Machine learning for structural design models from the inverse problem perspective

by

Adrien Gallet

A thesis submitted to the Department of Civil and Structural Engineering
in Partial Fulfilment of the Requirements for the Degree of

Doctor of Philosophy

at the

University of Sheffield

March 2024

Declaration of Authorship

I, Adrien Gallet, declare that the work in this thesis, entitled, "Machine learning for structural design models from the inverse problem perspective", is my own unless referenced to the contrary in the text.

Signed: Adrien Gallet

Date: March 8th, 2024

Acknowledgements

No one can ever fully acknowledge all that which renders a productive venture possible. This could not be truer than of the work presented here. The following is my best attempt to acknowledge all those, which in no insignificant part, ultimately contributed to the success of this thesis. To those that expect to be mentioned, yet whom I may have missed: I thank you first. My un-acknowledgement of your contribution is without doubt at the fault of my memory.

I would like to thank first and foremost Danny Smyl. I could not have asked for a better primary supervisor to guide and support me through the journey of this self-proposed thesis for the last three and half years. A great supervisor has the best interest of their PhD student at heart, and you, Danny, have shown that in more ways than I can remember - from being available for periodic meet-ups regardless of what time zones we happened to find ourselves in, to providing excellent feedback, giving me plenty of opportunities to develop as an independent researcher and handing out rock-solid advice based on your own research experience – this thesis would not have achieved what it did without your supervision.

Similarly, I would also like to thank the rest of my supervisory team, Andrew Liew and Iman Hajirasouliha, for their continued support and enthusiasm, and for lending their ears and eyes when reviewing my works. Regardless of circumstances, the two of you found time for me when it was needed, and for that I am grateful. I would like to also thank Giacomo Torelli for the support during the closing stages of my PhD.

During my undergraduate, I was lucky to have been taught by the excellent civil and structural engineering faculty at the University of Sheffield, and I would like to give special mentions to Matthew Gilbert, Jonathan Black, Richard Collins, Colin Smith, Elisabeth Bowman, Sam Clarke, Sam Rigby, Maurizio Guadagnini, Richard Harpin, Luca Susmel, Rachel Horn and Georges Kesserwani whose teachings I especially enjoyed. I would also like to specifically mention my personal tutor Paul Hulbert, who played no insignificant part in all of this by encouraging me to continue studying civil and structural engineering during my first year of my undergraduate degree.

I have been fortunate enough to have been taught by incredible teachers throughout my secondary education, whose combined efforts gave me the opportunity to attend university in the first place. I would like to especially thank Arnaud Morin, David Robs, Philip Kurbis, J. Briffa, Garret Newell, Jordan Finch and Kirti Joshi for all they have taught me.

The PhD would not have been nearly as enjoyable without the discussions and daily Heartspace lunches with my fellow peers including Adam Dennis, Leanna Bradbury, Isuru Nanayakkara, Oswald Li, Arthur Van Lerberghe, Emma Clarke, Wenchao Li, Edward Whiteside, Lewis Tetlow, Ross Waddoups and Ella Mendham. On this note I would also like to also acknowledge all my friends and loved ones that have made the past couple of years ever so slightly easier to bear with.

Ultimately I would like to thank my siblings, Alexandre, Aurélia, and Anaïs, all of whom continue to inspire me in tackling whatever challenges have and may continue to lay ahead, and most important of all my parents, Wolfgang and Véronique Gallet, for their never-ending love and support. Your sacrifices and resolute efforts to educate and give me and my siblings all the opportunities we have been presented with has created the deep, inappreciably strong foundations on which our achievements will forever be built on. *Je vous aime tous und ich liebe euch alle.*

To all of you, and probably more: thank you.

Machine learning for structural design models from the inverse problem perspective

by

Adrien Gallet

Submitted to the Department of Civil and Structural Engineering
on March 8th, 2024 in Partial Fulfilment of the Requirements
for the Degree of Doctor of Philosophy.

Abstract

The introduction of novel technologies has historically led to fundamental changes in structural design practices. Despite the increasing prominence of machine learning techniques, the application for structural design models is sparse in current engineering practice when compared to structural analysis models. This publication-based thesis posits that structural design is in fact an inverse problem and presents a methodology to develop accurate, generalisable, and verifiable machine learned structural design models for continuous beam systems. Unique to this perspective is the interlink between structural analysis, design, and optimisation, providing a fundamental shift in engineering philosophy that is conventionally dominated by the forward-problem oriented field of engineering science.

A comprehensive literature review on the range of domains in which inverse problems have been identified in civil and structural engineering is presented first, covering applications in blast engineering, structural health monitoring, and digital twins. This review highlights the extensive use of machine learning in such domains as opposed to traditional optimisation based inverse solvers, and underlines some of the unique advantages which machine learning models provide to help address the current design challenges in industry.

The thesis subsequently embarks on an in-depth investigation to build a non-iterative machine learned structural design model for continuous beam systems within the three subsequent chapters. The first of these three chapters introduces the novel concept of an influence zone of continuous beam systems, which acts as a generalisable and heuristic estimator of the pertinent local loading information relevant for the design of individual members within such a system. The second chapter takes advantage of the influence zone to develop a generalisable, neural network based structural design model to predict cross-sectional properties of a continuous beam system of arbitrary system size. This chapter explicitly frames the machine learned design model from the inverse problem perspective, ideates an appropriate feature selection for the input parameter space, tests various architectures, and evaluates the accuracy of the model. The last chapter utilises a novel physics-informed neural network to evaluate the performance of such an architecture compared to the previously achieved benchmark and showcases the viability of determining

the physical accuracy of predictions during inference. This work concludes by reflecting on the novel contributions achieved by this investigation, and the specific scopes for future work.

Key words: machine learning, neural networks, structural design, design models, inverse problems, influence zones, physics informed neural networks.

Thesis supervisors

Danny Smyl, Assistant Professor, Georgia Institute of Technology
Andrew Liew, Computational Structural Engineer, Unipart Construction Technologies
Iman Hajirasouliha, Professor, University of Sheffield

Table of Contents

List of Figures

List of Tables

Part I

Preliminaries

Chapter 1

Introduction

1.1 Objectives and scope of work

For a field as old as engineering, there is a surprising lack of discussion on the philosophy of engineering design [1]. Unlike the sciences, which have historically enjoyed considerable deliberation by various philosophers and scientists alike, with well established concepts such as the "scientific method" [2] and principles such as Karl Popper's "falsification theory" [3], an equivalent in-depth discussion is lacking in engineering, despite researchers stressing the benefits such efforts could bring [4].

The thesis's primary assertion, presented here in a publication-based format, is that structural design may in fact be viewed as an inverse problem, and embarks an investigation to build rapid, accurate and verifiable machine learned structural design models. The investigation is motivated by the growing interest in developing digital and computational based approaches to address current challenges faced in the Architecture, Engineering and Construction (AEC) industry. The research aims to demonstrate the value of the inverse problem perspective for structural design, and to highlight the viability of machine-learned design models within this context. The thesis pursues the following objectives:

- Provide an overview of current structural design trends, and propose the use of machine learning as a potential solution to the industry's current design challenges.
- Review the various research areas within structural engineering in which inverse problems exist, and demonstrate that structural design may in fact be viewed as an ill-posed inverse problem.
- Identify novel mechanical concepts to help address some of the common shortcomings of past machine learned structural design models, specifically in relation to generalisability and accuracy.
- Demonstrate a methodology that incorporates the inverse problem perspective as well as the novel mechanical concepts to develop generalisable machine learned design models for continuous beam systems.
- Incorporate physics-informed neural networks (PINNs), a relatively novel network

archetype established in 2019 [5], and test their viability for machine learned structural design models.

This investigation fundamentally distinguishes itself from previous design research predominantly focused on structural optimisation. First and fore-most, the thesis makes the crucial distinction between structural analysis models, which have historically received considerable attention in literature, and structural design models, which is the focus of this thesis. This distinction is reflective of forward and inverse operators that exist in inverse problems. Second, with help of this perspective, several novel concepts are developed, including influence zones, polarity zones and polarity sequences. These novel concepts are then uniquely used to build a mechanics-driven feature selector that contribute towards building generalisable design models. Third, an open-source structural design dataset named CBeamXP [6] is generated, on which all machine learning models presented are trained on, providing other researchers and engineers the opportunity to review and contribute towards research efforts within this field. Lastly, novel physics-loss functions are formulated which allow physical verifiability of machine learning inferences outside of training.

1.2 Organisation of thesis

This publication based thesis consists out of four individual journal articles that were written during the three and a half year-long investigation. Each article is accompanied by a short foreword commentary and are tightly linked together as shown by Figure 1.1.

Chapter 2 provides some background knowledge to help contextualise the research. Chapter 3 is a literature review that proposes that structural design is an inverse problem, and demonstrates the wide range of existing inverse problems within the civil and structural engineering domain. Chapter 4 is a mechanics-focused research article that introduces the novel concept of influence zones. Influence zones play a fundamental role in Chapter 5 to build a generalisable machine learned structural design model from the inverse problem perspective. Chapter 6 then extends the previous design model by incorporating a physics-based loss function to ideate a neural network architecture capable of providing verification of predictions during inference. Chapter 7 summarises the key-findings and novelties of the research. Finally, a number of appendices have been made available either immediately after journal article chapters or at the end of the thesis to help provide further context on some of the outputs achieved during this investigation.

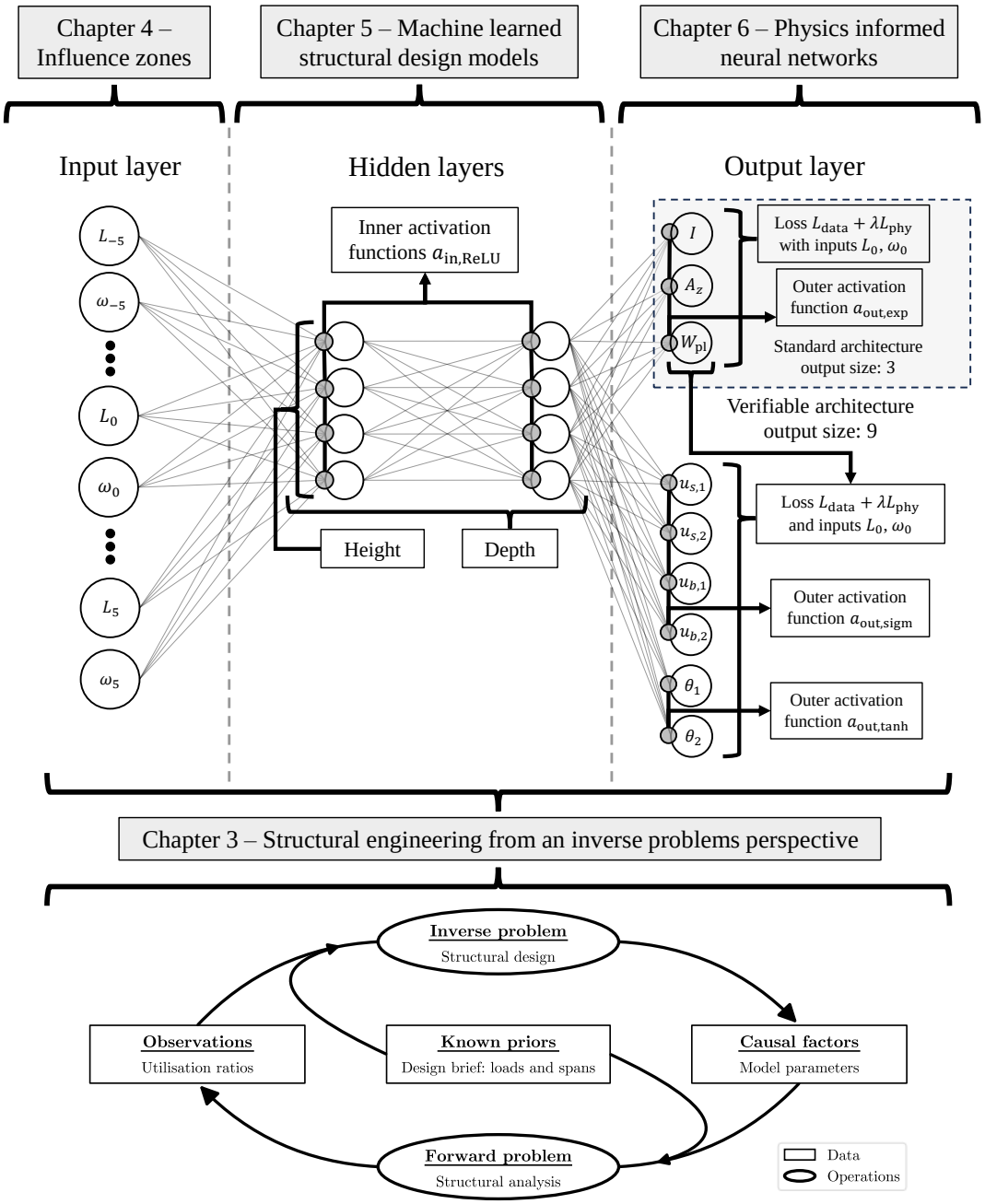


Figure 1.1: Overview of the publication-based thesis from the perspective of the neural network based structural design model. Each chapter is presented as a stand-alone journal article. Chapter 3 provides the philosophical backing behind developing machine learned inverse operators, Chapter 4 introduces the influence zone concept that plays a fundamental role in selecting the feature space, Chapter 5 develops the network architecture, and Chapter 6 investigates the impact from embedding physical knowledge into the neural network loss function.

Chapter 2

Background and contextualisation

2.1 Defining structural engineering design

Although the term "design" is used rather ubiquitously within engineering, both in industry and academia alike, "design" takes a surprisingly large number of meanings and a multitude of various definitions within literature. For example, one popular definition by A.J. Harris is "the determination of what is to be built and the instructions to the builders adequate for them to build it", whilst keeping in mind the factors of "function, economy and safety" [7]. William Addis, one of the few academics to have studied the history of engineering design, supplements Harris's definition centred around description and instruction with that of justification, whilst also stressing that design is principally a collection of unique human skills (from conceiving and proposing ideas, evaluating and choosing solutions, analysing structural behaviour and communicating design decisions) [8].

The lack of a rigorous and unifying definition for what "design" entails might suggest that the term encompasses such a large variety of various meanings that a single definition is unlikely. However, other researchers such as William Bulleit have suggested that it might highlight a lack of reflective and philosophical investigation into how engineering is done and why it is done that way [4]. For example, science, unlike engineering, has benefited from centuries of rigorous philosophical contemplation including Bacon's *Novum Organum* [9], Descartes *Discours de la Méthode* [2], Popper's *The Logic of Scientific Discovery* [3] and Kuhn's *Structure of Scientific Revolutions* [10]. Billy Koen argues that the lack of such investigations in engineering results in engineers identifying themselves in terms of the artefacts they create (aeroplanes, bridges, CPUs, transformers, etc.) instead of the design process which created them [11]. This would be the equivalent of physicists identifying themselves with the knowledge they establish (e.g. the laws of thermodynamics) instead of the scientific method which underpins all scientific disciplines (physics, chemistry, biology, psychology, etc.). As a result, each individual engineering discipline generally develops its own, specific understanding of what design signifies.

For example, within civil and structural engineering specifically, the *Institution of Structural Engineers* (IStructE) identifies the stages a project undergoes starting with the client brief and finishing with construction and handover of the structure as shown in Figure 2.1. While this diagram highlights the different stages that design comprises of

(scheme/conceptual/detailed), it only brings to the foreground the global design process, and hence provides little insight on the elements which a local design procedure would comprise of for various structural systems. Furthermore, the traditional, linear procurement route implied in this view is rarely the one which takes place in practice [12] and also ignores the often cyclical and repetitive nature which is inherent to design [13].

There is little doubt that today, in the 21st century, the design process is becoming increasingly complex due to client and societal demands imposed on structures, and policy decisions such as the UK's *Construction 2025* initiative are there to accelerate project deliveries and help motivate more sustainable, low embodied carbon solutions [15]. However, sometimes challenges are best addressed through research, and by properly appreciating the development of design methods through history and identifying the latest trends and challenges faced by the industry, the motivation and reasoning behind this thesis will be properly contextualised.

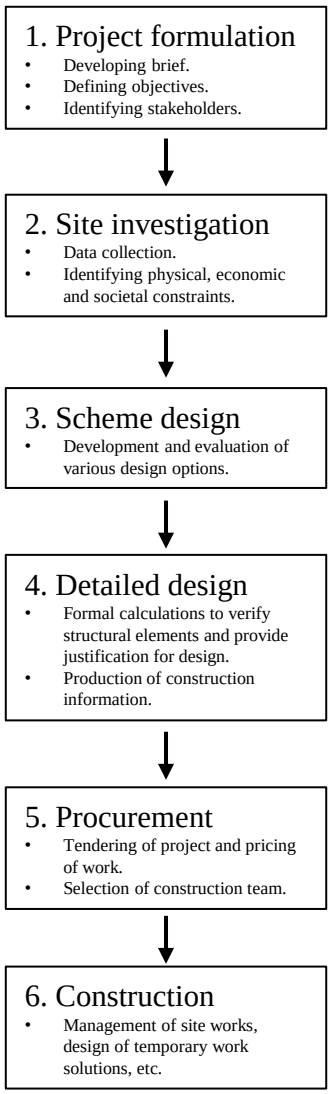


Figure 2.1: Global design process as specified within the *Structural design - the engineer's role* guide by the IStructE [14].

2.2 Historical development of design

A common misinterpretation of the historical development of design is to centre the discussion around the developments of engineering science and analysis techniques, which broadly speaking occurred between the 17th and 19th century as shown in Table 2.1 [16]. It is worth noting that there are plenty of historical structures which were built millennials ago that still stand today despite not relying on modern structural theories for design. These include notably the Great Pyramid of Giza (2600 BCE), the Greek Parthenon (438 BCE), the Roman Pantheon (126 CE), the hammer beam roof of Westminster Hall (1399), and numerous cathedrals and churches such as Brunelleschi's Cathedral Dome in Florence (1436) [17].

The study of design procedures throughout history suggests that design was not a matter of "trial and error", but instead followed a deliberate process, which typically was a reflection of the culture and thinking of that era [18]. Furthermore, throughout history various design revolutions took place as shown by Table 2.2, and should indicate that design procedures have undergone changes in the past as new materials, theories and technologies come about, a trend which is likely to extend into the future [8].

The last two major design revolutions, that of plastic design which was established in the 1930s and matured in the 1950s and led to the formulation of limit-state design [20], as well as that of the rise of the personal computer, which

Field of study	Years of significant development	Prominent scientists and engineers
Elasticity	1650s to 1810s	Hooke, Euler, Young
Beams	1630s to 1870s	Galileo, Mariotte, Bernouilli, Coulomb, Mohr
Columns	1750s to 1890s	Euler, Hodgkinson, Rankine, Moncrieff
Framed Structures	1850s to 1870s	Rankine, Maxwell, Ritter, Williot
Indeterminate Structures	1830s to 1910s	Clapeyron, Mohr, Castigliano, Maxwell, Mohr, Wilson, Maney, Beggs

Table 2.1: Overview of major developments in structural theories since the 1600s.

made techniques such as the *Direct Stiffness Method* and the subsequent *Finite Element Methods* accessible [21], principally reflect the status quo as to how design is carried out in industry today.

2.3 Design: current trends

Typical design workflow

The prevalence of computers and relatively in-expensive modelling software has pushed the Architecture, Engineering and Construction industry towards federated Building Information Models (BIMs), which places emphasis on developing digital tools which can be used during the entire lifecycle of a structure, for visualisation, scheduling, communication and collaboration among project stakeholders [22, 23]. Although technical issues such as software interoperability and computational costs, as well as non-technical issues ranging from costs, employee training, legal uncertainties, disruption of traditional workflows and contracts exist [22, 24, 25], there are a multitude of potential advantages and savings from BIM from enhanced exchange of information and knowledge management [26, 27], reducing project duration and cost [28, 29] and other benefits such as integrated clash detection [23]. These factors, amongst others, constitute of a collection of critical success factors through which the successful implementation of BIM can be measured against [22].

As a consequence of BIM, it is typical for the engineering workflows to revolve around a synchronously linked analysis model which automatically updates the BIM-model as changes, such as for the rationalisation of section sizes or in response to new client demands, are required. This workflow is summarised in Figure 2.2, and highlights the various steps involved in the process. Depending on the size of the model, the types of analysis required and the assumed section properties of the material, this workflow can often become computationally demanding and time consuming, especially when carried out manually [30]. Consequently, there has been a noticeable drive towards parametric models coupled with optimisation schemes to create more generative design procedures to efficiently evaluate complex, multi-objective design spaces [31].

Design revolution	Time period	Description
Greek	600-200 BCE	Numerical design procedures, using either Doric or Ionic design methods. Rationale was deeply rooted in the Pythagorean ideology of numbers, ratios and patterns.
Gothic	1100s to 1150s	Designs focused on linking Christian religious ideology to form and geometry.
Statics	Late 1600s	Use of parallelogram of forces and graphical methods translated into improved numerical design guidance for builders and formation of scientific academies.
Elastic	Early 1800s	Mathematical models of elastic behaviour of structures (stress and strain) begin to be used in design, as well as more advanced statical principles (such as the method of sections developed by Johann Schwedler).
Rankine	1840s	Introduces the idea of a factor of safety to address the issue of the idealisation made by structural theories, overcoming the reluctance of tradesmen of using theory in practice instead of precedence.
Plastic	1930s to 1950s	Response to the unrealistic and un-economic design constraints imposed through maximum stress design, ushering the concept of limit-states which still governs the code in the 21st century.
Computers	1970s	Ease and abundance of deterministic finite element analysis allows more complex structures to be built, yet also draws attention to the importance of fundamental design principles such as ductility and robustness for various material and building types.

Table 2.2: Major historical design revolutions that occurred in the West [8, 18, 19].

Parametric modelling, data interoperability and optimisation

Parametric modelling is the practice of developing models based on a set of key parameters which describes the structural solution (such as floor-heights, span-lengths, number of column rows, etc.) and allow a designer to interactively make updates to the model automatically. The two most common software packages used in industry include *Grasshopper* [32] and *Dynamo Revit* [33], both featuring visual programming interfaces which allow engineers to drag and drop various components to process data and generate geometries in real time. Third-party add-ons such as *Karamba3D* [34] or *Kiwi!3D* [35] often provide built-in structural analysis and design tools, which together with other packages often allow the engineering workflow as identified in Figure 2.2 to be tackled. However, industry surveys suggest that parametric modelling is typically reserved for early stages of design, specifically for design exploration [36], due to the lack of sufficient data compatibility and interoperability of software packages used during the later stages of the design [37].

The issue of data interoperability leads to losses of bi-directional links between various software packages, and was estimated to cost the American AEC and capital industry 2004 $15.8 billion USD annually in 2004 due to the additional resources spent on resolving

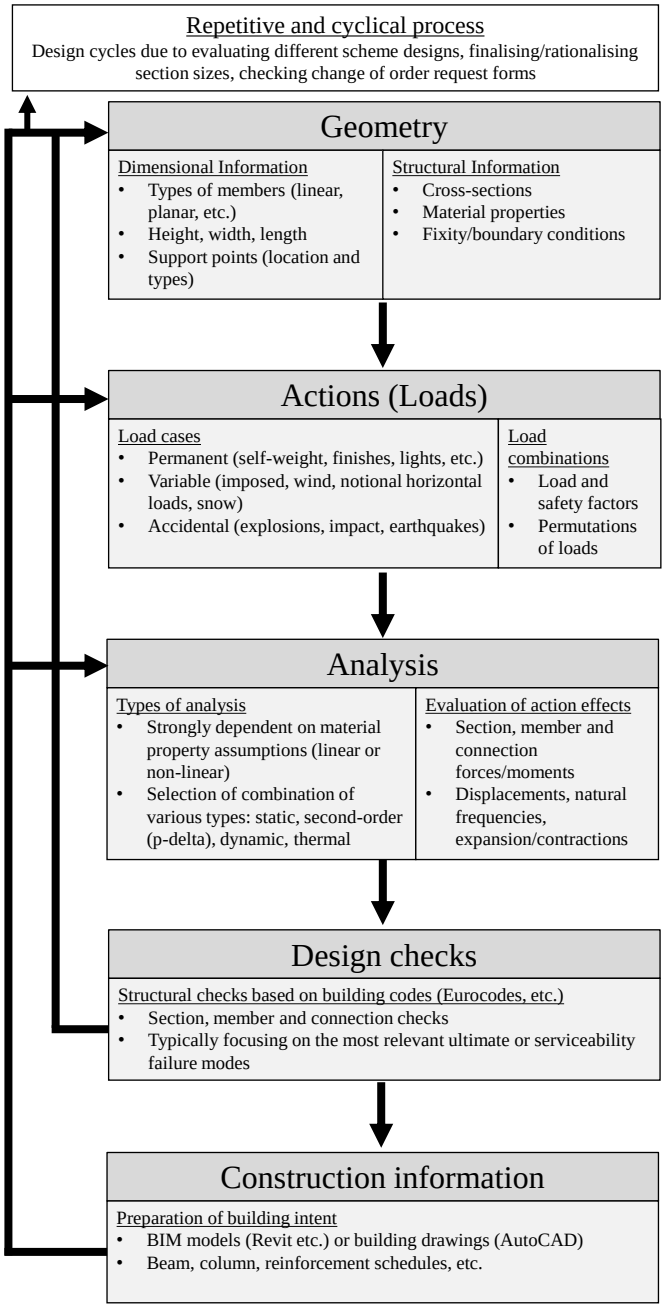
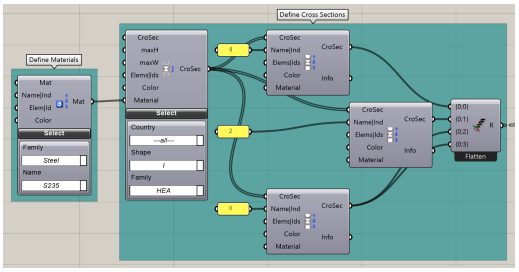
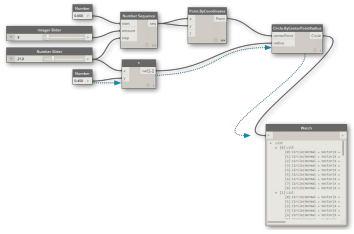


Figure 2.2: Typical engineering analysis and design workflow.

data exchange issues [38]. It was initially intended that the creation of a neutral data format based on the Industry Foundation classes (IFC) scheme would help to circumvent this problem, yet the diversity of software and their individual interpretation and implementation of the IFC scheme still results in errors during data exchange processes [39, 40]. During the late 2010s, instead of relying solely on the IFC scheme, various industry stake-



(a) Exemplary script from *Grasshopper*.



(b) Exemplary script from *Dynamo*.

Figure 2.3: Parametric modelling tools used in the AEC industry.

holders developed either open-source ot third-party packages to help alleviate this issue, the most popular being open-source packages such as *BhOM* [41], *Speckle* [42] or third-party solutions such as *Geometry Gym* [43]. Whilst these tools are a step forward in the right direction, they can at times create additional cost-burdens for designers, or in the case of open-source packages, rely on the support of the community for the latest adapters to be updated as new versions of software APIs are released.

