mani Bhushan, and Sachin C Patwardhan. State and parameter estimation using extended kitanidis kalman filter. *Journal of Process Control*, 76:98–111, 2019.

[202] Monowar Hossain, ME Haque, and Mohammad Taufiqul Arif. Kalman filtering techniques for the online model parameters and state of charge estimation of the li-ion batteries: A comparative analysis. *Journal of Energy Storage*, 51:104174, 2022.

[203] Wenbo Zhang and Wei Gu. Parameter estimation for several types of linear partial differential equations based on gaussian processes. *Fractal and Fractional*, 6(8):433, 2022.

[204] Zhongwei Deng, Xiaosong Hu, Xianke Lin, Yunhong Che, Le Xu, and Wenchao Guo. Data-driven state of charge estimation for lithium-ion battery packs based on gaussian process regression. *Energy*, 205:118000, 2020.

[205] Xiaoyu Li, Changgui Yuan, Xiaohui Li, and Zhenpo Wang. State of health estimation for li-ion battery using incremental capacity analysis and gaussian process regression. *Energy*, 190:116467, 2020.

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

[207] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 2021.

[208] QiZhi He, David Barajas-Solano, Guzel Tartakovsky, and Alexandre M Tartakovsky. Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Advances in Water Resources*, 141:103610, 2020.

[209] Shuai Zhao, Yingzhou Peng, Yi Zhang, and Huai Wang. Parameter estimation of power electronic converters with physics-informed machine learning. *IEEE Transactions on Power Electronics*, 37(10):11567–11578, 2022.

[210] QiZhi He, Panos Stinis, and Alexandre M Tartakovsky. Physics-constrained deep neural network method for estimating parameters in a redox flow battery. *Journal of Power Sources*, 528:231147, 2022.

[211] Alexandre M Tartakovsky, C Ortiz Marrero, Paris Perdikaris, Guzel D Tartakovsky, and David Barajas-Solano. Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resources Research*, 56(5):e2019WR026731, 2020.

[212] Karan Taneja, Xiaolong He, QiZhi He, Xinlun Zhao, Yun-An Lin, Kenneth J Loh, and Jiun-Shyan Chen. A feature-encoded physics-informed parameter identification neural network for musculoskeletal systems. *Journal of biomechanical engineering*, 144(12):121006, 2022.

[213] Ramakrishna Tipireddy, David A Barajas-Solano, and Alexandre M Tartakovsky. Conditional karhunen-loeve expansion for uncertainty quantification and active learning in partial differential equation models. *Journal of Computational Physics*, 418:109604, 2020.

[214] Alexandre M Tartakovsky, David A Barajas-Solano, and Qizhi He. Physics-informed machine learning with conditional karhunen-loève expansions. *Journal of Computational Physics*, 426:109904, 2021.

[215] Long Nguyen-Tuan, Tom Lahmer, Maria Datcheva, Eugenia Stoimenova, and Tom Schanz. A novel parameter identification approach for buffer elements involving complex coupled thermo-hydro-mechanical analyses. *Computers and Geotechnics*, 76:23–32, 2016.

[216] Long Nguyen-Tuan, Tom Schanz, Maria Datcheva, and Eugenia Stoimenova. Parameter identification for a thermo-hydro-mechanical model of the buffer material: Stochastic based back analysis. *Numerical Methods in Geotechnical Engineering*, 1:1001–1006, 2014.

[217] Chirag Mevawala, Xinwei Bai, Debangsu Bhattacharyya, and Jianli Hu. Dynamic data reconciliation, parameter estimation, and multi-scale, multi-physics modeling of the microwave-assisted methane dehydroaromatization process. *Chemical Engineering Science*, 239:116624, 2021.

[218] Xuyang Li, Hamed Bolandi, Talal Salem, Nizar Lajnef, and Vishnu Naresh Boddeti. Neuralsi: Structural parameter identification in nonlinear dynamical systems. In *European Conference on Computer Vision*, pages 332–348. Springer, 2022.

[219] Chris Rackauckas, Mike Innes, Yingbo Ma, Jesse Bettencourt, Lyndon White, and Vaibhav Dixit. Diffeqflux. jl-a julia library for neural differential equations. *arXiv preprint arXiv:1902.02376*, 2019.

[220] JL Randall. Finite difference methods for differential equations. *A Math*, 585, 2005.