Lastly, the combination of parametric tools and data-interoperability packages has ushered in a new era of single and multi-objective optimisation schemes which theoretically allow better exploration of design spaces, including size, shape and topology optimisation algorithms [44]. Optimisation schemes typically rely on either gradient-based (including Steepest Descent Methods, Conjugate Gradient Method or Sequential Linear Programming Algorithms [45]) or heuristic search methods (such as genetic or swarm algorithms [46]). The application and use of structural optimisation is prevalent in civil and structural engineering, including applications such as the optimisation of steel arch bridge designs [47], high rise buildings [48], as well as more exotic structures such as tensile domes [49]. Whilst these approaches clearly demonstrate the potential in reducing weights and costs in structural solutions, principal issues remain including the computational demands made by such optimisation schemes, specifically for more complex types of analysis and multi-objective performance criteria, as well as the need to possess a strong understanding of the inner workings and drawbacks of such schemes [50].

2.4 Modern design challenges

A useful insight on the challenges faced by the AEC industry can be found within the newly updated *2020 RIBA Plan of Work* guide [51]. The RIBA (Royal Institute of British Architects) plan of work is the industry's accepted framework for how construction projects, especially for those involving buildings, are carried out. Projects are separated into various stages, similar to the general "design procedure" outlined in 2.1. The RIBA plan of work outlines a linear design process as shown in Figure 2.4, yet the report is quick to acknowledge the various procurements and project management strategies employed which often leads to stages occurring concurrently, which brings us to the first modern design challenge.

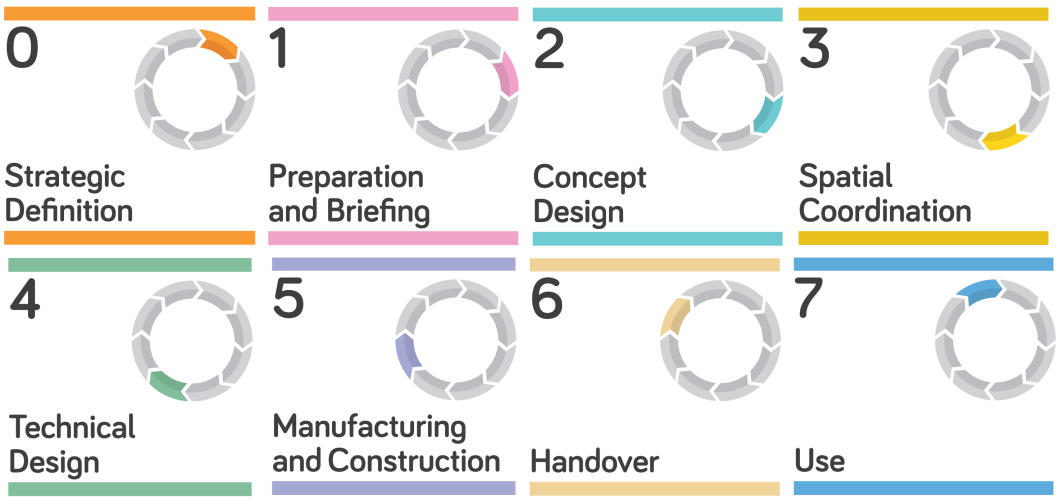


Figure 2.4: RIBA's 2020 Plan of Work stages [51].

Project management challenges

"Manufacturing and Construction", that is Stage 5 of RIBA's Plan of Work shown in Figure 2.4, is generally speaking the most time and labour intensive. Consequently, there is generally a concerted effort to start construction as early as viably possible in a project, typically by using non-standard procurement strategies such as "design and build", "management contracting" or "contractor-led" solutions. Whilst there are differences between these strategies in terms of who employs the design team, when exactly the construction team's involvement begins, and which party is ultimately responsible for the design [12], the key appreciation to be made is the additional pressures that these non-traditional procurement strategies exert on design engineers.

In order to be able to start construction earlier, it is necessary to correctly estimate what the ultimate loads and forces of the structure will be, specifically foundations and substructure elements, since those are generally speaking built first. Unfortunately, detailed engineering analysis can be highly resource intensive, and is therefore often only carried out at Stage 4 of a project. Consequently, since it is in the design teams' interest to minimise project and contract risks, conservative design assumptions which provide "slack" within calculations are often made. This leads to over-designs, not due to a lack of technical competence, but the lack of resources (time) to properly investigate the design. As non-standard procurement strategies continue to gain in prominence [52] and environmental pressures for lower embodied carbon-intensive structures mounts [53], developing methods which enable accurate engineering insight at an early stage of a design project becomes favourable.

Time-lag between design proposals and engineering analysis

Whilst there is no shortage of computational tools available to engineers, each software and design package requires expertise which is often outside the scope of the majority of

engineers in the modern workplace [54]. Consequently, when a design proposal or design change is made, it often takes a significant amount of time to yield any useful engineering feedback on the impact of that said change. Dale Sinclair (former Director of Innovation at *AECOM*) believes that instead of more complex analysis techniques, the industry is in need of tools which can provide real-time, evidence-based feedback on design decisions, especially at Stage 2 and Stage 3 of a project [51]. In order to over-come this time lag, early design decisions are typically taken based on engineering judgement, past experience and simple rules of thumb, which aside from leading to potentially conservative design decisions, also result in not properly engaging with the design space. Significant research has been made in parametric modelling and design optimisation as shown in section 2.3, yet the best way to integrate these techniques in actual design workflows remains an open research question.

Leveraging past design data

The rise of BIM explained in section 2.3 has led to the creation of enormous amounts of data, ranging from 3D models of the final structure, along with all of the documentation and calculations which took place during the design process. Unfortunately, to this date, this data has not been effectively taken advantage of within the AEC industry [55, 56]. To overcome the slow implementation of data-driven solutions, researchers are suggesting that a new cultural shift and mindset is required [57].

To this day, there are still no established practices on technologically leveraging past-design data to inform future design decisions. Instead, past design data is only employed qualitatively through the experience gained by engineers when they worked on past projects. Given the extended project times in constructions which can easily be in the excess of 5 years, as well as the natural progression of engineering designers into team managers as their career progresses, it is crucial for design consultancies to have some form of "knowledge management" system in place which prevents this "learned data" to go to waste [58]. This points towards an opportunity to investigate methods capable of using past data to streamline future design decisions instead of requiring engineers to go through the laborious process of starting from scratch.

2.5 Research motivation

The three challenges identified above can be summarised as a) project management pressures due to a lack of early engineering insight of projects, b) time-lags between a suggested design proposal and the associated engineering analysis and c) not having a technologically driven solution to leverage past design data all point towards needing a new methodology to streamline design workflows. Whilst data-interoperability packages can help to send data across various design software, and whilst generative design methods linked with real-time optimisation schemes can support the exploration of the design space, these approaches are often computationally expensive due to their deterministic nature, and requires specialists knowledge of particular software packages. Most importantly, optimisation schemes typically do not learn from their own results. These challenges, and the associated limitations of present-day solutions, motivated the research for alternative

solutions.

Starting in the 2010s, it was evident that machine learning techniques, and in particular deep learning methodologies, were making enormous strides in pattern recognition [59], decision making abilities [60] and natural language processing [61]. Subsequently, there was a growing awareness on the potential for machine learning to address some of the aforementioned challenges [62, 63]. Machine learning models could help address all three challenges, namely leverage past design data, reduce time-lags between design proposals and associated engineering analysis, and relieve some of the project management pressure by providing engineering insights on early decision making. Knowing the current design challenges faced in industry gained from work experience, a research proposal was submitted in early 2020 to study the viability of machine learning for structural *design* applications, and led to the research investigation that is presented in this publication-based thesis.

2.6 Literature review: machine learning for civil and structural engineering

A short literature review on machine learning for civil and structural engineering applications is presented next given the substantial amount of literature covered in Chapter 3, along with tailored literature reviews for each of the individual journal articles presented in Chapters 4, 5 and 6. This review serves mainly to identify the largest knowledge gaps that motivated the very same journal articles presented later in this thesis.

2.6.1 Periods of distinctive machine learning research

A short literature review on machine learning for civil and structural engineering applications is presented next given the substantial amount of literature covered in Chapter 3, along with tailored literature reviews for each of the individual journal articles presented in Chapters 4, 5 and 6. This review serves mainly to identify the largest knowledge gaps that motivated the very same journal articles presented later in this thesis. The use of machine learned (ML) and data-driven methodologies for civil and structural engineering is pervasive, evidenced by the existence of recent, quasi-thematic review articles covering the application of ML from construction [64] to bridge health monitoring [65], crack detection [66], risk and resiliency assessments [67], earthquake engineering [68], and concrete properties predictions [69]. However, the use of ML for structural design has received far less attention, with only a few review articles that partially cover the corresponding literature [62, 70–72], with the notable exception of one recent "opinionated review" [73], which reviewed only a select few applications of artificial intelligence on structural design, and addresses the potential implications of machine learning replacing engineering intuition and creativity in the future.

The majority of review papers point towards distinctive periods of intensive ML research in civil and structural engineering. The first period was from 1988 to 2000, which was kickstarted by the theoretical development of artificial neural networks (ANNs) in 1982 [74] and by the back-propagation technique developed in 1986, both of which continue to underpin data-driven research [75]. Furthermore, there was a series of other

ML algorithms developed based on regression (LASSO [76], ridge [77], polynomial basis functions [78]), tree-based algorithms (decision trees [79], random forests [80], adaptively boosted tress [79, 81]), support-vector machines [82], and K-Nearest Neighbors [83] as partially reviewed by Sun et al. [62]. Unfortunately, the practical implementation of these methods was severely limited due to the significant lack of computational resources and readily accessible data [84] during this period.

The identification of these constraints significantly reduced investigations of machine learning applications within civil engineering between 2000 to 2010, and is generally reflected by a notable lack of papers and articles in the early 2000s. However, theoretical research in machine learning models continued, specifically for deep learning methods such as recurrent neural networks (CNNs) in the late 1990s [85], convolutional neural networks (CNNs) in the mid-2000s [86] and graph convolutional neural networks in 2009-10 [87]. These new machine learning models, along with an exponential increase in computational power to perform learning, revitalised ML research in civil and structural engineering during the 2010s [88].

Over the last decade, several new developments, most notably attention-based mechanisms called transformers in 2017 [89] and, more recently, domain-embedding networks such as physics informed neural networks (PINNs) [5, 90, 91] have attracted significant research efforts. It is worth noting that it is the transformer architecture that underpinned the success of large language models such as GPT, BERT and LAMA [92] that dominates deep learning research today. The machine learning literature for structural engineering applications tends to investigate the application of the latest theoretical machine learning techniques with an approximative two to three year delay. PINNs that were established in 2019 began to be investigated in earnest for structural engineering applications from 2021 onwards [93, 94], and the success of ChatGPT in 2022 has, for example, begun to spur research grants for steel design applications as of late 2023 [95].

2.6.2 Application areas of ML in civil and structural engineering

Sun et al. [62] have identified four application areas of ML in civil and structural engineering: surrogate modelling for predictive structural response, interpretative modelling of experimental data, information retrieval models based on images and written texts, and structural health monitoring models [88]. Although some examples of unsupervised or reinforcement learning exists in literature [96], the majority of research is dedicated to the creation of supervised learning regimes. One noticeable gap within the research which culminated between 2010 to today is significantly fewer investigations for structural *design* models.

As a matter of fact, examples of structural design specific ML driven models only tend to have occurred in the 1990s, with neural networks having been the predominant machine learning modality. Vanluchene and Sun for example developed a simple fixed frame concrete beam design model which took as inputs bending moment, reinforcing steel strength, concrete compressive strength, reinforcement ratio and an aspect ratio to predict the required structural depth of the beam [97]. Whilst their use of multi-layer perceptron models with two hidden layers was state-of-the-art in 1990, their research was severely limited in the amount of synthetic data which could be produced. Furthermore,

their approach was only suitable for statically determinate systems in which the bending moment was not a function of the required structural section. Another paper investigated the application of neural networks to aid truss design problems with 10, 15, 20 or 25 structural elements [98]. In this instance, the network was tasked with identifying the required cross-section areas for economical design based on a variety of loading conditions. Whilst the results were promising for the particular truss chosen, this particular methodology suffered from the inherent weakness of not being generalisable to truss designs of a different topology, which has continued to be a common issue in literature to this day [99–101]. This presents a unique research opportunity to develop new methodologies that enable existing machine learning modalities to generalise for different boundary conditions, systems sizes and design constraints.

As of 2020, there had been some attempts to implement ML for structural design applications directly [102–104], indicative that interest in this field was rising. Yet ML design models are significantly less represented within the literature than surrogate analysis models. This could be interpreted as evidence that such design models are more complex operators, and concurs with the inverse problem perspective for structural design presented in Chapter 3. ML driven surrogate models have been useful in streamlining computationally intensive simulations, which may indirectly help in designing structures, most notably for non-linear, dynamic analysis. The pre-dominant focus in the literature appears to be for seismic systems in steel [105, 106], concrete moment frames [107–109], steel braced frames [110, 111] and reinforced concrete shear walls [112]. One distinct difference between these papers with those from the 1990s is their more diverse use of various ML algorithms, such as linear regression models, support vector machines, and K-Nearest Neighbors in addition to feed-forward neural networks. However, these surrogate models simply accelerate the evaluation of structural response parameters (in other words forward operators) which could normally be evaluated less quickly by more deterministic techniques.

Interpretative ML models developed using data from experimental models have a distinctive difference to the surrogate models that were predominantly trained on synthetic data. The reliance on experimental data typically resulted in extreme small samples sizes (<100), and such models were used to evaluate various response variables most often related to structural capacity such as shear strength [113], drift capacity [114], reinforcement-concrete ultimate bond strength [115], and punching shear capacity [116]. Since the goal of these models is to estimate some form of structural properties they fall predominantly into the regression modelling category. However, recently there have also been examples of classification models which predict the probable failure mode of RC structures based on in-situ experimental data such as measured material properties and stiffness parameters of the system [117, 118].

The last two major application areas are related to information retrieval models which predominantly take advantage of pre-trained CNNs to extract information from images [86] using a technique known as transfer learning. Examples include evaluating RC cracks and spalling patterns [119], identifying loosened or corroded steel bolts [120] and more general structural component damage classification [121]. There was also one unique application of LSTMs models [122] to assign building damage classification tags based on natural language descriptors using a dataset from buildings affected by the 2014 South

Napa earthquake [123]. ML models have also been developed for structural health monitoring (SHM) purposes, which is further covered in Chapter 3. Commmon model archetypes include auto-regressive model fitting [124], Fast Fourier Transform [125] and wavelet transformation [126]. ML-specific algorithms have also been used on aluminium [127], steel [128] and RC frames/elements [112, 129] using neural networks, support vector machine and kernel ridge regression methodologies.

2.6.3 Criticism, challenges and areas for future research

The literature identified above indicates that machine learning research has grown significantly in civil and structural engineering. Not only are various ML techniques utilised, yet the ML model development workflow from data collection, preparation, training, analysis and deployment is fairly well understood. This is evidenced by comparing the content of two literature reviews across three decades; whilst in 1997 the literature primarily focused on describing the general problem solving workflow for machine learned models [84], in 2020, the authors instead provided far more detailed explanations such as an in-depth overview of the mathematics of each ML technique and other supplementary details such as the importance of feature engineering (covering techniques such as chi-square tests, correlation coefficients, Principle Component Analysis) [62]. Furthermore, there is a greater appreciation for the performance metrics used for both regression and classification models such as accuracy, precision and recall for binary classification, confusion matrices for multi-class classification, and mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) for regression models [130].

However, there are still significant challenges and hurdles before civil engineering applied ML research reaches further maturity. First, there are no accepted datasets on which civil and structural engineers train on, meaning that it can be difficult to properly compare the results between any two studies. Aside from making reproducibility and verifiability challenging, it also hinders comparative and future research since significant effort needs to be carried out in creating the datasets in the first place. This is in stark contrast to other machine learning investigations in which commonly used datasets such as MNIST [131] and ImageNet [132] provide a common benchmark dataset against which new and different types of machine learning models can be tested against. Recently there has been progress to create such data repositories, including DesignSafe [133] and the Datacenter-Hub [134] to specifically aid the distribution of engineering and science based machine learning projects. Any future research should aim to both openly share as well as contribute to the creation of high-quality datasets.

Second, the rationale for the use of a particular machine learning model over another one is often not provided in literature. Some researchers choose to use a single type of ML algorithm [105, 107, 109, 114, 120] whilst others conduct a comparative study of various learning types [112, 113, 118, 123]. This may on one hand be indicative that the applicability of certain ML models is still not properly understood, which might only be discernable a posteriori once more investigations have been conducted. Furthermore, it is likely that certain machine learning problems and model types will be better suited for certain types of datasets, which underlines once again the importance of establishing

benchmark datasets to investigate if a particular ML technique's performance is data-set or problem specific. It should also be stressed that a majority of investigations do not begin with a simple linear regression model and often instead directly jump into advanced ML techniques; however there is evidence that for at least certain problems, a linear regression model can often be entirely sufficient [135, 136]. Creating a simple base-line model should therefore be standard practice in applied machine learning research.

Lastly, a common issue with machine learned models is their inherent opaqueness, a problem that was already identified early on by Reich in 1997 [84]. In a traditional data-driven model, it can be difficult to explain the model's performance or to interpret the reliability of its results [88]. Within the civil engineering literature, there have been only a few attempts to tackle this problem [112, 137]. However, there are also a large set of untested methods, such as partial dependence plots and individual expectations curves [138, 139] which aim to unravel the inner workings of ML models, or at least improve our understanding of their performance. The most interesting and recent development to improve the interpretability and explain-ability of ML models is by embedding domain-specific knowledge to train the parameter space, which is generally identified as theory-driven data-science [90]. One example of such systems are the aforementioned physics-informed neural networks, which use auto-differentiation during the learning process to incorporate physical constraints in the loss function, that help the network to converge towards physically realistic predictions during training [90]. This kind of embedded knowledge also has the additional benefit of avoiding over fitted models [5]. PINNs however are a relatively novel contraption, and hence their application within a civil and structural engineering context, especially for structural design, is still relatively untested, providing fertile ground for further investigation.

In summary, the literature review points towards a clear knowledge gap in the application of machine learned structural *design* models in civil engineering. The lack of accepted benchmark datasets that allow a more honest comparison of modelling performance also suggests that future machine learning research should contribute to the generation of a coherent database which allows other researchers to contribute to this field. PINNs, a novel neural network model, fall into the category theory-driven data science approaches which attempt to tackle the "black-box" problem of data-only methodologies. Novel techniques to address the issue of generalisability through appropriate selection of the feature space also represents a unique opportunity for novel research findings. The research articles presented in the subsequent chapters were predominantly motivated by the gaps of knowledge identified in this literature review, and successful methodologies and techniques for each knowledge gap is considered crucial to help create machine learned structural design models capable of addressing the modern design challenges identified in Section 2.4.

Part II

Journal articles

Chapter 3

Article: "Structural engineering from an inverse problems perspective"

Foreword commentary

This article titled "Structural engineering from an inverse problems perspective" was published in *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* on January 26th, 2022 [140]. The goal of the paper was to provide an overview of the various fields of structural engineering in which inverse problems already exist including structural health monitoring, blast engineering and digital twins, but also to reconcile the view that structural design can itself be viewed as an inverse problem, with the hope of fostering cross-discipline collaboration between these various fields. The work was motivated based on the findings from philosophical engineering literature partially covered in Chapter 2, and sets the overarching framework of the structural design models developed in Chapters 5 and 6. The article was conceptualised by Adrien Gallet and Danny Smyl and all named authors contributed to writing this article, which included Adrien Gallet, Sam Rigby, Tyler Tallman, Xiangxiong Kong, Iman Hajirasouliha, Andrew Liew, Dong Liu, Liang Chen, Andreas Hauptmann and Danny Smyl.

Abstract

The field of structural engineering is vast, spanning areas from the design of new infrastructure to the assessment of existing infrastructure. From the onset, traditional entry-level university courses teach students to analyse structural response given data including external forces, geometry, member sizes, restraint, etc. – characterising a *forward* problem (structural causalities → structural response). Shortly thereafter, junior engineers are introduced to structural design where they aim to, for example, select an appropriate structural form for members based on design criteria, which is the *inverse* of what they previously learned. Similar inverse realisations also hold true in structural health monitoring and a number of structural engineering sub-fields (response → struc-

tural causalities). In this light, we aim to demonstrate that many structural engineering sub-fields may be fundamentally or partially viewed as *inverse problems* and thus benefit via the rich and established methodologies from the inverse problems community. To this end, we conclude that the future of inverse problems in structural engineering is inexorably linked to engineering education and machine learning developments.

3.1 Introduction

The estimation of structural response from loading and boundary conditions is a fundamental concept in structural analysis, from elementary Euler-Bernoulli beam theory to non-linear simulations involving complex structures subjected to extreme earthquake excitation. In fact, numerical computation of structural response from known causalities characterises a *forward* problem (cause $\rightarrow$ effect) and has rightly been the source of significant research since the advent of modern computing. Amongst the myriad of computational frameworks, the finite element method (FEM) [141–144], finite difference [145, 146], spectral element [147, 148], and hybridisations [149] have proven both widely applicable and successful over the years. The implementation of such forward models have aided engineers in their ability to model, analyse, and design structures with arbitrary geometry and precision, contributing greatly to the presence of skyscrapers, supersonic aircraft, large cruise ships, and many more engineering examples. In the near future, the pervasiveness of, for example, the FEM appears inevitable while its usefulness is unquestionable in structural engineering applications.

Pragmatically however, the final configuration of structural members is not known at the beginning of the design process; i.e. one iteration of a structural simulation is not generally sufficient in a real project. This reality implies that the design of structures is an iterative process – for example the identification of appropriately sized structural members, connections, and restraints (causalities) from design constraints, building codes, and environmental considerations (data). As is often the case, the iterative design process is carried out initially using design tables, rules of thumb, handcrafted protocols, optimisation regimes, etc. Nonetheless, this process is emphatically an *inverse problem*, where an engineer is given data alongside design objectives and challenged to determine the appropriate structural configuration (causalities).

Of course, the field of structural engineering is diverse, in which structural design is one of many sub-fields where inverse problems are applicable. Perhaps a more straightforward implementation of inverse problems is structural health monitoring (SHM), where real-time (or near real-time) data is used in the prognosis of structural condition. Indeed, the detection, localisation, prediction, and prognosis of potential damage processes is enabled in one of two ways, (a) via pattern recognition or (b) solving an inverse problem (or series of inverse problems) [150]. Moreover, certain non-destructive evaluation (NDE) approaches are also known to employ inverse problems (e.g. X-ray CT and emerging NDE approaches in academia) to asses the damage state of structural elements measured offline [151]. For contextualisation, a schematic example contrasting the differences between forward and inverse models is provided in Fig. 3.1; this well known problem is referred to as an *inverse elasticity problem*.

These, and many more sub-fields of structural engineering and structural mechanics

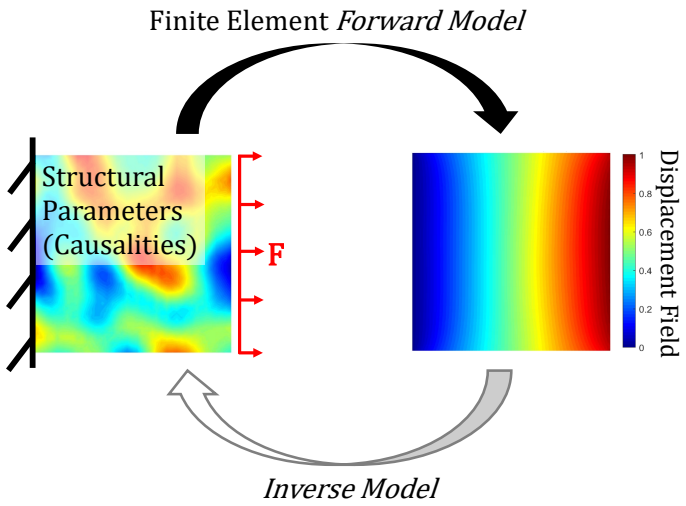


Figure 3.1: Schematic illustration depicting the forward and inverse problem relationship for a stretched elastic plate with randomised stiffness properties. The forward finite element model inputs (causalities) are shown as the non-homogeneous stiffness properties while the model output is the displacement field. In contrast, the inverse model aims to estimate the stiffness properties *given* the displacement field.

[152], can not only be (fundamentally or partially) viewed as inverse problems, but, as we aim to illustrate herein, are benefited by the systematic approaches comprising the rich area of inverse problems. Too often overlooked by structural engineers and structural researchers, the mathematics of inverse problems is an established field, ranging from classical statistical/Bayesian methodology [153] to cutting-edge implementation of deep neural networks [154]. Moreover, while the present use of inverse problem methodologies in structural engineering is limited, its potential is immense across the expanse of the structural engineering sub-fields.

In this paper, this potential will be discussed in detail and contextualised among a broad suite of existing inversion-based applications. To begin, a clear description of inverse problems and methods will be detailed. Following this, a review and discussion of inverse methodologies in modern structural engineering applications will be provided. Inasmuch, the intent of this manuscript is to examine the following topics in structural engineering through the lens of inverse problems. We remark, however, that the forthcoming topical sections are *not* intended to be exhaustive reviews, but rather, to provide substantiating evidence for the pervasiveness of inverse problems in structural engineering. Lastly, realisations, overview, paths forward, and conclusions will be presented.

3.2 Inverse problems, methods, and contemporary use

Traditionally, the field of inverse problems is concerned with the mathematical question of if and how one can determine the cause for certain measurements. Despite being primarily mathematically oriented, the underlying questions always stem from relevant

physics and engineering applications. This is especially true for one of the most prototypical inverse problems, the so-called Calderón problem [155] that asks: can one determine the conductivity of a body from electrical measurements at the boundary. In fact, this question arose during Calderón's time as a civil engineer, before he pursued an academic career in mathematics. In the following, we want to close the loop back to civil and structural engineering application that once motivated an entire field of mathematical studies, by utilising the insights gained in the last decades.

More generally speaking, inverse problems consist of finding the unknown characteristics of a structural system from some of the outputs, or measurements of that system. Most notably, this includes the above mentioned inverse conductivity problem in geophysics [156] and engineering, but also includes a large field with applications in medical image reconstruction [157, 158]. Mathematically, such problems are ill-posed, broadly meaning that the parameters to be estimated θ are highly sensitive to changes in the measurement data d. The solution to the inverse problem involves estimating the parameter θ from a fixed set of measured data d, in contrast to the forward problem of computing d from knowledge of the system parameter θ. Specifically, this means given the forward model U, that models the system equations, we first formulate the underlying observation model

$$d = U(\theta) + \delta d. \tag{3.1}$$

Here, δd denotes an error term, modelling several sources of possible errors, such as inaccurate measurements or even inaccuracies in the model simulation. The question that remains is: "how can we obtain θ from d given the above relationship?" Which we will call the reconstruction problem in the following. It is important to note that we can not simply invert the forward problem (3.1), as the ill-posed nature implies that there can be either no or multiple solutions and additionally under inevitable measurement noise these solutions are not stable to compute by direct inversion. This ill-posedness of the inversion procedure constitutes the underlying paradigm of an *inverse problem*.

In order to obtain stable reconstructions, we make use of a concept known as regularisation [159], which aims to assign a unique solution to each set of measurements in a stable manner, that means if the noise in the measurement vanishes, we would obtain the original system parameter. We can separate such stable reconstruction procedures into two primary classes: ones that compute a solution θ^* directly from measured data and those that iteratively aim to fit a solution by minimising a suitable cost functional. In the first case we aim to formulate an inverse mapping $U^\dagger$, such that

$$U^\dagger(d) \approx \theta. \tag{3.2}$$

The primary problem in obtaining such direct inversion algorithms is that they can be highly dependent on the problem under consideration. Especially so, when the relationship between d and θ is non-linear. Thus, obtaining such a mapping is a highly non-trivial task, but reveals much about the underlying problem characteristics and hence is a primary interest of mathematical research [157, 160, 161].

The second case, is a more principled approach that can be formulated for a large class of problems. The underlying premise is to reformulate the reconstruction problem as an optimisation problem. That is, we formulate a cost functional that measures how good our

reconstruction fits the data while simultaneously enforcing some additional characteristics and acting as regularisation for the reconstruction process. Specifically, the reconstruction problem then writes as finding a minimiser of

$$\theta^* = \arg\min_{\theta} \frac{1}{2}\|U(\theta) - d\|_2^2 + \alpha R(\theta). \tag{3.3}$$

Here, the first term enforces that reconstructions fit the data, whereas the second term is the so-called regularisation term. As discussed previously, this regularisation term is necessary when dealing with inverse problems, as it prevents a solution from overfitting the measurement noise. Importantly, by incorporating prior knowledge in the design of R [162, 163], we effectively choose preferred solutions and overcome the problem of non-uniqueness. Finally, the parameter $\alpha > 0$ balances both terms and depends on the noise amplitude. Solutions to (3.3) are computed by suitable optimisation schemes, for which repeated evaluation of the forward model U will be necessary. Consequently, computing solutions to (3.3) can be computationally highly expensive, if the evaluation of U is expensive. Thus, for non-linear problems fast converging algorithms, such as Gauss-Newton or related methods [164], are preferred.

Lastly, with the recent rising popularity of data-driven methods, researchers have designed computationally more efficient ways to address the reconstruction problem [165]. Such data-driven approaches are often inspired by classical reconstruction algorithms discussed above. For instance, one can replace the direct reconstruction operator or parts of it, with a data-driven component, typically consisting of a neural network [166–168]. Given a set of informative reference data, one can then learn a suitable mapping mimicking (3.2). Alternatively, researchers have investigated the possibility to improve the iterative process to compute solutions to (3.3), by replacing parts in the optimisation algorithm with learned components [154, 169–172], or entirely building network architectures motivated by the iterative solution process [173, 174]. The use of data-driven methods is expanded upon in Section 3.7.

3.3 Structural design as an inverse problem

3.3.1 Demarcating structural analysis and design

Prior to World War II, engineering higher education was originally focused on the art and practice of engineering design [175]. By the 1960s however, due to the success of science-based ventures such as the Manhattan Project and the rise of government-sponsored research grants that severed the link between academia and industry, engineering science became the main field of research and teaching at universities [176]. This research led to the development of powerful structural analysis techniques, yet also left engineering graduates with a noticeable loss of practical engineering skills [17]. In the 1990s and early 2000s, there was a push to introduce capstone design projects in university engineering degrees to address this shortcoming, the success of which is disputed [177].

There is a broad agreement within the literature that analysis and design are two distinct activities. Structural analysis, which falls into the field of engineering science, is

primarily concerned with establishing *knowledge-that* explains the world, whereas design is concerned with *knowledge-how* something works [4]. Design is often characterised as being ill-structured [178], open-ended [179] or even "wicked" [180], qualities which do not necessarily lend it to science-based research and helps motivate the search for a different paradigm which more adequately addresses its true nature.

One perspective to account for these differences is to recognise structural analysis and structural design as being two different types of problems. Although structural analysis is typically seen as a sub-field of design to validate and justify the adequacy of structural elements [14], we like to advocate the view that structural design is in fact an *inverse-problem*, with structural analysis forming the related *forward-problem*.

Unlike typical inverse problems such as the one shown in Fig. 3.1, where physically measured data (the displacement field) is used to identify the causalities (stiffness properties), in structural design we are dealing with a theoretical construct, or "theoretically" measured data. This data contains the set of complex design constraints which need to be adhered to, such as ultimate (ULS), serviceability limit states (SLS), sustainability and constructibility. From this perspective, structural design can be seen as the process of arriving from a specific set of constraints to a viable structural solution, with analysis being the process of checking if the proposed structural solution adheres to those constraints as shown in Fig. 3.2.

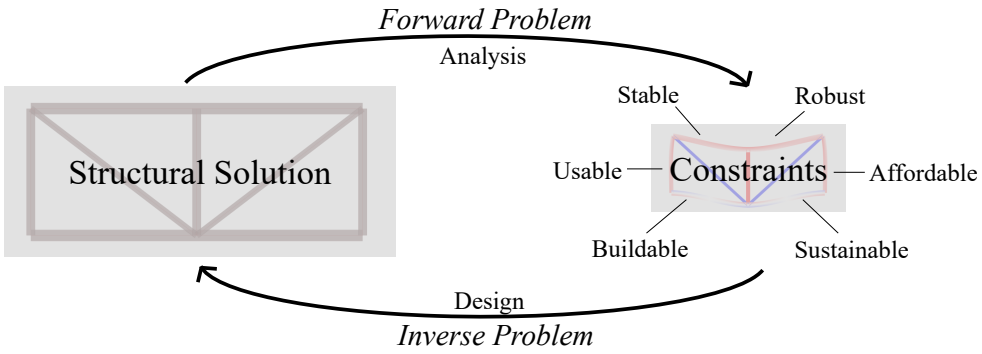


Figure 3.2: Relationship between structural analysis and design, in which design is the inverse problem of evaluating a suitable structural solution *given* a set of constraints. Some constraints (such as stability constraints) render themselves more easily to quantitative treatments.

This perspective is evidenced by realising that the key features of inverse problems, specifically their ill-posed nature as explained in Section 3.2, manifest themselves in design. Their shared characteristics include being unstable, non-unique and sometimes un-solvable problems [181]. Notice how these qualities were alluded to as "ill-structured", "open-ended" and "wicked" by previous researchers, yet to the best of our knowledge, this is the first time that literature has attempted to view structural design as an inverse problem.

We suggest that the application of the inverse problem perspective in design gives rise to the idea of a "design model". Similar to how an "analysis model" allows us to evaluate the action effects of a given structural solution, solving the forward problem shown in Fig.

3.2, a "design model" would generate structural solutions which adhere to the given set of user defined constraints, solving the inverse problem. This perspective sheds light on the characteristics which design problems directly share with other inverse problems, such as the non-uniqueness property. Multiple viable solutions often exist to any given design brief, one example of which are the various truss-archetypes for bridge designs to span a similar distance. In order to create a viable design model, it would be necessary to provide some form of regularisation, examples of which are provided in Section 3.5 in the context of structural health monitoring, which effectively encourage design models to choose preferred solutions based on prior knowledge. This could be achieved by constraining the solution space to a sub-set of structural systems, cross-sections or materials based on the specific constraints provided. Other properties of inverse problems which arise in design are shown in Table 3.1 and a comprehensive understanding of the application of the inverse problem perspective warrants further research using specific design examples.

Ill-posed inverse problem characteristics	**General description**	**Examples in structural design**
Unstable[1]	Small changes in constraints can lead to large changes in the solution space	Impact of grid-spacing on the selection of appropriate floor options
Non-unique	For a given set of constraints, multiple solutions exist	Various truss archetypes which can span the same distance
Unsolvable	Lack of realistic constraining data impedes the search for an adequate solution	Current lack of appropriate structural materials for space elevator designs

Table 3.1: Overview of ill-posed inverse problem features in the context of structural design.

We note that the design problem can be also formulated as an optimal control problem [182], where the optimal design parameters are thought to be found as minimiser of a penalty function while satisfying the system equations. Whereas, the optimal control approach can provide an effective way to solve complicated design problems, it falls short in accounting for uncertainties or inaccuracies in the forward model, and especially the link between measurement and system parameters. We believe that the here lies the strength of the inverse problems viewpoint, that offers a rich interpretation and link between system parameters and measurements given as the forward model. This is not only promising for the optimisation task in the design process, but also for new ways to approach the modelling of the forward problem.

3.3.2 The link to structural optimisation

As explained in Section 3.2, inverse problems can effectively be solved iteratively [13]. Effectively, this process involves making an estimate of the structural solution based on experience and design heuristics (such as simple rules of thumbs), checking those estimates through an analysis (forward) model and updating the model if necessary; in other words, this approach is characterised by creating an optimisation problem. Some examples of

[1] In reference to the solution space, not structural instability (buckling), see [181].

forward-driven optimisation models used in structural design include: optimising the deformed shape of flexible formwork structures to predefined target geometries [183, 184], best-fit geometry optimisation of thrust networks in the design of shell structures [185], and finding the optimal structural forms for long-span bridges [186], gridshells [187], trusses [188], portal frames [189] and structural sections [190].

A key theme in these research works is that the structural geometry or member proportions are not initially defined, but rather are form-found or discovered in the process, based on the defined loading, boundary conditions and objectives. Often these discovered structural forms may lead to step change benefits in terms of performance or reduced material usage, as they are unbiased by our preconceived perception of what a "good" structure is, or by what we currently design and build in the construction industry with standard template solutions. It is also true for many cases, that a solution may not even exist, forcing us to accept a closest best fit solution, or it can be that an optimal solution may have multiple candidates by virtue of the structure's static indeterminacy. A common problem with a forward (sometimes brute-force) optimisation approach, can be lengthy computational times for structural analyses in the objective functions, and the sheer size of the design (and hence optimisation) search space, stemming from many wide ranging input design variables. While fast and globally convergent convex optimisation programs can be set-up, many practical structural engineering design problems are inherently non-linear in nature, forcing a slower approach that uses local searches or heuristic methods with no guarantee of a global optimum. This is a current challenge faced in solving inverse problems iteratively with forward models.

3.3.3 Implications of treating structural designs as inverse problems

This inverse problem perspective has various implications. Firstly, it should emphasise that design problems ask a different question than those related to analysis. An analysis model solves the forward-problem, and answers questions such as "what is the ULS utilisation ratio of a particular beam system for a specific load, with the following specific cross-sections and support conditions?". An appropriate design model would ask the reverse: "what is the group of cross-sections and support-conditions which ensures a utilisation ratio of less than 1.0 for this particular beam system to carry this particular load?". In this formulation, the magnitude of loading and the utilisation ratio serve as the design constraints (both are known "data"). Whilst engineering science has produced sophisticated analysis models, the research of such "design models" is lacking in academia.

Secondly, the inverse problem perspective sheds insight on the possibility of using data-driven approaches as opposed to more typical optimisation techniques. The rise of machine learning and deep neural networks could be used for the development of such "design models", which focus on directly identifying a set of structural solutions from a given set of constraints using learned data. Such models could address some of the challenges faced in design due to tight deadlines that force early design decisions, whose full implications might only be realised at the detailed design stage where changes become cost and time prohibitive. This can lead to structural solutions that are difficult to build, have poor sustainability metrics, and be costly to engineer and fabricate. If one instead considers the design process in an inverse manner, rooting firmly first at the end goal, it

could be possible to reduce project risk and pick more effective structures by appreciating many solutions to the brief from the onset. This was already alluded to in Section 3.2 and is expanded upon in section Section 3.7.

Lastly, the ultimate benefit of an inverse problem perspective is that it helps to clearly distinguish between analysis and design procedures and provides academia with an adequate framework to contextualise *design model* research. Engineering academia has been dominated by the engineering science perspective [191], predominantly choosing to research and teach forward problems. One of the uncomfortable implications of this view is that engineers from over 150 years ago, which trained primarily in the art and practice of engineering design, may in fact have been better "inverse-problem solvers" than academics and graduating engineers of today (who are stronger in solving the well-structured forward problems) [8]. This might help swing the pendulum away from focusing exclusively on forward models (analysis) towards a more stable equilibrium with inverse models (design) by acknowledging the existence of these two related, albeit distinctively different types of problems. The use of inverse problem and inverse analysis in a related field to structural design, notably blast engineering, is discussed next.

3.4 Extreme loads on structures

3.4.1 Blast loading and inverse analysis

High-rate dynamic loads can arise from events such as earthquakes, wind, tidal waves, impacts, and accidental or malicious explosions. Here, the imparted load may be comparable or several times larger than the strength of the material it is acting on, it can be applied and removed in sub-second durations, and is often highly localised. Accordingly, the notion of static structural design according to a pre-determined distribution of stresses and strains may not be appropriate, and instead the designer must consider energy balance, non-linear analyses, and deformation modes for which there is no equivalent static counterpart.

Blast loading is undoubtedly one of the most aggressive forms of dynamic loading. When an explosive detonates it undergoes a violent and self-perpetuating exothermic chemical reaction, releasing energy through the breaking of inter-molecular bonds during oxidation [192]. The explosive material is converted into a high pressure (10–30 GPa), high temperature (3000–4000°C) gas which violently expands, displacing the surrounding air at supersonic velocities (6–8 km/s). This displacement causes a shock wave to form in the air, termed a blast wave, which eventually detaches from the expanding detonation product 'fireball' and continues to propagate into the surrounding air, decreasing in pressure and density as it expands.

When a blast wave encounters an obstacle some distance from the source it will impart momentum as the air is momentarily (either fully or partially, depending on the compliance of the obstacle) brought to rest at the air/obstacle interface. Prediction of blast even in the most simple settings is a considerable challenge to the scientific community. This becomes an increasingly complex and multi-faceted problem when considering issues such as: obstacle orientation; proximity of the obstacle to the source and additional momentum transfer from fireball impingement; secondary combustion effects either at the air/obstacle

interface or in late-time due to partial or full confinement of the explosive products; the presence of mitigating or blast-enhancing materials (soil, reactive munitions, etc.).

Real-world blast events are highly uncertain, and the need for inverse analysis is clear: it is very rare that the exact size, shape, composition, and location of an explosive device is known *a priori*. Instead, information relating to the *cause* of an explosion should be estimated, within reason, from the more readily observed *effects*, i.e. the magnitude and severity of structural damage to surrounding buildings, and cratering of the ground surface. Whilst inverse analysis is well-established for practical post-event assessment of explosions—and has been used to determine the size/location of blast events through forensic investigation of social media videos [193] or numerical modelling correlating structural damage [194, 195]—the use of inverse modelling in an academic context is yet to be exploited fully. In the former, order-of-magnitude estimates are typically deemed sufficient, whereas the latter requires repeatable, precise measurements and high levels of experimental control.

The lack of robust yet high-fidelity experimental techniques has stifled academic research into close-in blast for some time. Close-in blast is typically defined as the region within approximately 20 radii from the charge centre, where loading in this region is characterised by a near-instantaneous rise to peak pressure in the order of 100–1000 MPa, followed by a rapid decay to ambient conditions typically occurring in sub-ms. Subsequent structural response may reach a peak value in the order of 10-100 mm and, whilst this may occur orders of magnitude slower than load application, deformation cycles are still generally within ms durations.

Recently, two notable advancements have been made in experimental characterisation of close-in blast and structural response. In the first of these, researchers at the University of Sheffield (UoS), UK, developed a large scale apparatus for the spatial and temporal measurement of blast pressures from close-in explosions [196, 197]. In the second, researchers at the University of Cape Town (UCT), South Africa, adapted the well-known digital image correlation (DIC) technique to measure the transient response of the rear-face of blast loaded plates [198]. These two techniques were combined in a recent study [199, 200] where, for the first time, detailed loading maps *and* temporal structural response profiles were developed independently, in a single-blind study, for identical (scaled) experimental set ups.

3.4.2 Proof-of-concept experimental studies

Of the experiments performed in [199], 12 are relevant to the notion of inverse analysis of blast loading and structural response, and will be discussed here. Six tests were performed with spherical explosive charges; three at UoS measuring blast loading and three at UCT measuring structural response. In the UoS spherical tests, 100 g PE4 charges were located at 55.4 mm clear distance from the centre of a nominally rigid target, on which the reflected blast pressures were measured. In the UCT spherical charge tests, 50 g PE4 charges were located at 44.0 mm clear stand-off distance from the centre of the flexible target plates: 300 mm diameter, 3 mm thick, Domex 355MC steel plates, fully clamped around the periphery. The plate response was filmed using a pair of stereo high speed video cameras and DIC was used to determine transient plate response. The two test series can

be expressed at the same scale using well-known geometric/cube-root scaling laws. Here, it is assumed that the flexible targets deform on timescales orders of magnitude longer than loading application, and therefore differences between the loads imparted to the rigid and flexible plates are negligible.

Specific impulse is given as the integral of pressure with respect to time. Numerical integration of the UoS pressure histories (at various distances from the centre of the plate) yields *directly-measured* specific impulse distributions. The first few frames of the UCT tests were used to determine initial velocity distributions of the plate, from which imparted specific impulse could be *inferred* through localised conservation of momentum: $i(x) = v(x)\rho t$, where i is specific impulse, x is distance from the plate centre, v is out-of-plane velocity of the plate, ρ is density (7830 kg/m^3 for Domex 355MC), and t is plate thickness.

The results for the spherical tests are shown in Fig. 3.3. The full-field inferred specific impulse distributions are in close agreement with the discreet, directly measured values, and both measurements form a tight banding in an approximate Gaussian distribution [201]. Not only does this indicate a high level of test-to-test repeatability for each method, but demonstrates that the two methods are measuring the same underlying phenomena, albeit in entirely different ways. Thus, it can be said that an imparted impulse will result in an initial velocity uptake which is directly proportional, and therefore measurement of one allows for the other to be determined. This proves the concept of using plate deformation under blast loads in an inverse approach—namely that from knowledge of plate deformation one may be able to determine the imparted load—and provides physical verification of the inverse approach developed by [202, 203].

In addition to the spherical charge tests, six tests were performed in [199] using squat cylinders (height:diameter of 1:3). Such charges are known to produce a more concentrated load, with the fireball propagating at higher velocities along the axis of the charge [204]. This accelerates the growth and emergence of surface instabilities [205], which gives rise to a more variable loading distribution [206]. A key research question in this study was: "will a more variable loading result in more variable structural response?" In the UoS cylindrical tests, 78 g PE4 charges were located at 168.0 mm clear distance from the centre of the target, and in the UCT cylindrical tests, 50 g PE4 charges were located at 145.0 mm clear distance from the centre of the target. Again, specific impulse distributions were both directly-measured and inferred from plate response respectively, and the experiments were expressed at the same scale using common scaling laws.

The results for the cylindrical tests are shown in Fig. 3.4. Whilst the two methods again show good agreement, the results can be seen to form a much wider spread. In contrast to the spherical tests, where peak specific impulse was seen to consistently act in the plate centre for all tests, here the peak value is often up to 25 mm from the plate centre (approximately equal to the charge radius), in both the directly-measured and inferred values. The inferred values are generally bounded by the directly-measured values, which suggests that this spread is indeed a genuine feature caused by application of more variable load.

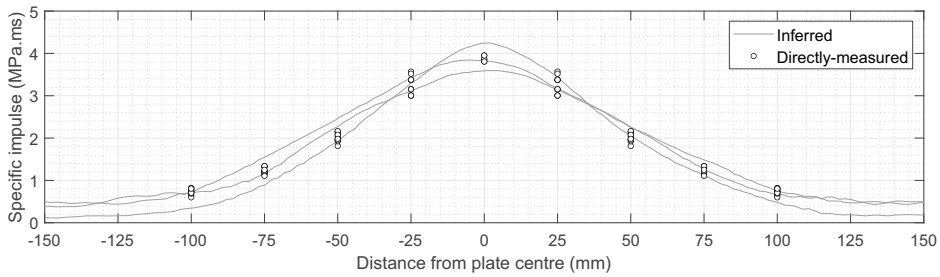


Figure 3.3: Directly-measured (UoS) and inferred (UCT) specific impulse distributions from studies of blast loading and plate deformation following detonation of spherical explosives, expressed at 100 g (UoS) scale, after [199].

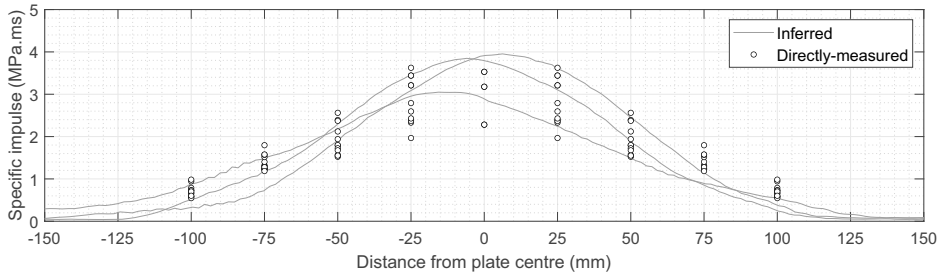


Figure 3.4: Directly-measured (UoS) and inferred (UCT) specific impulse distributions from studies of blast loading and plate deformation following detonation of cylindrical explosives, expressed at 78 g (UoS) scale, after [199].

3.4.3 Outlook

The aforementioned studies [199, 200] have provided a firm physical basis for inverse analysis in the context of extreme loading and structural response. The results have clear implications for the future of research in this area. Namely, it has been demonstrated that not only can inverse analysis provide excellent predictions of blast loading in repeatable, well-controlled situations (as with the spherical tests in [199]), but structural response measurements are potentially sensitive enough to detect localised variations in loading (as with the cylindrical tests in [199]). This is particularly important in situations where a highly variable loading might be expected (e.g. from explosives buried in well-graded soils [207]), but statistical variations cannot be determined in a robust sense when using direct measurements (note the discrete nature of the direct measurements in this study, compared to the effectively continuous nature of the inferred measurements). This technique may permit, through inverse methods, fundamental scientific studies of complex mechanisms governing blast loading following close-in detonation of explosive charges in situations where previous research has not yet been possible.

3.5 Structural health monitoring

3.5.1 Background on inverse methods in SHM

Unlike the fields of structural design or blast engineering, it is well known that inverse problems are deeply connected to SHM. In fact, computerised damage detection can generally be recognised as either a pattern recognition or inverse problem [150, 208, 209] where unknown or uncertain parameters (causalities) are estimated via quasi-static or dynamic structural response data. Among the numerous inversion-based approaches, FEM updating methodologies are among the most pervasive [210–213]. Much of the noted popularity is owed to the flexibility of the FEM for comparing modal parameters to an undamaged state and in compensating model errors.

Meanwhile, analytical [214], wavelet [215], fractal [216], fuzzy system [217], Kalman filter [218], chaotic interrogation [219], shape function [220], and particle filtering [221] approaches, among many others, have proven successful in uncertain inverse parameter identification applications within SHM. Broadly speaking, inverse SHM approaches can be grouped as either deterministic or probabilistic [222] – which is also generally the case in classical inverse problems [153]. In the latter case, estimation of uncertain SHM parameters takes the form of a probabilistic term, for example a value with an associated certainty, a probability itself, etc.

Irrespective of the computational approach used in damage detection, two key realisations affect the efficacy of inversion methodologies: (a) a baseline is generally needed to detect/quantify damage [223] and (b) the presence of damage inherently influences the linearity of structural behaviour [224]. In addressing (a), reference-free or baseline-free frameworks have been introduced [225–228] via the introduction of either assumptions on the reference state, implementation of prior physics knowledge, or probabilistic regimes. On the other hand, non-linearity in the structural response can either act as a corrupting entity when linear forward models are used (e.g. unacceptable forward model error) or used as an advantage when properly leveraged. Regarding the latter, as noted in [224], methods based on non-linear indicators, dynamical systems theories, and non-linear systems identification approaches can be used to aid or enrich the damage identification process; such a conclusion can also be extended to the pure usage of inverse approaches in damage detection.

In the past thirty years, implementation of inversion-based damage detection methods in SHM has steadily increased. This is due to both the increase in inverse problems know-how and computational resources. Yet, since the emergence of contemporary machine learning, the ability to solve problems deemed previously intractable has exponentially increased opportunities in this area. For example, in many cases, forward models may not be available or are too computationally expensive, sufficient non-linearity may exist to effectively model the desired physics, errors in highly reduced models may be excessive, the ability to compute model gradients may be overly expensive, etc. Moreover, the ability of classification networks to readily classify important variables such as the probability and/or severity of damage from structural data is intuitively appealing and pragmatically useful. In the following, we will provide contemporary examples highlighting the use of both classical and machine learning based SHM inverse approaches.

3.5.2 Static inverse problems in SHM

Incorporation of discrete static (or quasi-static) data measured from structures into SHM frameworks is well established. For example, a number of sources, including corrosion, relative humidity, fibre-optic, topography, laser, potentiometer, strain gauge, electrical, and thermal sources, have been successfully integrated into long-term condition monitoring protocols [229]. The richness of spatial-temporal data obtained from these sources lends itself well for use in inversion-based SHM, i.e. given a set or sets of static SHM data, use an inverse methodology in capturing (potential) damage. This is true in cases where numerical models are available for the problem physics and when they are not (e.g. learned models can act as surrogates when physics-based models are unavailable).

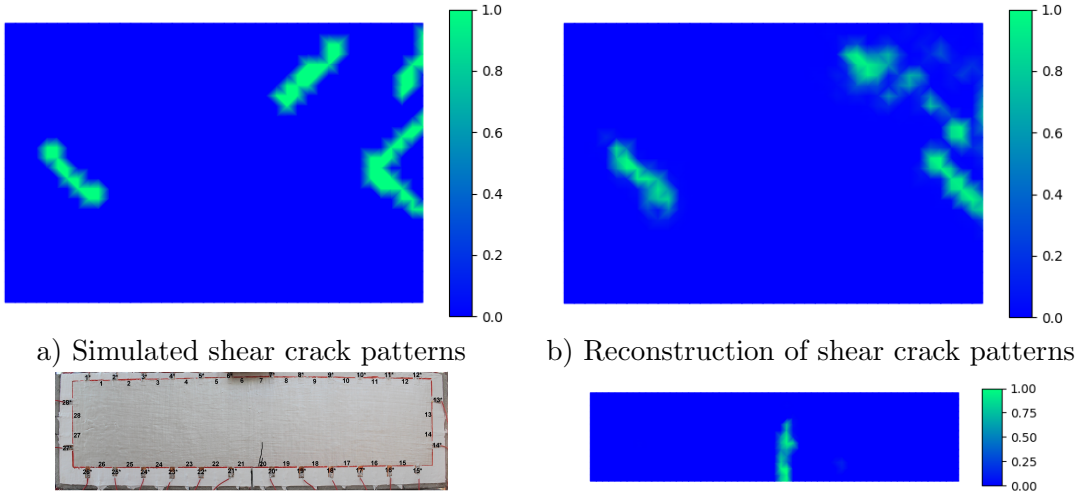
The sheer volume of literature available reporting the successful use of static inversion methods in SHM is formidable. However, roughly speaking, static inverse methodologies have been implemented within three areas, (a) point sensing, (b) area sensing, and (c) volumetric sensing. Holistically, it can be difficult to distinguish between each of these areas; for example, when lower dimensional measurements are extrapolated to characterise damage in area or volume targets [230, 231]. One such example is Digital Image Correlation (DIC), where the displacement field at discrete points on a structure is inversely computed via pixel movement and then extrapolated (interpolated) to a full-field, whereby the quality of the computed field is highly dependent on the quality of the contrasting area speckle pattern [232]. Similarly, one may consider the distinguishing of static measurement dimensionality as a local-global phenomenon where discrete local changes contribute to the analysis of the global structural system [230]. Lastly, to complicate matters even more, the use of discrete measurements can yield 2- or 3D information – as in the case of penetrative electrical measurements, where currents diffuse through the entirety of a body [233].