[221] Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

[222] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

[223] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[224] Bicheng Yan, Dylan Robert Harp, Bailian Chen, and Rajesh Pawar. A physics-constrained deep learning model for simulating multiphase flow in 3d heterogeneous porous media. *Fuel*, 313:122693, 2022.

[225] Bicheng Yan, Dylan Robert Harp, Bailian Chen, Hussein Hoteit, and Rajesh J Pawar. A gradient-based deep neural network model for simulating multiphase flow in porous media. *Journal of Computational Physics*, 463:111277, 2022.

[226] Mohammad Mahdi Rajabi, Mohammad Reza Hajizadeh Javaran, Amadou-oury Bah, Gabriel Frey, Florence Le Ber, François Lehmann, and Marwan Fahs. Analyzing the efficiency and robustness of deep convolutional neural networks for modeling natural convection in heterogeneous porous media. *International Journal of Heat and Mass Transfer*, 183:122131, 2022.

[227] Gege Wen, Catherine Hay, and Sally M Benson. Ccsnet: a deep learning modeling suite for co2 storage. *Advances in Water Resources*, 155:104009, 2021.

[228] Jianchun Xu, Qirun Fu, and Hangyu Li. A novel deep learning-based automatic search workflow for co2 sequestration surrogate flow models. *Fuel*, 354:129353, 2023.

[229] Honghui Du, Ze Zhao, Haojia Cheng, Jinhui Yan, and QiZhi He. Modeling density-driven flow in porous media by physics-informed neural networks for co2 sequestration. *Computers and Geotechnics*, 159:105433, 2023.

[230] Parisa Shokouhi, Vikas Kumar, Sumedha Prathipati, Seyyed A Hosseini, Clyde Lee Giles, and Daniel Kifer. Physics-informed deep learning for prediction of co2 storage site response. *Journal of Contaminant Hydrology*, 241:103835, 2021.

[231] QiZhi He and Alexandre M Tartakovsky. Physics-informed neural network method for forward and backward advection-dispersion equations. *Water Resources Research*, 57(7):e2020WR029479, 2021.

[232] Michael Fienen, R Hunt, D Krabbenhoft, and Tom Clemo. Obtaining parsimonious hydraulic conductivity fields using head and transport observations: A bayesian geostatistical parameter estimation approach. *Water resources research*, 45(8), 2009.

[233] Haiyi Wu and Rui Qiao. Physics-constrained deep learning for data assimilation of subsurface transport. *Energy and AI*, 3:100044, 2021.

[234] Victoriya Kashtanova, Mihaela Pop, Ibrahim Ayed, Patrick Gallinari, and Maxime Sermesant. Simultaneous data assimilation and cardiac electrophysiology model correction using

differentiable physics and deep learning. *Interface Focus*, 13(6):20230043, 2023.

[235] Victoriya Kashtanova, Mihaela Pop, Ibrahim Ayed, Patrick Gallinari, and Maxime Sermesant. Aphyn-ep: Physics-based deep learning framework to learn and forecast cardiac electrophysiology dynamics. In *International Workshop on Statistical Atlases and Computational Models of the Heart*, pages 190–199. Springer, 2022.

[236] Yan Barbosa Werneck, Rodrigo Weber dos Santos, Bernardo Martins Rocha, and Rafael Sachetto Oliveira. Replacing the fitzhugh-nagumo electrophysiology model by physics-informed neural networks. In *International Conference on Computational Science*, pages 699–713. Springer, 2023.

[237] Md Shakil Zaman, Jwala Dhamala, Pradeep Bajracharya, John L Sapp, B Milan Horácek, Katherine C Wu, Natalia A Trayanova, and Linwei Wang. Fast posterior estimation of cardiac electrophysiological model parameters via bayesian active learning. *Frontiers in Physiology*, 12:740306, 2021.

[238] Tadeusz Liszka and Janusz Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers & Structures*, 11(1-2):83–95, 1980.

[239] KC Chung. A generalized finite-difference method for heat transfer problems of irregular geometries. *Numerical Heat Transfer*, 4(3):345–357, 1981.

[240] Mark L Trew, Bruce H Smaill, David P Bullivant, Peter J Hunter, and Andrew J Pullan. A generalized finite difference method for modeling cardiac electrical activation on arbitrary, irregular computational meshes. *Mathematical biosciences*, 198(2):169–189, 2005.