Fortunately these, perhaps philosophical realisations, are often washed out via the nature of inversion methodologies themselves. Pragmatically, at least in the context of SHM, the solution to static inverse problems generally requires a model, either physics based or learned. As such, the amalgamation or assimilation of data and solution dimensionality is often simply a matter of discretisation or model generation. In a similar vane, when static inverse problems are ill-posed, solutions generated using lower dimensional data are regularised/biased using prior models consistent with the solution dimension.

Many examples are available in the literature illustrating the efficacy of static inversion methodologies for applications in a suite of SHM implementations. For example, the use of displacement measurements for capturing SHM causalities in various structural geometries was reported in [234–236]. Of note, the specific applications using displacement fields to reconstruction elastic and elasto-plastic properties (and corresponding damage characteristics) has been the source of significant research [237–252]. In the pervasive case where displacement/strain measurements are discretely measured from strain gauges/fibre optics, inverse methodologies have also been fruitfully employed for damage characterisation, pressure and strain mapping, and shape sensing [253–264]. Perhaps one measure illustrating the success of such inverse approaches is highlighted by the recent interest in optimising the related sensing schemes [253, 265–267].

In the past decades, the emergence of electrical inverse methods has also proven a viable approach to static condition monitoring [268, 269]. This family of inversion-based

modalities generally utilises three different data sources including capacitative, direct current, and alternating current based measurements. Capacitative sensing is perhaps the newest of these approaches to sensing, where SHM causalities can be inversely computed using smart bricks [270], area sensors [271–274], and electrical capacitance tomography [275, 276]. On the other hand, owing to its established history in medical and geophysical applications, electrical impedance tomography (EIT, or electrical resistance tomography, ERT) is becoming a well established approach to damage detection, reconstruction, and localisation especially in concrete applications. For these, EIT presently manifests via two approaches, reconstructing conductivity maps using boundary voltages measured from area sensing skins [277, 278] or directly imaging a 3D cementitous body [279]. Representative 2D EIT reconstructions are provided in Fig. 3.5 using a machine learned approach (direct reconstruction) for imaging flexural and shear cracks in concrete elements. To this end, EIT has also been used for characterising area corrosion [278] and localising area temperature variations [280].



a) Simulated shear crack patterns b) Reconstruction of shear crack patterns

c) Experimental image of a flexural crack d) Reconstruction of a flexural crack

Figure 3.5: Reconstructions (right column) report probabilistic predictions of local flexural and shear cracking in a concrete elements. The colour bars represent the probability of cracks at nodal locations: (a) simulated shear cracking pattern, (b) probabilistic prediction of the shear crack pattern using a convolutional Neural Network, (c) Photo of a flexural crack in an area sensing skin, and (d) probabilistic prediction of the flexural crack using a feedforward neural network.

In summary of this subsection, it is clear that the use of inverse methodologies in static SHM applications is pervasive. Meanwhile, a number of inversion-based modalities are still emerging – or are yet to emerge. Indeed, given the number of potential static data sources available at present, there exist substantial opportunities to investigate or formulate new inversion based modalities. In the light that some physical models for various underlying physics remain unavailable (either in open source or in general), the use of learned models may bridge this gap. Lastly, there currently exists tremendous opportunities in the areas of data fusion and joint imaging, which remain predominately unexplored in the area of

static inversion based SHM [281].

3.5.3 Dynamical inverse problems in SHM

The use of dynamical data for monitoring the health and condition of structures is well established [282]. For this, a number of data sources are available, for example discrete acceleration, strain, displacement measurements, and recently, coupled electromechanical impedance via piezoelectric transducers [283, 284]. In the case of typical civil infrastructure [285], the ability to actively excite monitored structures is pragmatically challenging due to the extreme magnitude of the excitation required to attain a distinguishable response. For this reason, ambient monitoring methodologies have gained significant popularity in recent years [286–288]. Irrespective of the dynamical monitoring approach used, extracting dynamical structural properties of interest can be viewed as an ill-posed inverse problem [289]. The ill-posedness of such problems results from a number of actualities, not limited to uncertainties in environmental conditions (wind, temperature, ground conditions, humidity, etc.), traffic, measurement noise, the discrete nature of measurements, material characteristics and numerical modelling error.

One of the most pervasive frameworks used in solving dynamical inverse problems is model updating, which generally aims to match a physics-based model (such as a representative finite element model) to measured dynamical data [208, 224], commonly using a form of modal analysis [290–292]. The physics-based techniques are particularly efficient in providing higher accuracy when testing is restricted. It is often the case that reconstructing the dynamical SHM properties of interest proves difficult, requiring an innovative approach; some proposed frameworks have included advanced optimisation protocols [293] and mode decomposition/superposition [294, 294]. Alternatively, the use of phase space [295], state space [296], singular value decomposition, λ-curves method [297][298], autoregressive [299, 300], Gaussian process [301], and Bayesian/stochastic approaches [302–305] have proven successful. As noted in the previous subsection, one metric for assessing the progress in this field is the number of works aiming to optimise sensing information, for example in [267, 306–313].

In the past twenty years, guided wave based modalities have emerged as a viable approach to dynamical inversion based SHM [314–321]. Common physical manifestations of guided waves include Lamb waves (propagating through thin shell and plate structures) [322], Rayleigh waves (surface waves) [323], and shear waves. Generally speaking, SHM systems consist of transducer systems used for actuation and measurement accompanied with an inversion algorithm aiming to reconstruct SHM causalities of interest. Owing to a number of numerical challenges, conventional solutions to related inverse guided waves problems are generally not feasible [318]. Consequentially, alternative methodologies have been proposed including, for example, inverse filtering [324], reverse time migration [325, 326], Bayesian/probabilistic [327, 328], amongst emerging inversion approaches.

While the use of dynamical inversion based techniques is well established in conventional monitoring, in some areas (such as guided wave monitoring) it remains in the early stages of development and affords numerous research opportunities. It is worth mentioning that with the advent of modern machine learning methods, we can only anticipate significant advances in forthcoming years as trained networks are now capable of address-

ing key SHM challenges related to, for example, model error estimation/correction [329] and reducing computational demands associated with many SHM facets [330, 331].

3.5.4 Computer vision inverse problems in SHM

In addition to the discussed static and dynamic inverse problems in SHM, computer vision-based SHM methods have become an emerging field in inverse engineering problems in approximately the past decade. Relying on digital images and videos, vision-based SHM techniques enable affordable and rapid structural prognosis. The concept of vision-based inverse problems is straightforward: visual information from the external surfaces of structures is captured through digital cameras, serving as input data for computer vision algorithms in detecting, localising, and quantifying structural damage in a variety of contexts.

Computer vision-based inverse problems can be either static or dynamic in nature. In dynamical environments, a digital camera is treated as a vision sensor measuring dynamic structural responses. Instead of directly capturing the structural vibration through contact-based sensors (e.g. accelerometers), vision-based algorithms can track structural responses through a video stream. For example, in [332], researchers applied a video feature tracking technique to measure the pixel movements of a steel girder in a football stadium under a service load using a consumer-grade digital camera. These movements were then converted into displacements using a scaling factor. Similar efforts have been reported in [333, 334]. Furthermore, through the usage of cameras as displacement sensors, other key structural features, such as natural frequencies/mode shapes [335, 336], beam influence lines [337], and bridge cable loads [338] have been be estimated.

In addition to tracking the surface motion, computer vision algorithms can offer rapid and reliable inspections against structural damage such as cracks [339], concrete spalling [121], steel corrosion [340], and other structural deterioration [341]. To make it viable, researchers develop vision-based algorithms to scan and extract damage-induced visual features either across the entire image scene or within a small predefined image patch (e.g. region of interest) that is prone to structural deterioration. In general, the image-based damage extraction techniques can be categorised as: (a) machine or deep learning-based methods; and (b) non-learning based methods.

The idea of machine or deep learning-based (computer vision) methods is to train a damage detection classifier through an image dataset with pre-labeled structural damage. Then, the classifier is applied in characterising structural damage from newly captured images. Some of the successful applications include detection of concrete cracks [342, 343] and spalling [121], steel cracks [344], bolt loosening [345, 346], steel surface defects [121], pavement cracks [341], and complex situations where multiple damage types exist [120]. In contrast, non-learning based methods can directly pick up image features caused by structural damage, and hence do not require any prior knowledge in training the classifier. For instance, fatigue cracks in steel bridges can be identified through crack breathing behaviour [347]. Also, loosened bolts in steel connections can be quantified by extracting the differential features provoked by bolt head rotations [348]. Fig. 3.6 illustrates an example by comparing two images of a steel connection at different inspection periods to extract the differential features provoked by the loosened bolts.

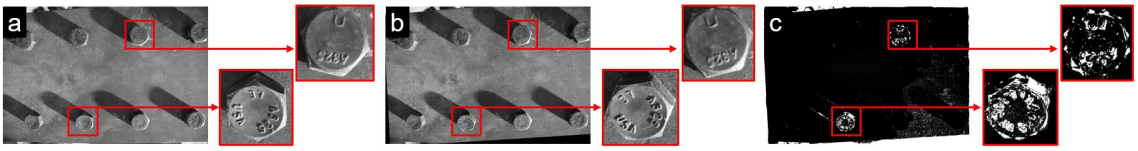


Figure 3.6: An example of vision-based bolt loosening detection where image (a) and (b) are images of a bolted connection taken at different inspection periods. Two loosened steel bolts are shown in the blow-up figures with counterclockwise rotations in their bolt heads. Using a series of image processing techniques, the differential features caused by the bolt loosening can be identified in c). Detailed discussion can be found in [348].

Computer vision-based methods can be extended for damage detection and pattern recognition in full scale civil structures. Utilising the platform of unmanned aerial vehicles (UAVs), the on-board UAV camera can rapidly scan the structure including the locations that are challenging to be accessed by traditional contact-based sensors. For example, researchers in [349, 350] applied UAVs and vision algorithms to leverage effective approaches for post-earthquake building safety inspections. Similar efforts have been reported for inspecting dams [351, 352], tunnels [353], and railways [354]. Other researchers adopted satellite images to examine damage status over a larger scope of work (i.e., multiple buildings at the community level) after natural disasters such as flood, earthquake, volcanic eruption, hurricane, and wildfire [355, 356].

UAV platforms are also capable of collecting a large volume of images of civil structures under different camera angles through automated route planning. Such an advantage can be further enhanced by a computer vision workflow, termed photogrammetry, for the purpose of reconstructing the 3D model of the structure. Relying on structure-from-motion with multi-view stereo (SfM-MVS) algorithms [357], photogrammetry technique can create a 3D structure model (e.g. a 3D point cloud) based on 2D images. Photogrammetry leverages several potentials in inverse SHM problems. For instance, in recent work, [358] created a dense point cloud of a building in the construction site based on UAV images. Then the point cloud was integrated with a building information modelling (BIM) model for labelling the structural components in original UAV images. In [359], the researchers utilised the dense point cloud to assist the creation of a finite element model of a masonry bridge. The authors argued the benefit of using the point cloud was twofold: the point cloud depicted the accurate geometric information of the bridge and offered the results of bridge crack distribution. In [360], researchers leveraged the photogrammetry workflow to find concrete cracks and spalling of a highway bridge. Lastly, in the context of the 2021 Hernando de Soto Bridge incident [361], where a large crack was discovered in a "fracture critical" element by a private engineering firm – yet, previously identified approximately two years earlier by a local operating a commercially available drone: the use of coupled UAV/computer vision approaches to SHM may be more valuable than ever.

3.5.5 Digital twins and outlook

As this section has illustrated, the use of inverse methodologies in SHM is both well established and an area of active development. With the rapid digital transformation of

structural assessment and infrastructure asset management, the emergence of numerous digitally-inspired technologies will play a key role in the future trajectory of inverse problems in SHM. At the forefront, *digital twins* have been the source of increasing research and industrial interest in recent years [362]. While the scope of digital twins' applications spans beyond SHM alone, its basic aim is to provide information on the current or future state of an asset by combining real-time data and a physical/data-driven model offers many potential avenues for engagement with the inverse problems community. Nonetheless, in specifically considering a classical SHM application, such as damage localisation [363]: developments stemming from the inverse community including, for example, state estimation [364–366], uncertainty/model error approximation/compensation [329, 367, 368], regularisation [369], and model reduction [370], have excellent potential for enriching or enhancing digital twin frameworks.

As a whole, the future outlook for the integration and advancement of inverse methodologies in SHM is very bright. Indeed in the past 20 years, we have seen an exponential increase in high-performance computing and graphical processing units development and assimilation into modern civil and mechanical engineering applications [371, 372]. Coupled with powerful inverse frameworks for large-scale problems (e.g. Krylov solvers [373] and distributed computing [374]) and machine learning [375], we can only expect a steady increase in (a) the breadth of inverse problems the SHM community is able to address and (b) and an evolution in innovative inversion-based approaches to solving increasingly challenging SHM problems.

3.6 Smart materials and structures

Smart, *self-sensing* materials have received immense attention in recent decades [376–378]. A material is said to be self-sensing if it exhibits a property change in response to external stimuli. These materials are able to intrinsically report on their health or condition in a spatially continuous manner and with less hardware/instrumentation burden than traditional point-based sensing technologies (e.g. strain gauges, piezoelectric patches, accelerometers, etc.). In structural engineering, external stimuli are often mechanical effects such as deformation, damage, or loads. Hence, integrating smart materials into next-generation structures may allow for unprecedented health monitoring and diagnostics. A discussion on smart materials in structural contexts may therefore also be considered as a subset of the preceding discussion on SHM (see Section 3.5). Nonetheless, we will treat it as a distinct topic in the forthcoming due to its unique inverse problems.

Although self-sensing is an umbrella term encompassing many different physical effects, the *piezoresistive effect* has perhaps received the most attention to date (see recent reviews [376, 379, 380]). Piezoresistive materials are so-named because they exhibit a change in electrical conductivity (or its inverse, resistivity) upon deformation. This means that every point of the material is capable of relaying information on its mechanical state. Damages such as voids, ruptures, or fractures can also be detected since the removal of material represents a conductivity loss. But spatially-continuous piezoresistive sensing presents two challenges of relevance to inverse problems: (a) It is not practical to instrument electrodes at every point on a structure, which means that it is necessary to deduce conductivity distributions from a finite set of measurements. And (b), even if we could

recover a spatially continuous mapping of the conductivity, electrical properties are of little consequence to the structural engineer. We would much rather know the underlying mechanics that give rise to an observed conductivity distribution. We will address both of these inverse problems. First however, a brief summary of the physics and piezoresistive modelling approaches is given. It will be seen later that modelling techniques are essential to solving inverse problem (b) above.

3.6.1 Piezoresistive nanocomposites

Many materials intrinsically exhibit piezoresistive properties. For example, carbon fibre-reinforced polymers (CFRPs) are well known to change conductivity when loaded elastically [381, 382]. Here however, we will instead focus on materials that have been *engineered* to be self-sensing; that is, an additional constituent has been added to the material system without which it does not exhibit piezoresistivity. This is most commonly done by adding a conductive phase to a non-conductive matrix such as polymers (including structural polymers like epoxy vinyl ester [383], polymeric thin films for use as sensing skins [384], laser-induced graphene inter-layers in continuous fibre composites [385, 386], and even polymer binders in energetic materials [387, 388]), cements [389], or ceramics [390]. Electrical transport is then a consequence of percolation – the composite conducts electricity when enough fillers have been added to form an electrically connected network. Because the percolation threshold decreases with aspect ratio, fillers with ultra-high aspect ratios like carbon nanotubes (CNTs) are popular. There are considerable challenges associated with manufacturing CNT-based nanocomposites such as achieving good dispersion – ultra-small fillers such as CNTs have a tendency to agglomerate which can degrade the mechanical properties of the composite. But manufacturing is outside of the scope of this manuscript and is well-covered elsewhere [391, 392].

Considerable effort has also been devoted to the development of piezoresistivity models – computational and/or analytical means of predicting how conductivity changes for a prescribed strain. These efforts can be broadly categorised as (a) equivalent resistor network models [393–395], (b) computational micro-mechanics models [387, 396–398], or (c) homogenised macroscale models [399–401]. In (a) equivalent resistor network models, high aspect ratio fillers like CNTs are represented as either straight or wavy/curved sticks in a micro-domain [393–395]. These sticks are discretised into resistors based on the length, diameter, and conductivity of the fillers, and junctions between nearby fillers are discretised into resistor elements based on the equivalent resistance felt by an inter-filler tunnelling electron [402]. The conductivity of the nanocomposite can then be calculated from the overall resistance of the discretised nanofiller network and the dimensions of the micro-domain. For a given deformation of the micro-domain, the translation and rotation of the fillers can be calculated by treating them as rigid inclusions [403]. Post-deformation, the conductivity of the micro-domain is recalculated thereby allowing piezoresistive properties to be predicted.

Second, (b) computational micro-mechanics models use computational means to simulate both phases of the composite – the non-conductive matrix and the conductive fillers [387, 396–398]. Because of this, and unlike equivalent resistor network models, computational micro-mechanics models can incorporate nuanced effects such as nanofiller-to-matrix

debonding, nanofiller deformation, etc. A limitation of this approach is computational expense due to individual nanofillers and the enveloping matrix being explicitly simulated. It is therefore difficult to scale these models to structural levels.

And third, (c) homogenised macroscale models describe the conductivity of the nanocomposite as an analytical function of the strain state without simulating individual fillers. Rather conductivity/resistivity-strain are coupled through analytical functions based on excluded volume theory [400] or via 'constitutive' tensors (note that these are not proper constitutive relations because conductivity and strain are not energy complements) [401, 404, 405]. This approach is therefore less computationally expensive than equivalent resistor network and computational micro-mechanics models. Importantly, homogenised approaches can be readily integrated with structural analysis tools such as the finite element method, thereby allowing for macroscale piezoresistive analyses. As will be discussed in Section 3.63.6.3, this allows for strain recovery via piezoresistive inversion. Despite these advantages, analytical models suffer from having to make assumptions regarding average inter-filler spacing, average orientation of nanofillers, and the need for calibration data.

3.6.2 Conductivity imaging via EIT/ERT

As discussed in Section 3.5.2, conductivity imaging modalities such as EIT (or DC resistivity imaging via ERT) have been explored for health monitoring in structural materials [380]. EIT is a natural complement for piezoresistive materials because it allows for the spatial localisation of damage and the *mapping of deformation and strain.* There are several factors that make the EIT interesting to pair with self-sensing materials: (a) This combination allows for sub-surface strain imaging. That is, a myriad of techniques exist for monitoring surface strains such as strain gauges, DIC, holographic methods, etc. Tools for studying sub-surface strains, however, are much more limited, often require ionising radiation, and can be costly (e.g. volumetric strains via X-ray digital volume correlation [406]). (b) Traditional 'limitations' of EIT can be strengths in structural applications. For example, the simplest implementation of EIT favours spatially smooth solutions. This is obviously undesirable when imaging discontinuous features with distinct boundaries such as organs in a living body. Strain fields, however, are often smoothly varying thereby playing into EIT's strengths. And (c), because piezoresistive materials are engineered, we can leverage our knowledge of their piezoresistive response to build bounds into the EIT inverse problem. Two examples of strain imaging via EIT in self-sensing polymeric composites are shown in Fig. 3.7. Both examples leverage knowledge such knowledge of conductivity change bounds to build constraints into the solution space. Beyond these advantages for structural imaging, the pairing of EIT/ERT with self-sensing materials may also have keen, as-of-yet unrealized potential for extreme loading (particularly self-sensing energetic materials [387, 388]). That is, it may be illuminating to image energetic materials as they detonate. However, most prevailing imaging modalities have not the temporal resolution to capture these fleeting moments and require hardware that is too expensive to risk damaging. EIT, on the other hand, can have ultra-high temporal resolution (on the order of hundreds of microseconds for optimized systems [407]) and uses only low-cost hardware (i.e. sacrificing the hardware during a detonation is of virtually no

financial consequence). Thus, there may be much exciting potential for overlap between smart materials + EIT and extreme loading as described in Section 3.4.

There have been many studies on the topic of EIT and piezoresistive materials. A few representative examples are summarised in this manuscript, but interested readers are directed to a recent review for a more in-depth discussion [380]. Some of the first work in this area made use of self-sensing nanocomposite *sensing skins* produced by a layer-by-layer fabrication technique [384, 408]. These skins were applied to substrates, and EIT was then used to identify and spatially localise deleterious effects including mechanical etching, pH exposure, impact damage, and residual strains from impacts. Further on the topic of polymer-based composites, thin pressure sensors were produced by embedding a non-woven textile modified with CNTs into a soft elastomer. EIT was then employed to visualise various pressure distributions including non-uniform distributions [409]. EIT was also recently applied for damage detection in a ceramic-based composite that was modified with micron-sized waste-iron particles [410]. And as a final representative example, self-sensing cement composites have been produced by spray-depositing a CNT-modified latex on the aggregate phase. EIT was then successfully used to localise various damages in the material [411].

3.6.3 Piezoresistive inversion

Even though solving the EIT inverse problem allows for the spatially continuous visualisation of mechanical effects in these materials, this poses an obvious problem – structural engineers are generally not interested in conductivity. They would much rather know the spatially-varying components of the strain tensor, stress tensor, or precise damage characteristics since these factors drive structural analyses and health assessments. Recalling also that various macroscale piezoresistivity models exist as described in Section 3.6.1, we can formulate another inverse problem as follows: "for a given EIT conductivity distribution (or, more directly, for a given set of EIT boundary data) and with an accurate model of conductivity-strain or conductivity-damage coupling for a particular material, can the precise mechanics of the system be recovered?"

Both of these piezoresistive inversion problems—strain recovery and damage recovery—are challenging. The former is challenging because, under ideal circumstances, we seek six components (in 3D) of a strain tensor from a scalar conductivity field. Prospects can be improved somewhat by instead seeking the displacement field (i.e. three unknowns in 3D) from the conductivity data and making use of reasonable assumptions (e.g. plane strain and plane stress) where applicable, but the displacements-from-conductivity inverse problem is nonetheless under-determined. The challenge is exacerbated by the fact that circumstances are never truly ideal; conductivity and conductivity changes are not exactly isotropic, and EIT cannot image individual components of a conductivity tensor. Even for the case of damage imaging, an accurate model of material breakage-induced conductivity change is needed. For simple nanofiller/matrix phase nanocomposites (i.e. without reinforcing fibre), material breakage can be treated as a cessation of conductivity. For more complicated material systems such as nanofiller-modified continuous fibre composites, however, material breakage-induced conductivity changes must account for factors such as anisotropy and residual post-damage conductivity due to inter-laminae contact.

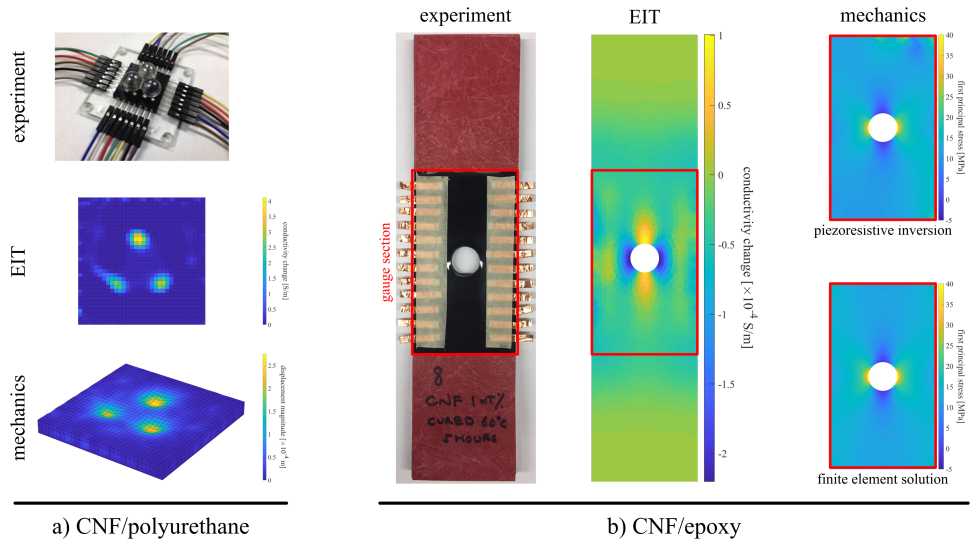


Figure 3.7: Examples of EIT and piezoresistive inversion applied to self-sensing nanocomposites. a) A soft CNF/PU is deformed by rigid, non-conductive indentors [412]. EIT is then used to image the deformation-induced conductivity changes, and piezoresistive inversion is used to recover the displacement field (multiplied by a factor of five for ease of visibility). b) A hard CNF/epoxy is loaded in tension with a stress raiser at its centre [413]. EIT is again used to image the conductivity change. Lastly, piezoresistive inversion is used to recover the underlying displacement field. With knowledge of the material's elastic properties, strains and stresses can be spatially mapped. The first principal stress of the guage section is shown here along with comparison to a traditional FEM solution for validation.

And even if a suitable damage model is developed, the inverse solver needs to be capable of reproducing potentially complex damage shapes that are not readily amenable to parameterisation. Both strain and damage recovery are additionally hampered by the fact that EIT does not produce accurate conductivity distributions in an absolute sense or a spatial sense.

Despite these challenges, the piezoresistive inversion problem has received some attention. An initial effort used an analytical inversion framework predicated on iteratively minimising the l_2-norm of an error vector between a predicted and observed conductivity distribution [414]. Although this work was entirely computational and limited to simple deformations and infinitesimal strains, it nonetheless demonstrated that piezoresistive inversion was possible. The next work in this area used EIT to image strain-induced conductivity changes in a carbon nanofiber (CNF)-modified polyurethane (PU) composite [412]. Three marbles (i.e. comparatively rigid non-conductive indentors) were pushed into the CNF/PU as EIT measurements were taken. A similar analtyical approach was used to reproduce the displacement field. Two important factors differentiated this study – experimental validation of piezoresistive inversion and successful application to materials undergoing finite strains. Later works looked at utilising metaheuristic algorithms for solving the strains-from-conductivity inverse problem in a CNF-modified epoxy [413, 415].

The CNF/epoxy was moulded in the shape of a plate with a hole and loaded in tension, causing strain concentrations in the vicinity of the hole. Genetic algorithms, simulated annealing, and particle swarm optimisation were explored because it was observed that the analytical formulation failed to converge to the physically correct solution for this more complex loading state. It was found that genetic algorithms performed the best for this inverse problem, but all methods compared favourably to DIC experimental validation. Because epoxy is relatively brittle, these studies were necessarily limited to infinitesimal strains. Despite the successes of the preceding studies, they were all limited to electro-mechanically isotropic materials. Translating these capabilities to electro-mechanically anisotropic materials remains a daunting challenge. Fig. 3.7 summarises results for strain recovery via piezoresistive inversion.

Some work has also been done for damage recovery via piezoresistive inversion. Recall that a key challenge with the damage recovery inverse problem is shape parameterisation. To that end, these preliminary studies have considered relatively simple damage cases. For example, various machine learning algorithms were used to categorise three damage conditions in a self-sensing bone cement directly from EIT boundary voltage data [416]. Thus, self-sensing materials also have overlap with machine learning methods (see Section 3.7). Bone cement is poly(methyl methacrylate) (PMMA) used to facilitate robust contact between an othopaedic implant (e.g. a total joint replacement) and hard bone. Failure of the PMMA interface is often difficult to detect via radiographic imaging, hence the motivation for alternative diagnostic tools. This clarification aside, the parameterisation was relatively simple in this case – only four distinct states were possible (three damaged plus one healthy). The combination of EIT, self-sensing PMMA bone cement, and machine learning allowed for correct damage classification with over 90% accuracy. In another study, image recognition-based machine learning was used to identify, size, and localise through-hole damage to a self-sensing composite plate [417]. The image recognition algorithm was trained using computationally generated EIT images on a simple square domain punctured by a random number of randomly sized circular holes. The trained network was able to adeptly predict through-hole size, location, and number from EIT conductivity maps with good accuracy – likely better accuracy than human interpretation of EIT images. But this again utilised a very simple damage parameterisation (i.e. needing only to predict hole number, radius, and in-plane coordinates). As a final example, a recent study looked at delamination shaping from EIT images in CNF-modified glass fibre/epoxy laminates [418]. In this study, delaminations induced by low-velocity impacts were parameterised as ellipses of unknown major and minor axes and centred at unknown in-plane coordinates. A genetic algorithm was used to inversely determine these parameters by minimising the l_1-norm of the difference between experimentally collected EIT boundary voltage data and boundary voltage data predicted by a computational model of the damaged domain. Destructive analyses of the post-impacted laminates revealed that the GA-predicted damage state much more closely matched the actual delamination size and shape than the EIT conductivity images. This third example of damage recovery is particularly noteworthy because it represents a much more realistic damage state.

In summary, this section has looked at smart, self-sensing materials from the perspective of structural inverse problems. Two noteworthy inverse problems were discussed – the EIT inverse problem and the strain/damage recovery problem. The former has been

extensively researched in other fields (e.g. [419]). The latter, however, is much more recent and has only been the subject of a few precursory studies. Much work remains to be done regarding the inversion of electrical data to obtain underlying mechanical effects. Nonetheless, it can be seen that the field of smart materials in structural applications has much potential and cross-cutting overlap with other topics of this article including extreme loading, SHM, and machine learning.

3.7 A look forward: machine learning and education

This paper has evidenced the pervasiveness of inverse problems and methodologies used in the field of structural engineering. Yet, the following remains to question: "what is guiding the future trajectory of inversion in structural engineering?" In this section, we examine what we foresee as the two most influential future areas in addressing this question: machine learning and education.

3.7.1 Machine learned inversion

Many areas of structural engineering rely heavily on applied mathematics and science – from the integration of material models within finite element frameworks to experimental measurement of structural response excitation. From a broad scientific perspective, there are two paradigms to research, either (a) the Keplerian paradigm (data driven, obtaining discoveries via data analysis) or (b) the Newtonian paradigm (first principles, discovery through fundamental principles) [420]. Without question, structural engineering research uses both principles.

Often, first principles approaches are manifested via partial differential equations (PDEs) and their analytical or numerical solutions. However, both in research and practice, obtaining solutions to PDEs can be infeasible or intractable, for example owing to computational demand, a dearth in available numerical regimes and/or the "curse of dimensionality" [421]. In such cases where engineering problems are governed by such PDEs, solving a related inverse problem using a conventional methodology would also be a dubious task. When faced with this situation, we are most likely constrained to adopting a Keplerian approach.

Unmistakably, machine learning has provided the science and engineering communities with a powerful tool for data-driven analysis, prediction, assessment, and significantly more. Structural engineering research has also greatly benefited from machine learning, especially in the areas of performance assessment [65], SHM [150] and analysis of various structural phenomena [422]. Yet, significantly less attention has been paid to the use of machine learning for solving inverse problems in structural engineering outside of areas such as SHM and NDE (as identified in Section 3.5).

Exemplifying this reality, structural design highly under utilises machine learning and data science. Design is traditionally associated with an iterative nature, in which various structural concepts are tested conceptually until a prevailing option is identified which adheres to constraints initially identified. The nature of this iterative design process has been understood in the past using both positivist, pragmatic and post-modernist epistemologies ranging from Simon's "science of the artificial" [423], Schön's "reflexive practice"

[424] and Buchanan's "placements for contextualisation" [425] respectively. The complexities involved in design from satisfying conflicting demands to exercising appropriate judgements is hence often attributed as being an innate human skill [8].

However, recent developments successfully solving inverse problems using data-driven approaches suggest that such methodologies could also be incorporated within structural design [165]. This challenges the notion that design is an exclusive human ability, a development which mimics the success of self-driving cars through data-driven approaches [426]. What is typically considered "intuition" or "engineering judgement" may in fact be recalling "data-points", committed to long-term memory through the process of experience, that are suitable solutions based on the unique set of constraints one is presented with [423]. Indeed, supervised machine learning was originally understood to create mapping functions that correlate a set of inputs (a specific set of constraints for a design situation) to associated outputs (viable structural solutions) by learning from a given data set (experience) [427]. We note that this view has significantly evolved with the onset of deep-learning, as well as the use of unsupervised and reinforcement learning regimes.

We therefore believe that this inverse problem perspective evidences the need for research on the development of machine learning tools for structural design. Data-driven approaches are often dismissed because they are "black-boxes" which lack a scientific rationale for their outputs. However for design, this criticism might be unwarranted, and instead highlights the dogmatic concentration within academia on exclusively solving forward problems, which due to their well-posed nature, lend themselves to engineering science thinking. Recently, researchers have started investigating data-driven models. In one example, neural networks and clustering were applied to form a bridge and navigate the design space and shortlist viable and fitting solutions [102, 103]. Other examples include researchers applying machine-learning models to build suitable structural predictors for conceptual design related to building massing [104].

The former developments are, in our view, only a starting point for the employment of machine learned models used in solving inverse problems in structural engineering as a new horizon of data-driven approaches emerge. Especially, for the cases involving intractable forward problems, model reduction techniques have been promising [153, 428–430], but these are either difficult to design by hand or are restricted by overly simplistic assumptions. Here, data driven approaches are a powerful alternative to compensate for modelling errors [329, 367, 431] or reducing computational cost of iterative optimisation schemes by model approximations [172, 432]. Finally, we note that recent developments in geometric learning extend deep networks on Euclidean meshes to general meshes, such as finite elements, by a embedding them into graph structures essentially utilising the underlying geometry [433, 434]. This opens the possibility to extend many data driven approaches to complex structural problems.

3.7.2 Inverse methodology in structural engineering education

Over the last 100 years, engineering education has experienced a number of fundamental shifts: a shift in focus away from design to engineering science in the 1960s, the rise of outcome-based accreditation in the UK and USA in the 1990s, along with a re-emphasis of teaching design through capstone projects in the 2000s [435]. There also is the continued

tension between teaching graduates both the technical knowledge as well as the interpersonal skills demanded from industry deemed necessary to become effective designers [175]. To this day, there exists the debate on how to find the required balance between *knowledge-that* and *knowledge-how* as identified in Section 3.3 [436]. For civil and structural engineering disciplines, we believe that some of these challenges might be addressed by communicating the existence of inverse problems, their pervasive occurrences as shown by the sections above, and teaching the various methods and techniques for solving such problems.

The dominance of engineering science within university curricula today, which primarily focuses on identifying and solving forward problems, might unintentionally generate the wrong supposition that all problems in engineering are well-posed with idealised assumptions. Without adequately addressing the existence of inverse-problems, and their distinctively qualitative differences with forward problems, it is easy to mistakenly assume that, for example, structural design is the application of such "forward problems". However, the idea that engineering is simply "putting theory into practice" [8] or "applied science" [437] has been strongly argued against by numerous engineers and philosophers [1, 191, 438, 439]. The challenge which students face when dealing with real-world design problems might be accounted for by the fact that during the majority of their engineering education, they might lack a conceptual framework to adequately demarcate design from analysis. Similarly, in order to engage with other promising fields of structural engineering, such as structural health monitoring, blast engineering and smart materials, educating students on inverse problems is crucial.

Hence, a possible improvement for current civil and structural engineering curricula is introducing students to the existence of forward and inverse problems, how they relate to one another and provide examples where each type of problem arises and how to solve them. As identified previously, this will also potentially require teaching students a host of new skills, especially if data-driven models continue to be effective tools for solving inverse problems, as is the case in structural health monitoring and increasingly likely in design. More importantly, especially when taking into consideration the recent developments in data science ranging from CNNs [86], transformers [89] and graph neural networks [440], there exists a vast spectrum of knowledge and applications we may not even be aware of.

As a matter of fact, in terms of research, we are perhaps faced with a unique situation in academia today. Although only time will tell, it could be argued that similar to how the "invention" (or discovery) of calculus in the 18th century was instrumental in providing us necessary tools for solving forward problems, resulting in material models and PDEs which allow the creation of complex finite-element methods, so too might the rise of machine-learning and data-science, which is only now starting to gain serious attention in mathematics [420], allow a more rigorous treatment of solving inverse problems. By realising the pervasiveness of inverse problems in structural engineering, but also the fundamental differences with forward problems, there is potentially a vast, untouched and exciting realm of research which awaits.

3.8 Conclusion

This article aimed to demonstrate that numerous structural engineering sub-fields may be either fundamentally or partially viewed as inverse problems. It was shown that this concept is well accepted in, for example structural health monitoring; however, sub-fields such as structural design are not commonly (formally) defined as inverse problems. We argue that, by shifting this paradigm in structural engineering academia and industry, we may collectively capitalise from the rich methodologies and approaches already established in the inverse problems community. This beneficial relationship between structural and inverse communities is expected to pay exponential dividends as new tools, such as machine learned models, emerge and develop – offering new opportunities for solving previously inaccessible, intractable, and/or unforeseen structural challenges.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Data statement

Data used in this article are available from the author(s) upon request.

Chapter 4

Article: "Influence zones for continuous beam systems"

Foreword commentary

This article titled "Influence zones for continuous beam systems" was published in *Structures* on August 9th, 2024 [441]. The paper introduces a novel concept known as the structural influence zone and demonstrates how it can help identify the pertinent structural information for the design of individual elements that comprise a structural system. The need for this kind of information arose during the development of the generalisable machine learned structural design model presented in Chapters 5 and 6, and due to its significance, it merited its own article. The concept also demonstrates how the inverse problem perspective introduced in Chapter 3 lends itself to the development of novel concepts such as the influence zone presented here. The work was entirely conducted by Adrien Gallet, with co-authors Andrew Liew, Iman Hajirasouliha and Danny Smyl providing supervision along with reviewing and editing the final draft.

Abstract

Unlike influence lines, the concept of influence zones is remarkably absent within the field of structural engineering, despite its existence in the closely related domain of geotechnics. This paper proposes the novel concept of a structural influence zone in relation to continuous beam systems and explores its size numerically with various design constraints applicable to steel framed buildings. The key challenge involves explicitly defining the critical load arrangements, and is tackled by using the novel concepts of polarity sequences and polarity zones. These lead to the identification of flexural and shear load arrangements, with an equation demarcating when the latter arises. After developing algorithms that help identify both types of critical load arrangements, design datasets are generated and the influence zone values are extracted. The results indicate that the influence zone under ultimate state considerations is typically less than 3 adjacent members for any given beam within a continuous system, rising to a maximum size of 5 adjacent members for any given continuous beam. Additional insights from the influence zone concept, specifically

in comparison to influence lines, are highlighted, and the avenues for future research, such as in relation to the newly identified shear load arrangements, are discussed.

4.1 Introduction

Influence lines, which derive from Betti's theorem established in 1872 [442], are a well-established tool in structural engineering to identify the worst-case load placement on structural systems [443–445], and are widely applied in research related to continuous beam systems [446, 447], rigid frames [448], bridge engineering [449] and structural health monitoring [450, 451]. Influence zones, on the other hand, also known as zones of influence, are an established concept within the field of geotechnical engineering, helping to identify the area of engineering soils likely to be affected by loading due to sub- and superstructure construction [452], providing geotechnical engineers valuable design insight in deep foundation design [453, 454], settlement estimations [455] and preserving groundwater supplies [456].

Despite the obvious discipline link between geotechnical and structural engineering, the equivalent use of an influence zone in structural engineering does not exist in literature. Here, the term *structural influence zone* would refer to the zone in which applied forces, stiffness provisions and support conditions, or changes thereof, impact the design of the surrounding structural system.

The dearth of literature on such an *influence zone* is surprising. For instance, the concept of influence zones also exists outside of geotechnical literature. Some examples are available in research related to the study of saltwater-freshwater interfaces [457], harmful emission concentrations at traffic intersections [458], reverse k-nearest neighbour algorithms [459, 460], propagation path of surfaces waves [461] and ecological studies on below-ground plant competition [462].

Furthermore, one can readily identify situations where knowledge of the *influence zone* could be beneficial in design. For example, the size of the influence zone could allow an engineer to avoid the need to model an entire structure for the design of a single element whilst being confident that structural information outside the influence zone is irrelevant, with direct applications in multi-disciplinary projects [463]. The impact of late design changes (due to changes in loading or structural provisions), which are known to cause significant time lags until the associated engineering analysis is completed [464], could be more effectively addressed by knowing immediately the selection of members impacted by the said design change. Similarly, engineers are typically required to verify assumptions made in preliminary design [14]. In such cases, the use of an influence zone-based approach could guide what information to incorporate when building an independent model of the design problem. In all of these scenarios, there is valuable design insight to be gained from the *influence zone*.

This article aims to address the above mentioned knowledge gap by numerically introducing the concept of influence zones in relation to continuous beam systems. First, the theory and procedure for evaluating the influence zone will be introduced in Section 4.2, followed by the methodology in Section 4.3 which presents a systematic investigation of critical load arrangements that is key for the evaluation of influence zone values. This allows for the efficient generation of design datasets and the evaluation of their respective

influence zones with results presented in Section 4.4, which are then discussed in Section 4.5. In addition to the *influence zone*, this paper proposes other novel concepts such as *polarity zones*, identifies an entirely new set of critical pattern loads named *shear load arrangements*, and proposes efficient *load arrangement algorithms* for continuous beam systems of arbitrary member size.

4.2 Theory

4.2.1 Overview - continuous beam systems

Consider a continuous beam system, as shown in Figure 4.1, consisting out of m members, indexed by i, which is subjected to ω_i uniformly distributed loads (UDLs) from vector $\boldsymbol{\omega}$, with each member having span length L_i from vector $\mathbf{L}$. When designing this system to identify the minimum required structural properties of the members (size optimisation) denoted I_i to form vector $\mathbf{I}$, it will need to be designed against the worst-case load arrangement (also known as pattern load) from the set of load arrangements $\mathbf{J}$ of size p. The over-restrained nature of this structural system (a function of the support fixity and structural connectivity) renders the continuous beam indeterminate. This means that the performance of the system is a function of the structural properties which need to be evaluated, and generally makes the design process iterative[1]. Literature has well established formulations to design such indeterminate systems [465].

4.2.2 Influence zone definition

Suppose a member within a continuous beam system is designated as the *design beam* by index d as shown in Figure 4.2a), with UDLs $\boldsymbol{\omega}$ (which includes both permanent and variable UDL loads $\omega_{g_k,i}$ and $\omega_{q_k,i}$ respectively) and spans $\mathbf{L}$. In the Eurocodes (such as EN1993 [466] for steel systems), the ultimate limit state of a structural member can be expressed by a utilisation ratio u, which ratios over 1.0 denoting structural failure. Using Eurocodes (such as EN1993 [466] for steel systems), it is possible to determine the true utilisation ratio $u_{d,\text{true}}$ of the design beam d under the critical load arrangement from $\mathbf{J}$ if all UDLs $\boldsymbol{\omega}$ from the entire system are considered as shown in Figure 4.2b). Now suppose only a sub-selection of all UDLs $\boldsymbol{\omega}$ from the adjacently connected spans to that of the design beam d are considered; this sub-selection of UDLs are identified by the discrete integer $k \in \mathbf{Z}$ (where $\mathbf{Z}$ is the mathematical set of all integers) to form vector $\mathbf{K}$. Ignoring all other UDLs, there will be a different critical load arrangement for the design beam d, resulting in a slightly smaller captured utilisation ratio $u_{d,\text{cap}}$ as shown in Figure 4.2c). Whilst $u_{d,\text{cap}}$ will differ from $u_{d,\text{true}}$, the influence zone for member d is defined by $k_{\max} = \max(|\mathbf{K}|)$ (see Figure 4.2c) such that the captured utilisation ratio is sufficiently similar to the true utilisation ratio of the design beam; that is $u_{d,\text{cap}} \approx u_{d,\text{true}}$. Note that $k_{\max} \in \mathbf{N}^0$, with $\mathbf{N}^0$ representing the set of all positive integers and including 0.

In this formulation, it is proposed that there should be some value of $k_{\max}$ that allows for sufficient approximation of $u_{d,\text{true}}$. The theoretical foundation for this is based on the realisation that influence lines approach zero when moving away from a given influence

[1]For more information on the design process of continuous beam systems, see Appendix A

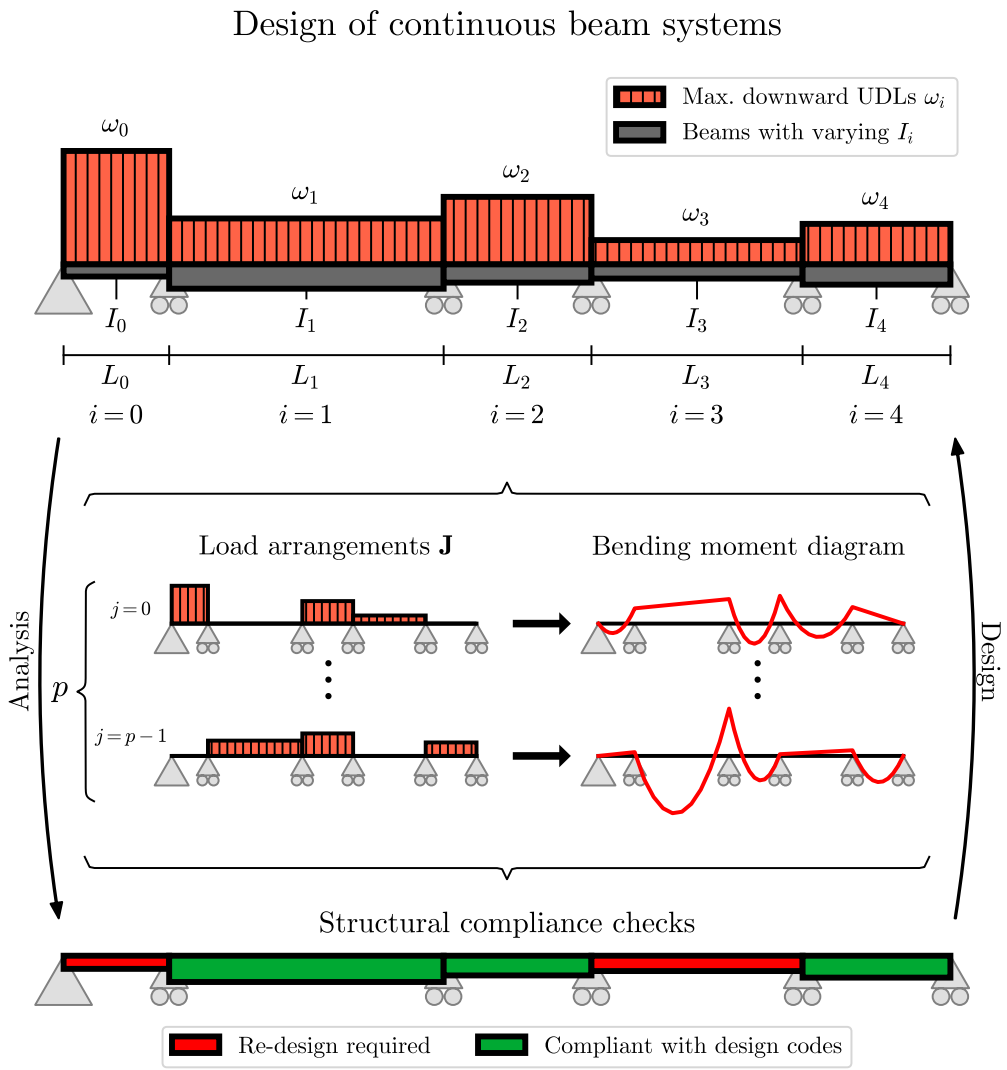


Figure 4.1: An exemplary continuous beam system with $m = 5$ members, subjected to UDLs $\boldsymbol{\omega}$, spans $\mathbf{L}$ and with designed cross-sectional properties $\mathbf{I}$, all indexed by i. The system's indeterminacy requires an iterative design process against various load arrangements $\mathbf{J}$ of size p indexed by j.

line (IL) location as shown in Figure 4.3a). Design information on spans further away from the design beam d therefore have a decreasing influence on the structural response of a design beam d; in other words, there are diminishing returns when considering UDLs $\boldsymbol{\omega}$ increasingly distant from a design beam d. For example, using the lengths, UDLs and members of the continuous beam system in Figure 4.2a), the impact in terms of bending moment response of a UDL at an IL location within the design beam d can be found by integrating the influence line diagram for each span L_i and multiplying it by the spans UDL ω_i as shown in Figure 4.3b). Using the principle of superposition, the moment

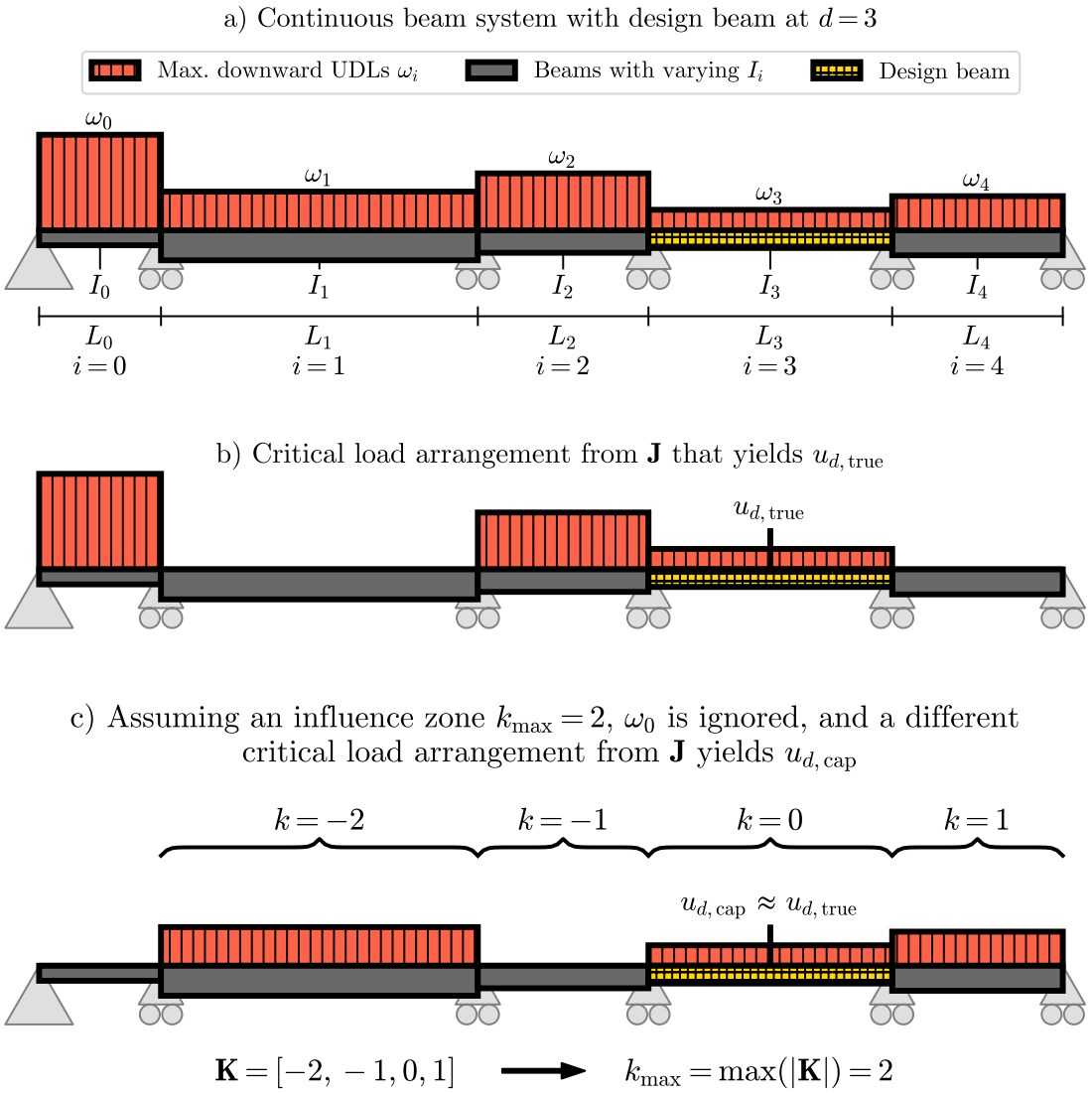


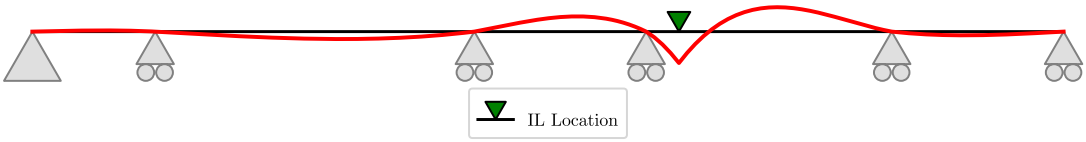
Figure 4.2: Key terminology for influence zones: a) highlights the design beam d, b) the true utilisation ratio of the design beam $u_{d,\text{true}}$, and c) the captured utilisation ratio $u_{d,\text{cap}}$ assuming an influence zone of $k_{\max} = 2$.

contribution about the specified IL location from span $i = 0$ is therefore small, and would be exceedingly smaller for spans even further from the IL location. The influence zone $k_{\max}$ of a member d is therefore defined as a metric of the design information required to sufficiently approximate the utilisation ratio of that member d.

4.2.3 Mathematical formulation

A more rigorous formulation is required to identify UDL values $\boldsymbol{\omega}$ which cease to be part of a member's influence zone $k_{\max}$. As explained previously, the influence zone is es-

a) Bending moment $M_{y,IL}$ influence line (IL) about specified location



b) Moment contribution of each span to IL location: $\int_i M_{y,IL} \times \omega_i$

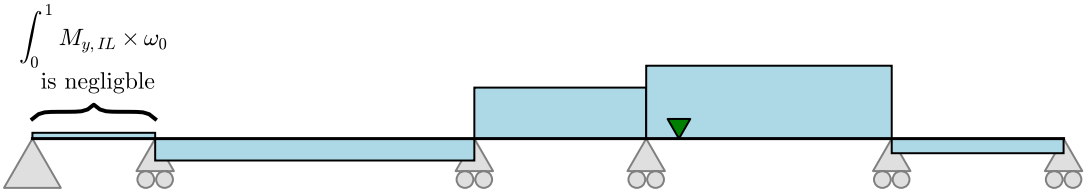


Figure 4.3: The influence of UDLs to the bending moment experienced at an influence line (IL) location becomes increasingly negligible the further away one moves from the IL location.

tablished when the captured utilisation ratio sufficiently approximates the true utilisation value, that is $u_{d,\text{cap}} \approx u_{d,\text{true}}$. For a given continuous beam system as depicted in Figure 4.1, and the design constraints expressed in Equation 4.1,

$$\begin{aligned} w_{\min} &< w_i < w_{\max} \\ L_{\min} &< L_i < L_{\max} \\ I_{\min} &< I_i < I_{\max} \end{aligned} \tag{4.1}$$

the size of the influence zone of a given design beam d shall be defined when the value of $k_{\max} \in \mathbf{N}^0 : k_{\max} \in [0, m]$ **and** all values larger than $k_{\max}$ fulfil the following condition:

$$\left| 1 - \frac{u_{d,\text{cap}}}{u_{d,\text{true}}} \right| \leq \epsilon_{\max}$$

$$u_{d,\text{cap}} = \max \left(\sum_{i=-k_{\max}}^{k_{\max}} \mathbf{u}_{d,i,j}(\boldsymbol{\omega}, \mathbf{L}, \mathbf{I}, \mathbf{J}) \right) \tag{4.2}$$

where $\epsilon_{\max}$ represents the maximum error threshold for the difference between $u_{d,\text{cap}}$, the captured utilisation ratio of the design beam d for a given value of $k_{\max}$, and $u_{d,\text{true}}$, the true utilisation ratio of the design beam d if the contribution of all UDLs of the continuous beam system had been considered. Note that, not shown in Equation 4.3 is that the value of i cannot exceed the maximum number of adjacently lying spans within the system. $\mathbf{u}_{d,i,j}$ is the utilisation ratio contribution function towards the design beam d by member i based on the UDLs $\boldsymbol{\omega}$, spans $\mathbf{L}$, structural properties $\mathbf{I}$ and load arrangements $\mathbf{J}$ indexed

by j. For the particular case of a continuous beam system loaded by UDLs as presented in Figure 4.1 it is possible to define the utilisation ratio contribution function $\mathbf{u}_{d,i,\,j}$ as:

$$\begin{aligned} \mathbf{u}_{d,i,\,j} &\rightarrow \mathbf{D}_{\mathrm{ULS}}(I_d, M_{d,i,j}, V_{d,i,j}) \\ M_{d,i,j} &= \omega_i \; J_{i,j} \int_i \mathbf{M}_{\mathrm{IL},d} \\ V_{d,i,j} &= \omega_i \; J_{i,j} \int_i \mathbf{V}_{\mathrm{IL},d} \end{aligned} \tag{4.3}$$

where $\mathbf{D}_{\mathrm{ULS}}$ represents the ULS steel cross-section design checks based on the appropriate Eurocodes of the material considered (such as EN 1993-1-1 6.2 [466]), I_d represents the cross-sectional properties, $M_{d,i,j}$ denotes the major axis moment while $V_{d,i,j}$ is the major axis shear force of the design beam d, w_i is the UDL, and $J_{i,j}$ is the activation factor (a value 0 or 1) from one possible critical load arrangement J_j of all viable critical load arrangements $\mathbf{J}_{\mathrm{crit}}$. Integrals $\int_i \mathbf{M}_{\mathrm{IL,d}}$ and $\int_i \mathbf{V}_{\mathrm{IL,d}}$ are the integrated influence line values across beam i for a particular influence line location within the design beam d as introduced in Figure 4.3b).

The mathematical formulation as written in Equation 4.2 and 4.3 determines the value of $k_{\max}$ at which the contributions outside of the influence zone become exceedingly small by measuring the difference between $u_{d,cap}$ and $u_{d,true}$ relative to $\epsilon_{\max}$. As $k_{\max}$ increases, the ratio $u_{d,\mathrm{cap}}/u_{d,\mathrm{true}}$ will approach unity, attaining unity if all structural members of the system are considered within the influence zone since the influence of all members is in that case accounted for. If the error threshold $\epsilon_{\max}$ is relaxed, an influence zone less than the total size of the system can be found.

4.2.4 Visualised influence zone evaluation procedure

The evaluation procedure used to find the influence zone value $k_{\max}$ based on a predefined $\epsilon_{\max}$ using Equations 4.2 and 4.3 is visualised in Figure 4.4. This particular example deals with a $m = 15$ homogeneous system (identical spans $\mathbf{L}$ and UDL $\boldsymbol{\omega}$ for all members) consisting out of hot-rolled S355 steel UB178x102x19 cross-sections [467]) each spanning 4 m. It is assumed that an un-factored permanent (dead) $\omega_{g_k,i} = 3\,\mathrm{kN/m}$ and a variable (live) $\omega_{q_k,i} = 15\,\mathrm{kN/m}$ load act on each span as shown in Figure 4.4a).

A detailed explanation of the influence zone evaluation procedure as presented in Figure 4.4 follows next. Suppose one intends to establish the influence zone value for design beam $d = 4$ with an $\epsilon_{\max} = 0.01$. In other words, how many adjacent UDL loads to those of the design member at index $i = 4$ need to be considered to capture 99% of the true utilisation ratio $u_{d,\mathrm{true}}$. As the first step, one assumes an influence zone of $k_{\max} = 0$ as shown in Figure 4.4b); this ignores all UDL values except for those on the design beam d. Using Eurocode load factors and combinations [468], the captured utilisation for an influence zone $k_{\max} = 0$ is $u_{d,\mathrm{cap}} = 0.5101$. In the next step, shown in Figure 4.4c), the influence zone is increased to $k_{\max} = 1$. Note that the critical load arrangement now occurs when members $i = 3$ and $i = 4$ experience the full variable (live) load. The captured utilisation ratio in this instance is $u_{d,\mathrm{cap}} = 0.7191$. The influence zone value is then increased again to $k_{\max} = 2$ as shown in Figure 4.4d). Notice that the critical load arrangement now occurs when spans $i = 2, 4, 5$ are fully loaded, and yields a captured utilisation ratio of

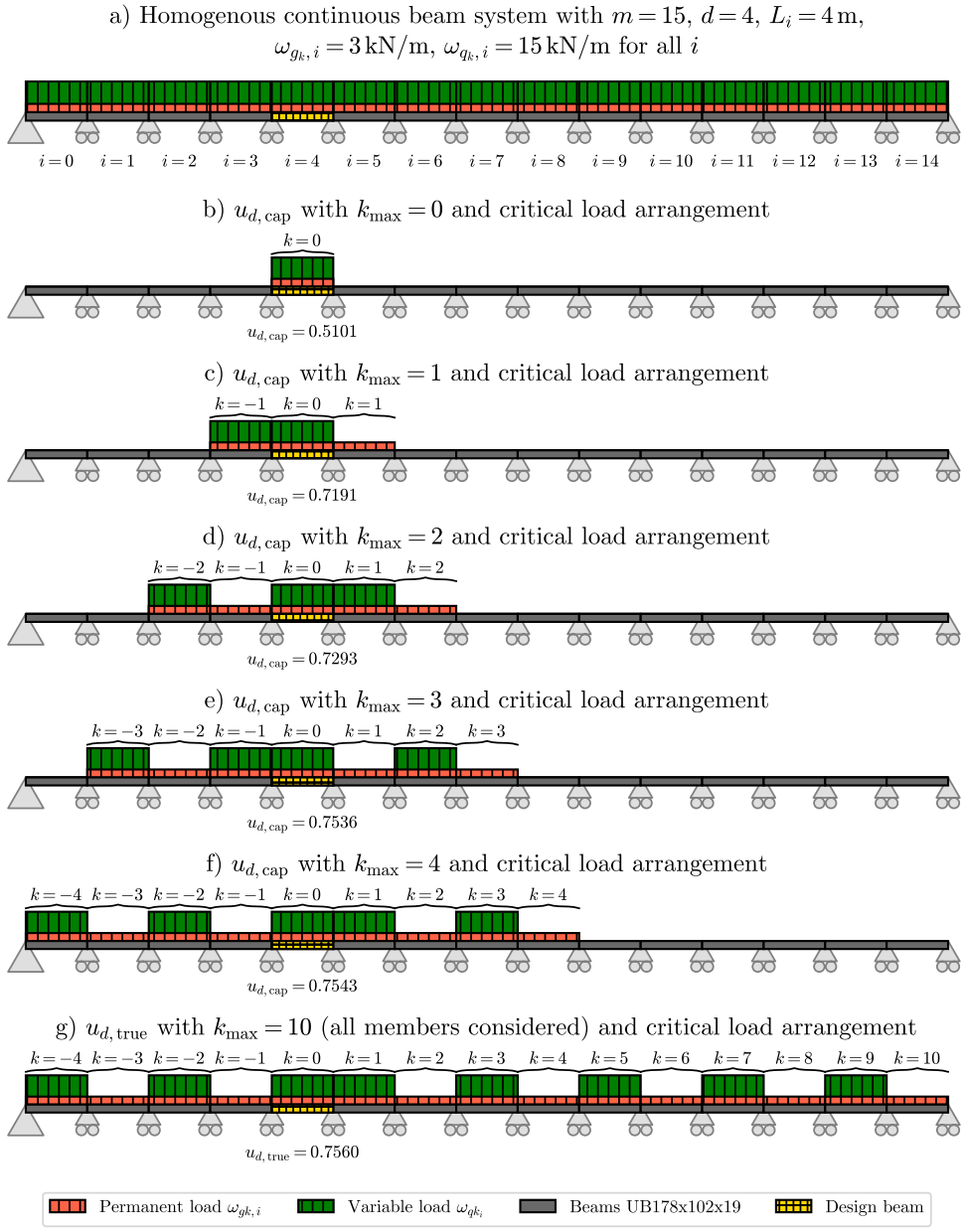


Figure 4.4: A visualised example of the influence zone evaluation procedure. The procedure begins with an existing beam system loaded as shown in a), and then evaluates the captured utilisation $u_{d,\mathrm{cap}}$ for various values of $k_{\max}$ in b) - f). The appropriate influence zone value becomes a function of $\epsilon_{\max}$ that ensures the captured utilisation ratio is sufficiently close to the true utilisation ratio evaluated in g).

$u_{d,\mathrm{cap}} = 0.7293$. By continuously increasing the assumed value of $k_{\max}$ and identifying the critical load arrangement for that particular assumed influence zone value, one yields the captured utilisation ratio $u_{d,\mathrm{cap}}$. This is precisely what is expressed by Equations 4.2 and

4.3.

The assumed influence zone value can be continuously increased to $k_{\max} = 3, 4$, etc., and their associated captured utilisation ratio $u_{d,\text{cap}}$ evaluated as shown in Figures 4.4e) and 4.4f). Eventually, the influence zone captures the entirety of continuous system, as shown by $k_{\max} = 10$ in 4.4g). In this instance, the utilisation ratio represents the true utilisation ratio $u_{d,\text{true}}$ of the design system, since it considers all UDL loads. In this instance, $u_{d,\text{true}} = 0.7560$.

Knowing the true utilisation ratio, it is now possible to evaluate which influence zone value $k_{\max}$ fulfilled the conditions identified by Equation 4.2. In this example, an influence zone value of $k_{\max} = 3$ corresponded with a captured utilisation ratio $u_{d,\text{cap}} = 0.7536$; since $\left| 1 - \frac{u_{d,\text{cap}}}{u_{d,\text{true}}} \right| = \left| 1 - \frac{0.7536}{0.7560} \right| = 0.0032 \leq \epsilon_{\max} = 0.01$, and because it can be shown this holds true for all larger values of $k_{\max}$, the influence zone value for design beam $d = 4$ with an error of less than $\epsilon_{\max} = 0.01$ is $k_{\max} = 3$.

The example above demonstrates the procedure to evaluate the influence zone value for one member within a particular continuous beam system containing a specific set of UDL $\boldsymbol{\omega}$ and span $\mathbf{L}$ values. By itself, this type of evaluation would already provide some utility to structural designers to understand, for example, if a new UDL load will significantly impact a beam design depending on whether it falls within that beam's influence zone.

However, the influence zone value is ultimately also a function of the *range* of UDLs, spans and cross-section properties that can arise. Consider for example that the influence zone of a continuous beam systems with UDLs of infinite magnitudes will always be the entire system size, regardless of the $\epsilon_{\max}$ value. For these reasons, it is of interest to study the statistical distribution of influence zone values for beam systems under pre-defined design constraints as defined by Equation 4.1. By systematically generating a multitude of different continuous beam systems, identifying the critical load arrangements, and evaluating the influence zone of each member within those systems, influence zone values can be studied in their aggregate. This can lead to prior knowledge of the size of the influence zone for members within a continuous beam system, which in turn could be of interest to practising structural engineers as explained in Section 4.1.

4.2.5 The key challenge

Most of the information required in Equations 4.2 and 4.3 can be easily found. Influence lines are relatively easy to extract for a continuous beam system, and integrating these for Equation 4.3 is numerically simple. The cross-section design checks $\mathbf{D}_{d,i,j}$ are clearly defined by design codes, and the summation of the individual utilisation ratio contributions in 4.2 is trivial. However, from the visualised example shown in Figure 4.4, it should be evident that accurate knowledge of the critical load arrangement for the entire system as well as for different sizes of the assumed influence zone value is required.

To find the critical load arrangements for Equations 4.2 and 4.3, one could use a naive, brute-force procedure to trial every possible load arrangement to create the set $\mathbf{J}_{\text{naive}}$ with a corresponding set size of $p_{naive} = 2^m$. This is not an issue for systems with few members, but if larger systems, such as the $m = 15$ system shown in Figure 4.4, need to be modelled to study the influence zone in depth, a brute-force approach becomes computationally expensive. For example, the $m = 15$ system has $p_{naive} = 2^m = 32,768$ possible load

arrangements, and only a smaller set of these can actually ever be critical. The issue of computational cost in relation to critical load arrangements of large-scale systems is well acknowledged in literature, and various methodologies have been employed using probability [469] and possibility theories [470, 471]. Among the latter, fuzzy sets using interval finite-element methods have been shown to be efficient and accurate [472, 473].

However, whilst these interval-based methods are effective at evaluating the bounds (the worst case force/moment value) due to the critical load arrangement, they do not in fact reveal what this load arrangement looks like in terms of the loading polarity of each span that relates to this worst case response. This is problematic for the evaluation of the influence zone, since Equation 4.3 relies on being able to identify this set $\mathbf{J}$ explicitly in terms of the individual activation factors $J_{i,j}$ (either 0 or 1) that make up the load arrangement J_j. Another approach would be to use the load arrangements prescribed by design manuals, yet these consist out of a heuristic set of load arrangements that are known to be non-conservative [473].

Due to these limitations, the methodology in Section 5.3 focuses on developing a systematic and universal procedure to evaluate the critical load arrangements $\mathbf{J}_{\text{crit}}$ of any continuous beam system of size m. The procedure will also focus on being applicable to both homogeneous (identical spans, UDLs and beam properties for all members) and heterogeneous systems. This knowledge will also feed directly to generating design datasets of continuous beam systems efficiently under various design constraints. Section 5.4 presents the results, first validating the critical load arrangement procedure established in this work, and subsequently finding the influence zone values for continuous beam systems under various design constraints.

4.3 Methodology

4.3.1 Assumptions and design constraints

This investigation will make the following design and modelling assumptions. First, cross-sectional properties are restricted to prismatic BS EN 10365:2017 UKB I-sections [467] made out of S355 steel with perfectly linear elastic behaviour using Timoshenko-Ehrenfest beam theory (yet the design was conducted using plastic section properties as allowed by EN 1993-1-1 5.4.2(2) [466]). It was assumed that all spans are laterally restrained (and hence not susceptible to lateral instability), with elements designed against EC3 ULS checks (and notably not SLS requirements) with EN 1990 Eq. 6.10 load combination factors [468].

The design constraints considered for Equation 4.1 were chosen for their relevance in the design of continuous steel framed buildings, and is reflected by the range of UDLs and spans of the design datasets. Four individual design scenarios are considered to study the influence zone in depth, with each set featuring an increasing variation in span lengths and applied loads, summarised in Table 4.1. Length and UDL values are discretized in 0.5 m and 5 kN/m increments respectively, and are drawn from a random uniform distribution, thereby providing an increasingly higher level of heterogeneity in Sets 2, 3 and 4.

Dataset	$\omega_{g_k,i} =$	$\omega_{q_k,i} \in$	$L_i \in$
Set 1 *Zero variation*	3.0 kN/m	a for all i, with $a \in$ [0 kN/m, 60 kN/m]	b for all i, with $b \in$ [1 m, 12 m]
Set 2 *Low variation*	3.0 kN/m	[20 kN/m, 40 kN/m]	[4 m, 8 m]
Set 3 *Medium variation*	3.0 kN/m	[10 kN/m, 50 kN/m]	[2 m, 10 m]
Set 4 *High variation*	3.0 kN/m	[0 kN/m, 60 kN/m]	[1 m, 12 m]

Table 4.1: Design constraints for various design scenarios used in this investigation based on Eurocode terminology, with G_k and Q_k being the characteristic permanent and variable actions. Increasing set numbers correspond to increasing design variation, a proxy for design complexity. Span and UDL values are discretized in 0.5 m and 5 kN/m increments respectively. Examples of span and load values of each set can be found in Figure 4.11.

4.3.2 Critical load arrangements

The critical load arrangement identification procedure relies on two new concepts, namely *polarity sequences* and *polarity zones*. Both of these concepts form the basis to systematically identify the critical load arrangement set $\mathbf{J}_{\text{crit}}$.

Polarity sequences

Influence lines can be used to identify the critical load arrangements for a given continuous beam system. By integrating the influence line of each member i, one can evaluate the net contribution (positive or negative) a UDL causes in terms of bending moments and shear forces at the influence line (IL) location. The net-contribution of each beam can be either positive or negative at the IL location, that is "hogging or sagging" for bending moments and "clockwise or anti-clockwise" for shear forces, respectively, which is termed as the *polarity* of that particular beam. This procedure is shown in Figure 4.5a) to b).

Since the design problem is restricted to positive UDL values only (see Table 4.1), it is possible to construct a *polarity sequence* for a particular IL location, as shown in Figure 4.5c). When all beams of positive polarity are loaded, then the maximum positive internal forces are generated at the IL location, and vice-versa, loading the negative polarity members leads to the maximum negative internal forces.

Polarity zones

A rigorous qualitative study of the polarity sequences for different IL locations and design scenarios revealed 5 unique polarity sequences that occur along specific segments of a given beam span termed *polarity zones*, which are illustrated in Figure 4.6 for the central beam highlighted in red. For example, the polarity sequences shown in the last row of Figure 4.5, which applies at the indicated IL location shown by the green triangle,

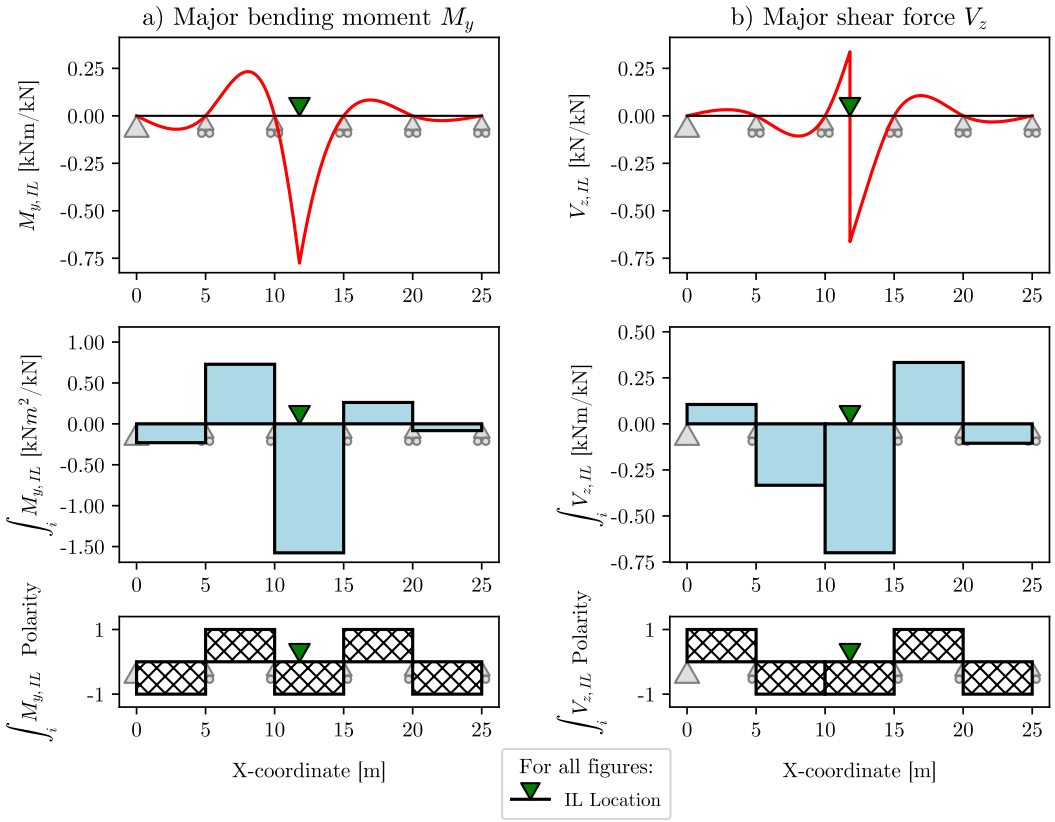


Figure 4.5: An exemplary process of arriving from influence line plots (top row) to polarity sequences (bottom row) via integrated influence lines (middle row) for a) major axis bending moment M_y and b) major axis shear force V_z about the specified influence line (IL) location.

corresponds to the Type III sequence shown in Figure 4.6e) for major axis bending, and Type IV sequence shown in Figure 4.6f) for major axis shear.

These 5 polarity zones are common to all members of both homogeneous (equal spans and cross-sections) as well as heterogeneous continuous beam systems, although the exact boundaries between one zone varied depending on the relative magnitude of spans and cross-section properties. The sequences identified in Figure 4.6 also extend to larger beam systems with the polarity direction alternating at each successive beam. For example, if the 5-member system was extended by an additional member on either side of the system (to give a 7 member system), the left-most member of the Type I polarity sequence would have a positive polarity, and similarly, the right-most member would have a negative polarity. The same logic extends to the other four sequences.

Each polarity sequence is indicative of two critical load arrangements that maximise the positive or negative internal member forces respectively. The maximum positive load arrangement for Type I is also equal to the maximum negative load arrangement for Type IV, since these sequences are polar opposites of each other, which is also true for the Type II and Type V sequences. Consequently, these 5 polarity zones correspond to 6 unique

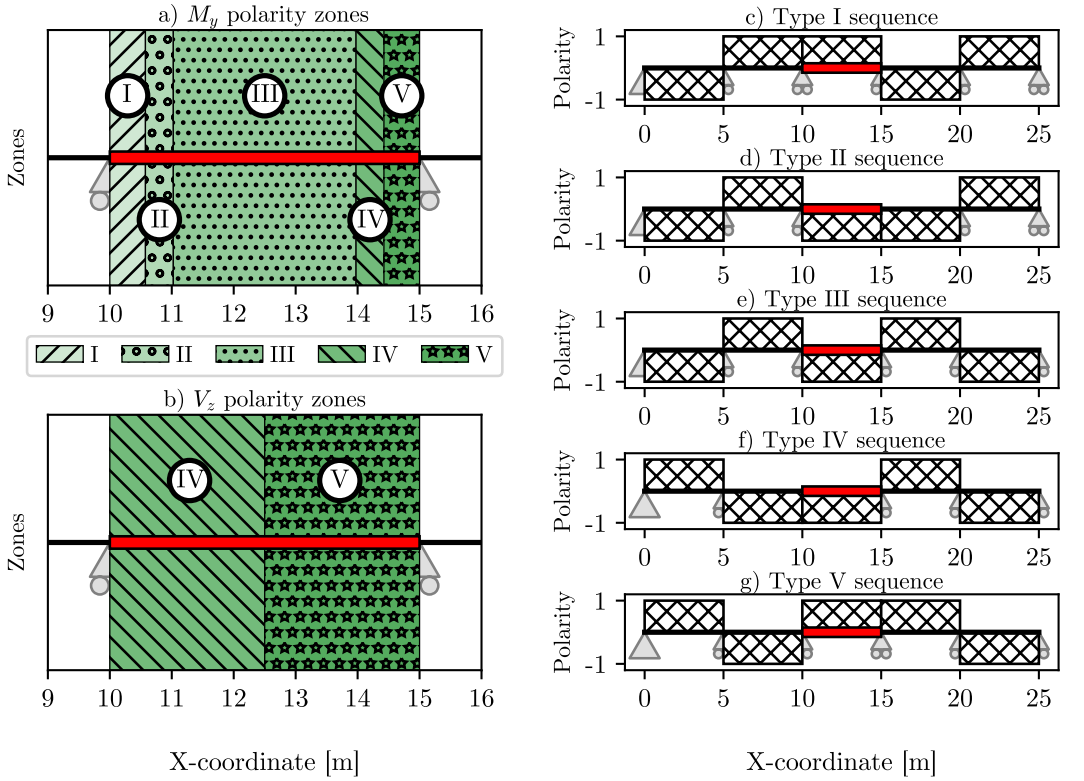


Figure 4.6: Polarity zones that occur along various span segments of a $m = 5$ homogeneous beam system of equal span and cross-sectional properties. The same zones and sequences, although at different boundaries, occur in heterogeneous (varying UDL and span) systems.

critical load arrangements for a given beam, namely positive Type I, II and III along with their (negative) polar opposites. The only exceptions occur for the beams at either end of the spans, named end-span beams, in which the Type I and Type IV sequences collapse into the Type III sequence (or its polar opposite) at the left end, and similarly for the Type II and Type V sequences at the right end, resulting in four unique load arrangements for end-span beams.

Flexural load arrangements

It is now possible to identify the first set of critical load arrangements for continuous beam system, defined as the flexural load arrangements $\mathbf{J}_{\text{flex}}$. Although each non-end-span beam has 6 unique critical load arrangements, it does not mean that a continuous beam system has $6m$ unique load arrangements (m is the number of members in the beam system). This is because, as shown in Figure 4.7, the maximum positive Type V load arrangement for one beam is identical to the maximum positive Type I load arrangement of the beam immediately adjacent to (the right of) it. A similar overlap exists between Type II and Type IV sequences, and the two Type III load arrangements (for maximum

and negative internal forces) are identical for all beams.

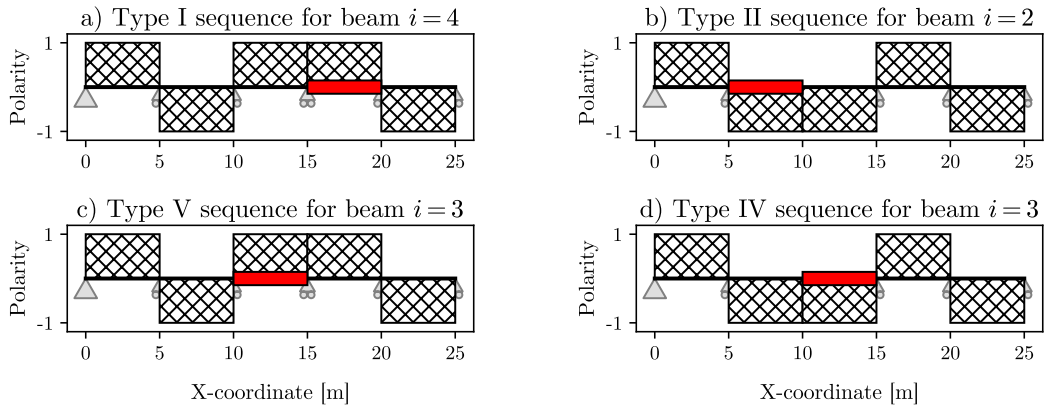


Figure 4.7: Polarity sequences are identical for adjacently lying beams (highlighted in red) for Type I and Type V sequences as shown by Figure a) and Figure c), as well as Type II and Type IV sequences, as shown by Figure b) and Figure d).

Through a process of elimination, it is possible to simplify the actual total number of potential critical load arrangements to $p_{\text{flex}} = 2m$. Algorithm 4.1 provided in 4.A can be used to evaluated $\mathbf{J}_{\text{flex}}$. An example output for a $m = 5$ system is shown in Figure 4.8, highlighting the $p_{\text{flex}} = 2m = 10$ critical load arrangements J_j, along with their individual activation factors $J_{i,j}$. The load arrangement set $\mathbf{J}_{\text{flex}}$ of size $p_{\text{flex}} = 2m$ identified here represents a literal exponential improvement in terms of computational cost when compared to the brute-force approach of analysing and designing against $p = 2^m$ load arrangements and for evaluating the influence zone with Equation 4.2.

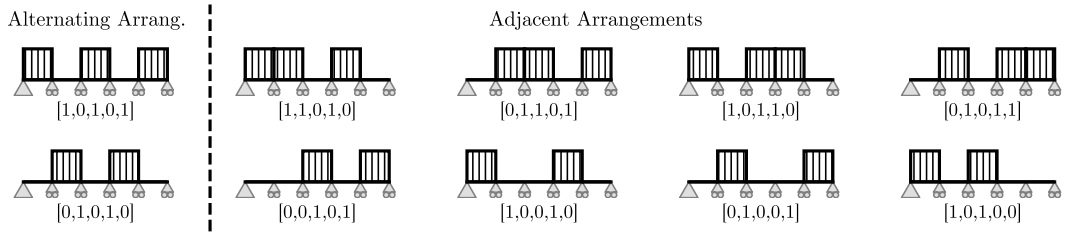


Figure 4.8: The critical load arrangements set $\mathbf{J}_{\text{flex}}$ of size $p = 2m$ for a 5-member continuous beam system ($p = 10$) grouped in alternating and adjacently loaded arrangements. Each load arrangement J_j consists out of $J_{i,j}$ activation factors of value 0 or 1.

Shear load arrangements

During initial validation of whether $\mathbf{J}_{\text{flex}}$ contained all the critical load arrangements $\mathbf{J}_{\text{crit}}$ (i.e. $\mathbf{J}_{\text{crit}} \in \mathbf{J}_{\text{flex}}$), other unique critical load arrangements were identified. Analysing these special cases in detail indicated that these different critical load arrangements occurred when the span of a member was less than a certain L_{shear} span limit quantified by:

$$L_i < \sqrt{\frac{6EI_{yy,i}}{GA_{z,i}}} \approx L_{\text{shear}} \tag{4.4}$$

where E and G are the Young's and shear modulus of the material respectively, L_i is the span of the beam, and $I_{yy,i}$ and $A_{z,i}$ were the major second moment of area and shear area of the prismatic beam, respectively.

Although the L_{shear} span limit appears to be related to shear beams, this is the first time that shear beams have been reported in literature to cause novel critical load arrangements. The shear limit identified by Equation 4.4 was derived by studying the differences between the more commonly used Euler-Bernoulli beam theory with the Timoshenko-Ehrenfest theory [474], specifically as expressed in their stiffness matrix form. As shown in Figure 4.9, shear beams appear to flip the polarity of the immediately adjacent member when measured outwardly from a given IL location , with all subsequent members alternating the polarity direction as before.

When shear beams (as defined by the L_{shear} limit) occur, they introduce new critical load arrangements not found within $\mathbf{J}_{\text{flex}}$. The increase in terms of the final utilisation factor of the beams was typically in the range of 4-5%, although larger increases were also observed. Whilst a thorough analysis of the increase in utilisation ratio caused by these newly identified load arrangements would be of interest, it falls outside the scope of this study. The validity of Equation 4.4 for all design conditions, especially for different cross-sections, as well as the physical cause for the unique influence line patterns shown in Figure 4.9b) would require further investigations.

Instead, an algorithm will be presented capable of identifying these new load arrangements, which is necessary for the evaluation of the influence zone. The principal issue when evaluating the shear beam induced critical load arrangements, hereafter referred to as the *shear load arrangements* $\mathbf{J}_{\text{shear}}$, is the fact that the final material and cross-sectional properties to evaluate the L_{shear} limit in Equation 4.4 are not known until the beam is designed. This creates a causality dilemma which needs to be addressed.

In clear opposition to the $\mathbf{J}_{\text{flex}}$ set, which does not depend on the continuous beam system properties, the shear load arrangements cannot be established *in universum* without some system knowledge. However, by taking advantage of the design constraints set by Equation 4.1, which are defined in Table 4.1, one can identify *a priori* what members are susceptible to cause shear load arrangements by re-writing Equation 4.4 as:

$$\sqrt{6\left(\frac{E}{G}\right)_{\max}\left(\frac{I_{yy}}{A_z}\right)_{\max}} < L_{\text{shear,max}} \tag{4.5}$$

The above equation groups the maximum material and cross-sectional property ratios together. By limiting the design space to S355 steel and UB section sizes as specified in Section 4.3.1, the maximum material ratio ($(E/G)_{\max} = 2.600$ (using EN 1993-1-1 material properties [466]) and cross-sectional property ratio ($(I_{yy}/A_z)_{\max} = 0.397m^2$) can be evaluated (based on BS EN 10365 [467] cross-sections). Beams shorter than the shear span limit are susceptible to cause shear load arrangements (in this case $L_{\text{shear,max}} =$ 2.49m). In identifying these susceptible members, it is possible to evaluate the shear load arrangements using Algorithm 2 provided in 4.B.

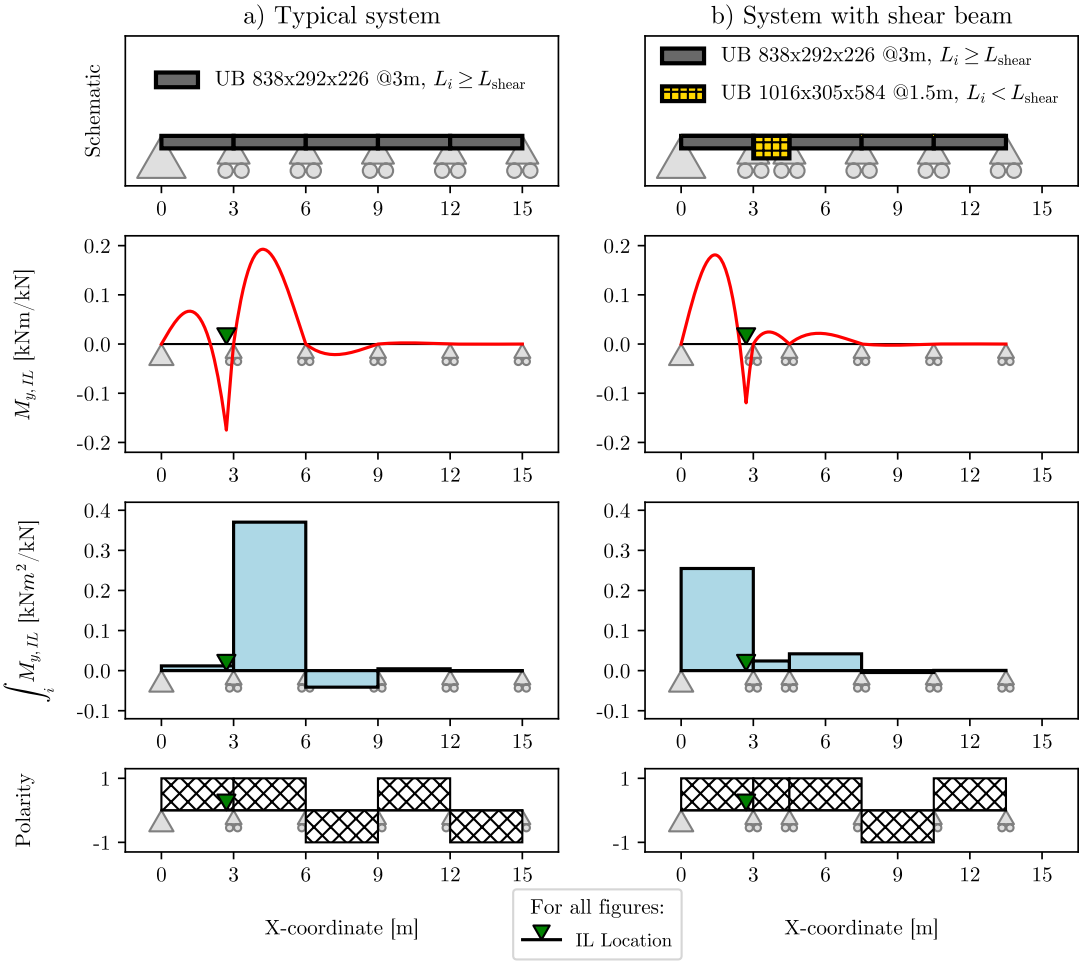


Figure 4.9: A schematic demonstrating the impact of a shear beam (highlighted in yellow) on a standard polarity sequence of a continuous beam system when spans shorter than the shear span limit L_{shear} (as identified by Equation 4.4) occur. A deeper beam is used on the second span to increase the effects of this influence line pattern. Note the flipped polarity directions of the members on the right-hand side of the system.

Algorithm 2 transforms the flexural load arrangement from set $\mathbf{J}_{\text{flex}}$ based on a list of susceptible shear beams identified by Equation 4.5. This is achieved by flipping the on/off activation factor (the 0 or 1 in $J_{i,j}$) of the load arrangement if a shear beam is encountered whilst travelling outwardly in both the left and right direction from a start beam index. This operation transforms the flexural load arrangement based on the behaviour identified visually in Figure 4.9, and needs to check four individual case conditions to account for continuous beam systems that have multiple, potentially adjacently lying, shear beams. A demonstrator file of how both Algorithm 1 and 2 are applied can be found within the referenced data repository [475].

Since every beam system is of size m, the time complexity of a single pass of Algorithm 4.2 is $O(m)$. However, since every flexural load arrangement $(2m)$, and every combination

of n potential shear beams ($2^n - 1$ combinations, as the zero set is already considered in $\mathbf{J}_{\text{flex}}$ by default), and every possible start-index (m) needs to be computed, the time complexity to evaluate the shear set $\mathbf{J}_{\text{shear}}$ would be $O(m^3\, 2^n)$. It should be noted that this process is computationally expensive.

It was observed that passing every possible start index generated either duplicate shear load arrangements, or occasionally existing flexural load arrangements. For example, for a given singular potential shear beam location, the algorithm would result in the same transformed shear load arrangement for all start-indices starting on the left and right hand-side of that susceptible shear beam location. Similarly, the two alternating arrangements from $\mathbf{J}_{\text{flex}}$ would result in an already existing adjacent arrangement from $\mathbf{J}_{\text{flex}}$ if only a singular susceptible shear beam exists.

Using such logic, it is sufficient to pass only adjacent arrangements from $\mathbf{J}_{\text{flex}}$ along with the left-hand (or right-hand) index of the adjacently loaded spans as the start index for Algorithm 4.2 to yield an effective set of potential shear load arrangements. By not having to evaluate Algorithm 4.2 for every possible start index of each load arrangement, the computational complexity reduces to $O(m^2\, 2^n)$. From this, it also follows that since the alternating load arrangement is never transformed (which leaves only $2(m-1)$ load arrangements to be passed to the algorithm) and since $2^n - 1$ possible shear beam combinations can exist, the maximum number of unique critical shear load arrangements should be of size $p_{\text{shear}} = 2(m-1)(2^n - 1)$.

Validation test of critical load arrangements for continuous beams

By adding the set of flexural and shear load arrangements together, it should be possible to explicitly define the set of critical load arrangements for any continuous beam system under defined design constraints. A validation test was conducted to check if $\mathbf{J} \rightarrow \mathbf{J}_{\text{crit}} \in \mathbf{J}_{\text{flex}} \cup \mathbf{J}_{\text{shear}}$, with the results presented in Section 5.4, with results specifically shown in Figure 4.10. The validation test was achieved by generating 1024 heterogeneous continuous beam systems with $m = 10$ members with spans $L_i \in [1\,\text{m},\, 12\,\text{m}]$, permanent (dead) UDL $\omega_{g_k} = 3\,\text{kN/m}$ and variable (live) UDLs $\omega_{q_k,i} \in [200\,\text{kN/m}, 400\,\text{kN/m}]$, all drawn from uniform distributions for each member i to model high design complexity. The higher UDLs $\omega_{q_k,i}$ increase the likelihood of deep beams occurring, which increases the chance of critical shear load arrangements, allowing the performance of both algorithms to be stress-tested. Each of the 1024 continuous beam systems were designed against the naive set of load arrangement J_{naive} of size $p_{naive} = 2^m = 1024$, with the most critical load arrangement of each member within each continuous beam system identified. This resulted in $10 \times 1024 = 10,240$ validation data-points. It was subsequently checked if the actual critical load arrangement was represented within the significantly smaller $\mathbf{J}_{\text{flex}} \cup \mathbf{J}_{\text{shear}}$ sets.

4.3.3 Design dataset generation

After validating the critical load arrangement procedure, design datasets were generated under the various design constraints established in Section 4.3.1. One key consideration is the size of the continuous beam systems in terms of m to model. The number of members m needs to be at least double the maximum influence size $k_{\max}$. This is because the highest influence zone measurable for the middle span of a continuous beam

is by design half the system length m. Therefore, size m needs to be chosen such that $\max(\mathbf{k}_{\max}) < m/2$, where $\mathbf{k}_{\max}$ is the list of all influence values $k_{\max}$ of the continuous beam system. A sufficiently large value for m needs to be assumed in case $k_{\max}$ is a large value; $m = 15$ was used for this purpose.

Individual design datasets consisting of 32 UDL and 32 span values sampled from a random uniform distribution for a $m = 15$ beam system were created based on the design constraints identified in Section 4.3.1. Sets 2, 3, and 4, each contained $32 \times 32 = 1024$ continuous beam systems, and since each system contained $m = 15$ members, this resulted in a total of $1024 \times 15 = 15,360$ influence zone values. For Set 1, the difference within the systems only varied in terms of the identical span L and UDLs $\omega_{q_k,i}$ of the members, which were also sampled in 0.5 m and 5 kN/m increments respectively. Given that this results in 23 span and 13 UDL increments for Set 1, Set 1 contained $23 \times 13 \times 15 = 4485$ influence zone values.

For the generation of the continuous beam design datasets, a coupled analysis and design approach was taken, optimising for minimum structural depth using third-party software (Rhino3D©, Grasshopper© and Karamba3D© [476]). Design sensitivity analysis was avoided by an implicit ordering of the UKB section list based on structural capacity. The influence zone values were subsequently extracted using Equations 4.2 and 4.3 based on the procedure visualised in Figure 4.4.

4.4 Results

4.4.1 Validation of flexural and shear load arrangement algorithms

The validation results for the critical load arrangement identification procedure are illustrated in Figure 4.10, which plots the critical load arrangement index for each design beam example. Every load arrangement index corresponds to a unique load arrangement out of the naive set $\mathbf{J}_{\text{naive}}$ of size $p_{\text{naive}} = 2^m = 1024$. The set $\mathbf{J}_{\text{naive}}$ was ordered so that the load arrangements for set $\mathbf{J}_{\text{flex}}$ are first, followed by those of set $\mathbf{J}_{\text{shear}}$, and subsequently all others. The design examples themselves were sorted twice: first in ascending number of shear beam occurrences, and subsequently in ascending load arrangement indices. This results in the gradual increase of the J_{crit} indices as seen in Figure 4.10.

Figure 4.10 sheds insight on a number of important points. The first is that the critical load arrangement J_{crit} for every single beam example from the 10,240 validation dataset occurred within the $\mathbf{J}_{\text{flex}}$ or $\mathbf{J}_{\text{shear}}$ sets. This is strong evidence that the critical load arrangement identification procedure based on polarity sequences zones and polarity zones, as well as Algorithms 4.1 and 4.2, are capable of identifying the critical load arrangement of homo- and heterogeneous continuous beam systems. Furthermore, the set size predictions $p_{flex} = 2m$ and $p_{shear} = 2(m-1)(2^n - 1)$ are also confirmed. For the $m = 10$ member system designed here, $p_{flex} = 20$, and depending on the number of shear beam occurrences of each system, which varied from $n = \{0, 1, 2, 3, 4\}$, the number of shear load arrangements varied from $p_{shear} = \{0, 18, 54, 126, 270\}$. This corresponded to $p_{total} = \{20, 38, 74, 146, 290\}$ respectively, as indicated by the y-axis of Figure 4.10 b).

Figure 4.10a) also emphasises how much smaller sets $\mathbf{J}_{\text{flex}}$ and $\mathbf{J}_{\text{shear}}$ are in comparison to $\mathbf{J}_{\text{naive}}$. This greatly reduces the number of load arrangements that need to be analysed

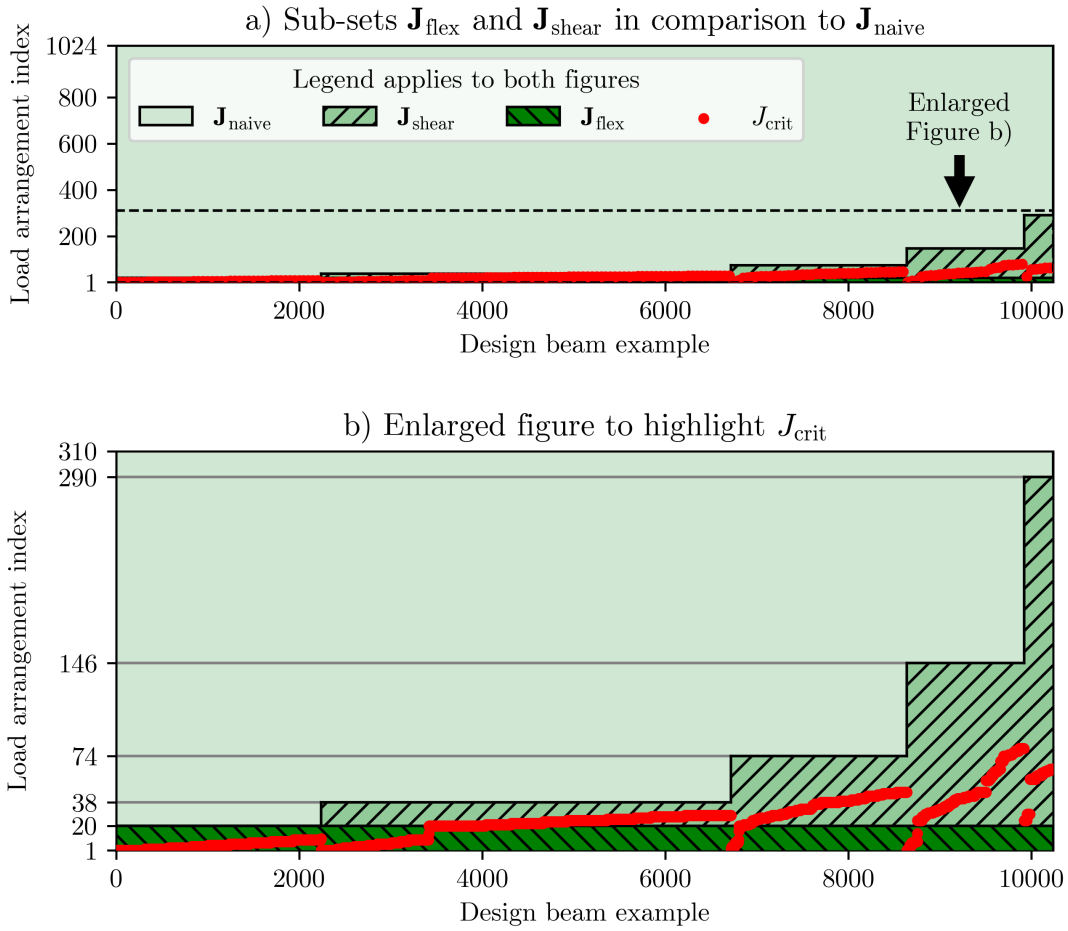


Figure 4.10: Load arrangement index for each design beam example ordered in increasing number of shear beam occurrences and critical load arrangement indices. This confirms visually that the critical load arrangement J_{crit} for each design beam example from the generated dataset falls within either $\mathbf{J}_{\text{flex}}$ or $\mathbf{J}_{\text{shear}}$ and are significantly smaller than $\mathbf{J}_{\text{naive}}$. Figure b) is an enlarged view of Figure a).

for the influence zone evaluation procedure, whilst also reducing the computational cost of designing the continuous beam system for system lengths of $m > 10$. A summary of these results is shown in in Table 4.2, and further insights are discussed in Section 4.5.

4.4.2 Influence zone results

Using the validated critical load arrangement evaluation procedure presented in Section 4.3.2, various continuous beam systems were generated based on the design constraints highlighted in Section 4.3.1 and 4.3.3. Subsequently, using Equations 4.2 and 4.3 and the influence zone evaluation procedure visualised in Section 4.2.4, the influence zone values k_{max} for every member from each system could be evaluated. The influence zone values for one random example from each of the designed continuous beam systems within each

Set	Set Size	Algorithm Complexity
Critical load arrangements per internal beam	6	$O(1)$
Critical load arrangements per end-span beam	4	$O(1)$
$\mathbf{J}_{\text{flex}}$ - Critical flexural arrangements per beam system	$2m$	$O(m)$
$\mathbf{J}_{\text{shear}}$ - Critical shear arrangements per beam system	$2(m-1)(2^n-1)$	$O(m^2\,2^n)$
$\mathbf{J}_{\text{naive}}$ - Naive load arrangements	2^m	$O(2^m)$

Table 4.2: Load arrangements set summary for m dimensional beam systems containing n shear beams with associated algorithm complexities.

dataset (Set 1: Zero Variation, Set 2: Low Variation, Set 3: Medium Variation and Set 4: High Variation, see Table 4.1) for a maximum error threshold $\epsilon_{\max} = 0.005$ are shown in Figure 4.11.

Within the examples presented in Figure 4.11, the influence zone $k_{\max}$ can vary for individual members within the same system. For example, in Figure 4.11c), the first member from the left end of the system with a span of $L = 9.5\,\text{m}$ and a variable (live) UDL value of $\omega_{q_k} = 45\,\text{kN/m}$ has an influence zone value of $k_{\max} = 1$. Within the same system, the fourth member from the right end of the system with a span of $L = 3.5\,\text{m}$ and $\omega_{q_k} = 20\,\text{kN/m}$ had an influence zone value of $k_{\max} = 5$. Another general observation is that the variability of the influence zone values appears to correlate positively with design complexity (the variability of the spans and UDLs of the system); this can be seen comparing the $k_{\max}$ values in Set 1 as shown in Figure 4.11a) with those of Set 4 as shown in Figure 4.11d).

The influence zone values of members within continuous beam systems that have already been designed (such as those shown in Figure 4.11) shed some insight for structural engineers on whether a design change in terms of a UDL is relevant for a given member (based on whether it falls within that member's influence zone). However, by analysing the distribution of influence zone values of many different homo- and heterogeneous continuous beam systems, it is possible to make that assessment *a priori* before designing the system itself. Assuming once again a max error threshold $\epsilon_{\max} = 0.005$, the aggregated influence zone results for each design set from Table 4.1 are shown in Figure 4.12. For these design datasets, the most common influence zone value (the mode) was $k_{\max} = 3$, and the majority of influence zone values were at $k_{\max} \leq 3$, meaning applied loading information of a given beam along with that of the three adjacent spans on either side captured the correct utilisation ratio of the design beam with an error of less than a $\pm 0.5\%$ in the majority of cases.

However, the various sets reveal differences in the maximum and distribution of the influence zone. The maximum influence zone value for Set 1 was $k_{\max} = 4$, whereas it was $k_{\max} = 5$ for Set 2, 3 and 4. Furthermore, as the set number increases, which corresponds with an increase in variation of the design information in terms of spans and UDLs, the

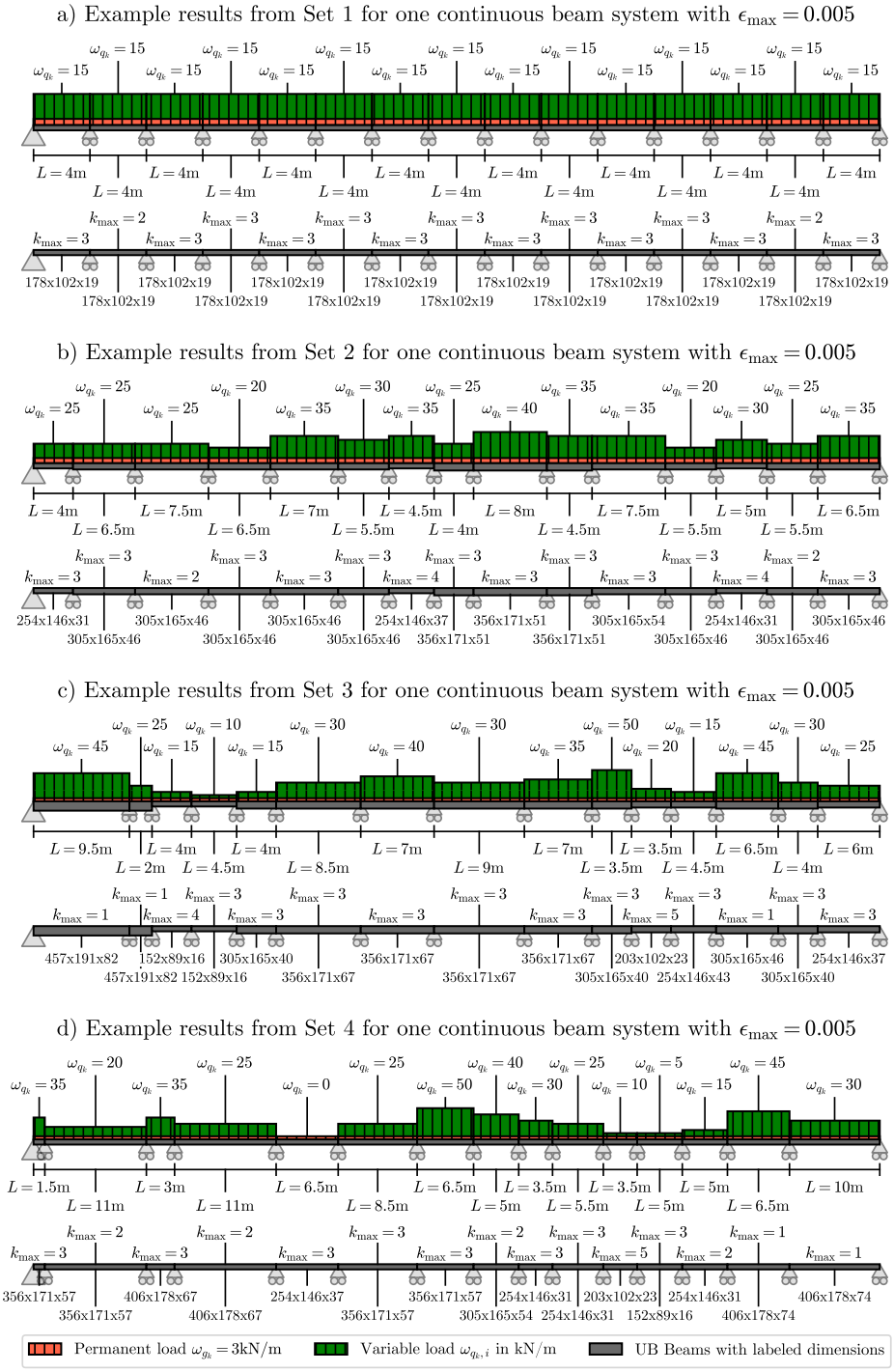


Figure 4.11: One example of the influence zone results within continuous beams systems from each design dataset specified in Table 4.1.

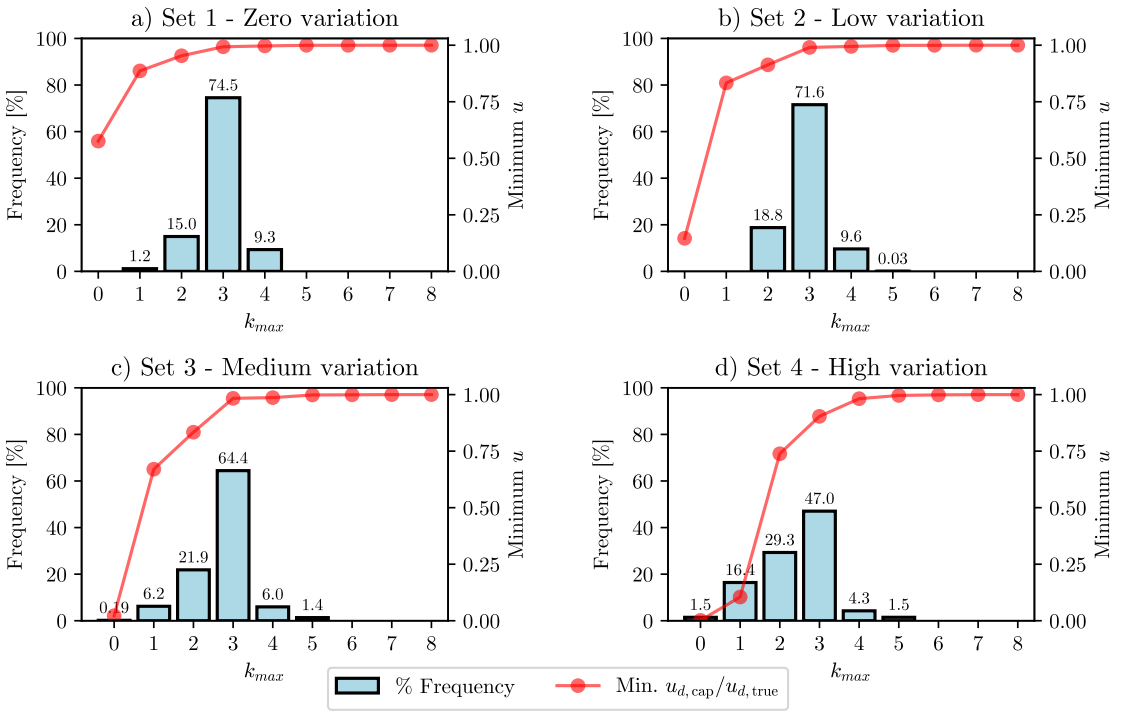


Figure 4.12: Influence zone results for various design constraints with a max error threshold $\epsilon_{\max} = 0.005$ indicating the percentage frequency distributions of the influence zone values $k_{\max}$ and minimum utilisation factors captured for each $k_{\max}$ value for a given design beam d.

influence zone value distribution appears to flatten and widen. For example, it was the high-variation Set 4 which actually contained the most influence zone values $k_{\max} = 0$ for 1.6% of the design examples, whereas the zero variation Set 1 only had 0.4% of its design examples exhibit an influence zone of $k_{\max} = 0$. The minimum utilisation curve (red curve with point markers in Figure 4.12) captured by each influence zone value suggests that, in general, increasing design variation leads to greater maximum influence zone values.

The average, maximum and 95th percentile influence zone values were also calculated for various error thresholds as shown in Table 4.3. Note that the maximum influence zone value of $k_{\max} = 5$ for Set 4 confirms that the $m = 15$ member-size assumption was sufficient for the purpose of this study. Together Figure 4.12 and Table 4.3 provide evidence for the following conclusions:

- A decrease in the acceptable error threshold correlates with an increase in both the average and maximum influence zone range.
- An increase in design variation correlates with an increase in the maximum influence zone range.
- An increase in design variation, however, correlates with a decrease in average influ-

ence zone range in most instances where the acceptable error threshold is relatively tight ($\epsilon_{\max} \leq 10\%$). At higher error thresholds the trend is less discernible.

Error $\epsilon_{\max}$ [%]	Average $k_{\max}$				Maximum $k_{\max}$				95th percentile $k_{\max}$			
	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
0.1	4.60	4.46	3.84	3.37	5	5	6	7	5	5	5	5
0.5	2.89	2.86	2.69	2.38	4	5	5	5	4	4	4	4
1	2.76	2.75	2.35	2.06	3	3	5	5	3	3	3	3
5	1.52	1.39	1.29	1.17	2	3	3	4	2	3	3	3
10	0.98	0.98	0.89	0.83	2	2	3	4	1	1	2	2
20	0.76	0.84	0.73	0.67	1	1	2	3	1	1	1	1
50	0.00	0.30	0.41	0.43	0	1	1	2	0	1	1	1

Table 4.3: Influence zone results for various maximum error thresholds $\epsilon_{\max}$ for each design dataset, evaluating average, maximum and 95th percentile influence zone values $k_{\max}$. Note that increasing set numbers corresponds with increasing design variation, a proxy for design complexity, see Table 4.1 for details.

It should be noted that an error threshold of less than 0.5% is relatively small in comparison to uncertainties that exist in structural design. These uncertainties include, for example, material yield strength and imposed UDL values (consider that variable UDL values ω_{q_k} are increased 50% with a load combination factor of 1.5 within the Eurocodes [468]). Furthermore, the design constraints of design set 4 represent the top end of design variation which may occur in typical continuous beam systems. It is therefore reasonable to suggest that for continuous beam systems with design constraints specified in section 4.3.1, the influence zone values are on average $k_{\max} \leq 3$, with the 95th percentile value being $k_{\max} = 4$ and only in the most extreme case $k_{\max} = 5$ at an error threshold of less than 0.5%.

4.5 Discussion

The results along with the proposed evaluation procedures to find the critical load arrangements and influence values for UDL loaded continuous beam systems have led to a number of important findings. These include gaining novel insight on how much surrounding loading information is relevant for a member's design, identifying novel *shear load arrangements* with the help of *polarity zones* and *polarity sequences*, and introducing *load arrangement algorithms* to explicitly identify critical load arrangements of continuous beam systems of any arbitrary system size. Each of these findings are discussed in detail and contextualised with relevant existing literature.

4.5.1 Influence zone insights

The influence zone results confirm that the impact of loading drops off sharply the further away one moves from the influence line location. This behaviour can be identified across all influence line diagrams found within this paper, such as Figures 4.5 and 4.9. This investigation has formulated this concept as the *influence zone*, shown how it applied to continuous beam systems, and rigorously studied the influence zone distributions under various design assumptions and error thresholds. The data related to the dataset

generation and influence zone evaluations can be found at an open-source data repository [475].

4.5.2 Demarcating influence zones from influence lines

Although there is a proximal relationship between the concept of influence zones and influence lines, mostly evidenced by Equation 4.3 where integrated influence lines play an important role for the evaluation of influence zones, these two concepts differentiate themselves in important ways. This distinction also applies to the two-dimensional application of influence lines known as influence surfaces [477–480].

Whilst influence lines/surfaces are exact analytical tools that define the mechanical response of a known structural system about a particular point, influence zones are a heuristic design tool that offer insight on what information is relevant to the design of the structural system to begin with based on certain analytical assumptions. The value of influence lines/surfaces arise during analysis on a system-by-system basis, whereas the value of influence zones arise during design after having studied them in their statistical aggregate.

This distinction could be considered further evidence supporting the demarcation between design and analysis in structural engineering. Previous literature has highlighted the difference between *knowledge-that* explains fundamental facts about systems (such as influence lines) versus *knowledge-how* something can be designed or solved (such as influence zones) [4, 436]. Recent literature has suggested that the processes of analysis and design solve related, albeit oppositely posed problems known as forward and inverse problems respectively [140]. Influence lines can be seen as a tool that solves the former, whereas influence zones solve the latter.

As a matter of fact, the influence zone concept was developed whilst developing a design model for continuous beam systems from an inverse problem perspective (see Chapter 5), and allows the *a priori* knowledge of what loading information is relevant for design of a particular continuous beam. It is possible that the influence zone concept could serve as an important heuristic tool in the design of continuous structural systems, supporting the view that the application of heuristics is a cornerstone for engineering design [11]. Further novel ideas might be uncovered when approaching engineering design from an inverse problem perspective.

4.5.3 Flexural load arrangements

An important contribution of this investigation was presenting the flexural load arrangements clearly through the use of *polarity sequences*. Notably the *polarity zones* highlight which load arrangement is critical for specific segments of a beam, which could be useful in the design of tapered (non-prismatic) continuous beam systems [481, 482].

The influence zone study allows the contextualisation of simplified load arrangement provisions. For example, whilst Annex AB.2 from EN 1993-1-1 [466] covers alternating flexural load arrangements in full, it specifies that for the adjacent flexural load arrangement type, only the two adjacently loaded spans of variable load ω_{q_k} need to be factored. In essence, the variable load information on all other spans aside from the beam under consideration and the two directly adjacent spans are ignored, which is the technical

equivalent of assuming an influence zone to $k_{\max} = 1$.

With help of Table 4.3, it is possible to infer that an influence zone value $k_{\max} = 1$ is likely to introduce an error between $5 - 10\%$ in terms of the true utilisation for design scenarios with no UDL or span variation (the average $k_{\max}$ value for $\epsilon_{\max} = 5\%$ and $\epsilon_{\max} = 10\%$ is 1.52 and 0.98 for Set 1 respectively). The simplified Eurocode provisions are therefore, on average, a reasonable simplification to capture the impact of variable load arrangements. However, the maximum influence zone value of Set 1 with $k_{\max} = 1$ corresponds to an error of $\epsilon_{\max} = 20\%$, and when considering non-heterogeneous continuous beam systems (reflected by Set 2, 3 and 4), this error can increase up to $\epsilon_{\max} = 50\%$ and more. This is further evidence, as already pointed out in literature, that the load arrangement provisions from building codes can be non-conservative and hence lead to unsafe designs [473].

The simplified provisions within the Eurocodes, which also exist within EN 1992-1-1 5.1.3 [483] and other codes [484], need to be understood in context of the $1.5Q_k$ load factors and the dead load contribution G_k, which invariably will lessen the underestimation made by the provisions. Nonetheless, the validity of the design code recommendations for flexural load arrangements could be investigated further, especially for highly irregular beam and floor arrangements [485].

4.5.4 Shear load arrangements

Unlike flexural load arrangements, which have been identified in literature and building codes, the shear load arrangements were a surprising find. To the authors' knowledge, this is the first time that deep beams have been identified to cause new critical load arrangements in literature. Although shear load arrangements sometimes resulted in identical utilisation ratios to that of flexural ones, initial analyses pointed to an average increase in utilisation ratio of 4-5%, while larger deviations were occasionally observed. Figure 4.10 also highlights that these shear load arrangements were relatively prevalent within the design scenarios considered.

Confirmation and validation of these shear load arrangements by future research is encouraged. Of particular interest is why Equation 4.5 defines the exact point when these critical load arrangements arise, and whether this equation is valid for all material and cross-section types. One notable difference in the mechanical assumption in this investigation of load arrangements as to that of previous studies was the use of Timoshenko-Ehrenfest rather than Euler-Bernoulli beam theory. For example, the two seminal works on establishing the bounds of critical load arrangements using fuzzy set based finite-element methods used Euler-Bernoulli beam theory [472, 473]. A re-investigation with deep beams as defined by Equation 4.4 and Timoshenko-Ehrenfest beam theory should reveal more critical bounds of load arrangements than previously identified with interval-finite-element methods. The extent to which these shear load arrangements require special provisions within building codes will require further exploration.

4.5.5 Critical load arrangement algorithms

The critical *load arrangement algorithms* provided in 4.A and 4.B, along with a study of their computational complexity, were key for the evaluation of the influence zone. Limit-

ing the design space to a fraction of the naive J_{naive} load arrangement set without making heuristic simplifications was crucial in both the dataset generation and influence zone evaluation steps. The algorithms have been made available at an open-source data repository [475].

It is likely that there is further room for improvement for Algorithm 2 to evaluate the shear load arrangements for a known list of susceptible shear beams. The current formulation still leads to either pre-existing flexural load arrangements, or creates duplicate shear load arrangements. On average, 74.7% of the outputs obtained from Algorithm 4.B were unique, with a best-case efficiency of 88.8% and a worst-case efficiency of 12.7%. This suggests that an algorithm with a lesser computational complexity than $O(m^2\,2^n)$ might be achievable through further investigation.

4.5.6 Further influence zone investigations

This investigation will hopefully serve as a starting point for future studies related to the influence zone, including formulations that take more information into account aside from UDL values only. There were several limitations within this study, notably not accounting for serviceability checks and limiting the design space to positively loaded UDLs. The effects of torsion and lateral loading could also be considered. Further studies could be conducted for different material and design information assumptions, while studies could also be expanded to 2D continuous frames and shells, with fixed and semi-rigid connections. The influence zone concept and associated results could be a helpful piece of information when designing any structural system in practical engineering by informing structural engineers on what design information is actually relevant to properly size structural members. Furthermore, the framing of this novel influence zone concept within structural engineering may encourage awareness in practising engineers on what the size of the influence zone of their particular structural system is, which may have applications in other research areas such as structural health monitoring [486], complex thermomechanical actions on structures [487] and in the development of generalised design models [140].

4.6 Conclusions

A novel concept termed the *influence zone* was proposed in relation to continuous beam systems. The investigation developed a local and global formulation, of which the latter one was explored numerically with design constraints applicable to steel framed buildings. The key challenge was the explicit definition of critical load arrangements to allow the computational feasible generation of design datasets and evaluation of their respective influence zones. The investigation led to three important outcomes:

- The development of polarity sequences and polarity zones which led to the demarcation between previously known flexural load arrangements and the newly discovered shear load arrangements, with an explicit span limit equation for when these novel load arrangements occur.

- Two algorithms capable of finding these two types of load arrangements, and providing evidence that they encompass all critical permutations in comparison to the naive, brute-force approach.
- The generation of design datasets from which the influence zone values for various degrees of design complexities and error thresholds could be rigorously studied. For error thresholds deemed acceptable in structural design, which in this work was considered to be $\epsilon_{\max} \leq 0.5\%$, the influence zone for continuous beams within steel framed building under ultimate state considerations is on average less than 3, going to a maximum influence zone value of 5. This influence zone value is likely to be valid for most design situations whose spans and load values fall within the range of the design constraints considered in Section 4.3.1.

The influence zone is a heuristic design tool that differentiates itself from influence lines (and influence surfaces) and demonstrates the value of the inverse problem perspective through which it was evaluated by. This study opens the scope for future research, notably in the evaluation of influence zones for various materials and structural systems, validating and explicating the existence of shear load arrangements, and encouraging research on improving the existing algorithm that identifies them.

Data statement

Data related to the design datasets and the influence zone results, along with the load arrangement algorithms, are available at an open-source data repository [475].

Appendix

4.A Algorithm 1 - Flexural load arrangements

```
# Create the alternating load arrangements - altLoadArr
if m % 2 == 0: altLoadArr = [[1,0]*(m//2)]
else: altLoadArr = [[1,0]*(m//2) + [1]]

# Create the adjacent load arrangements - adjLoadArr
adjLoadArr = []
if m > 1:
    for i in range(m-1):
        # Create the start loadArr
        if i % 2 == 0: startLoadArr = [1,0]*(i//2)
        else: startLoadArr = [0,1]*(i//2) + [0]

        # Create the end loadArr
        if (m-i) % 2 == 0: endLoadArr = [0,1]*((m-i-2)//2)
        else: endLoadArr = [0,1]*((m-i-2)//2) + [0]

        # Append loadArr together with adjacent loaded spans
        adjLoadArr.append(startLoadArr + [1,1] + endLoadArr)

# Create positive J_flex load arrangements
J_flex_pos = altLoadArr + adjLoadArr

# Evaluate polar opposites - negative J_flex
J_flex_neg = []
for loadArr in J_flex_pos:
    J_flex_neg.append([1 if act == 0 else 0
                       for act in loadArr])

# Evaluate J_flex
J_flex = J_flex_pos + J_flex_neg

```

Algorithm 4.1: Flexural load arrangement algorithm in Python with both alternating and adjacent arrangements for a continuous beam system with m members that creates set $\mathbf{J_{flex}}$ with O(m) time complexity. This algorithm first generates the alternating load case arrangements depending on the parity of the system size m. Subsequently, the algorithm enters a for-loop to create the adjacent load arrangements by iterating through each possible start-index. Finally, the algorithm creates the polar opposites of each load arrangement, and saves it as the *J_flex* variable.

4.B Algorithm 2 - Shear load arrangements

```python
# Function to identify shear load arrangements
def shearLoadArr(loadArr: list, shearBeams: list, start: int):
    # Iterate in both directions
    for direction in [-1, 1]:
        # Establish while loop variables
        finishing = False; finished = False
        updating = False; i = start

        # Iterate through the beam system
        while finished == False:
            i = i + direction # Move to the next beam

            # Case 1: End of beam system is reached
            if i < 0 or i >= len(loadArr):
                finished = True

            # Case 2: No shear beam has been encountered yet
            elif updating == False and finishing == False:
                # Check if current beam is a shear beam
                if i in shearBeams:
                    updating = True; updateAct = loadArr[i]

            # Case 3: A shear beam has been encountered
            elif updating == True and finishing == False:
                # Update activation factor of current beam
                loadArr[i] = updateAct
                # Check if current beam is a shear beam
                if i not in shearBeams:
                    updating = False; finishing = True

            # Case 4: Alternate remaining activation factors
            elif finishing == True:
                loadArr[i] = (loadArr[i-direction] + 1) % 2

                # If another shear beam is encountered
                if i in shearBeams:
                    updateAct = loadArr[i]
                    updating = True; finishing = False

    return loadArr

```

Algorithm 4.2: Shear load arrangement algorithm in Python to generate arrangements belonging to set $\mathbf{J_{shear}}$ based on a given flexural load arrangements *loadArr*, the indices of susceptible *shearBeams*, starting at beam index *start*, for a system size with m members. The algorithm iterates through the beam system from the provided *start* index in both the left and right directions. Within the for-loop, four different cases are checked,. When a shear beam is found (Case 3) then the polarity (positive or negative) of the load arrangement is translated to the next subsequent beam. Case 4 ensures that all further polarities retain the overall alternating load arrangement pattern. One single pass has a time complexity of O(m), yet generating the entire set $\mathbf{J_{shear}}$ is $O(m^2\, 2^n)$.

Chapter 5

Article: "Machine learning for structural design models of continuous beam systems via influence zones"

Foreword commentary

This article titled "Machine learning for structural design models of continuous beam systems via influence zones" was submitted for review to *Inverse Problems* on August 22nd, 2023. The paper develops a machine learned structural design model from the inverse problem perspective introduced in Chapter 3 and takes advantage of the influence zone devised in Chapter 4 to build a generalisable model for the design of continuous beam systems. Aside from developing the model, the work also generates and makes a dataset and neural network training script available to allow for verifiability of results and to encourage future research efforts. The work was entirely conducted by Adrien Gallet, with co-authors Andrew Liew, Iman Hajirasouliha and Danny Smyl providing supervision along with validating results, and reviewing the final draft.

Abstract

This work develops a machine learned structural design model for continuous beam systems from the inverse problem perspective. After demarcating between forward, optimisation and inverse machine learned operators, the investigation proposes a novel methodology based on the recently developed influence zone concept. The aim of this approach is to conceptualise a non-iterative structural design model that predicts cross-section requirements for continuous beam systems of arbitrary system size. After generating a dataset of known solutions, an appropriate neural network architecture is identified, trained, and tested against unseen data. The results show a mean absolute percentage testing error of 1.6% for cross-section property predictions, along with a good ability of the neural network to generalise well to structural systems of variable size. The CBeamXP dataset

generated in this work and an associated python-based neural network training script are available at an open-source data repository to allow for the reproducibility of results and to encourage further investigations.

5.1 Introduction

It was recently argued that structural design is an *inverse problem* [140], in which one estimates the *model parameters* (the causal factors, such as cross-section properties) of possible structural solutions from a set of *structural utilisations* (the observations, such as major axis bending capacity). This inverse problem perspective, highlighted in Figure 5.1, is underscored by the ill-posed characteristics structural design shares with other inverse problems [181], which in civil and structural engineering include subject areas such as structural health monitoring [208, 277], self-sensing smart materials [380, 488] and forensic blast engineering [193, 195].

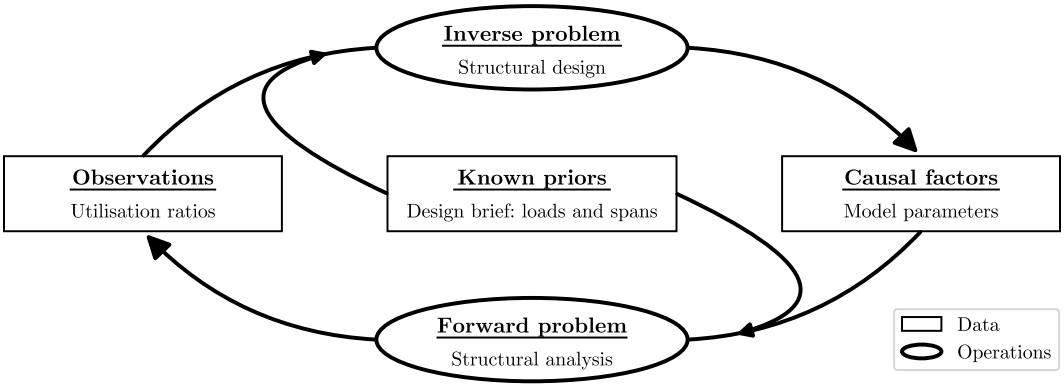


Figure 5.1: The inverse problem perspective for structural design, which relies on known priors such as design brief details of loading and span requirements along with observations of utilisation ratios that represent structural adequacy to evaluate the model parameters of a solution, such as size, shape and topology of a viable structure. Structural analysis is treated as the forward problem.

Inverse problems are predominantly solved iteratively [489], and unsurprisingly so is structural design [14], often with the help of structural optimisation such as size [490, 491], shape [492, 493], topology [494, 495] and layout optimisation [496, 497]. Provided that a clear objective function exists, these techniques are the state of the art for solving the structural design inverse problem *iteratively*.

However, in industry, the uptake of iteration based design approaches face certain barriers, including high computational costs [498], complex outputs that require additional post-rationalisation [188], and demand a particular expertise from practising design engineers that can be absent from engineering curriculums [499]. These challenges have encouraged researchers to investigate the use of machine learning (ML) methodologies for structural design [73]. This parallels a similar development of using ML within the domain of inverse problems [165], with exemplary applications in areas such as structural

health monitoring [129, 168], that aid or replace the optimisation problem with *learned* components.

The earliest application of such machine learned components for structural design occurred in 1989 [500] with simplified perceptron models. This research was followed in the 1990s by more advanced feed-forward neural networks for simple reinforced concrete beam depth estimations [97] as well as cross-sectional area predictions of trusses [98, 501]. Whilst other machine learning modalities such as support vector machines [502] have also been studied, neural networks tend to outperform other ML models archetypes in terms of prediction error [99].

More recently, deep learning techniques have been investigated for structural design. These include convolutional [100, 503] and generative adversarial networks [504, 505] to accelerate topology optimisation, and the application of variational auto encoders for structural design space exploration [506]. A common limitation across such investigations is the inability for the same machine learned model to generalise to differently sized topologies and structural arrangements. These two challenges, highlighted by design ill-posedness and the inability of previous machine learning models to generalise to structural arrangements of arbitrary size, have motivated the work presented here.

This investigation has two objectives. The first objective is to reconcile the relationship between structural design, inverse problems and machine learning by developing a non-iterative structural design model for continuous beam systems using a multi-layer neural network. The authors believe that this perspective could serve as a framework to distinguish between different types of machine learning applications within the field of structural engineering in the future. The second objective is to address the inherent issue of generalisability in respect to system size by taking advantage of a recently developed concept known as a continuous beam's influence zone [441]. This technique could potentially form the basis to generalise a design model for continuous structural systems of arbitrary topology, and might complement other techniques that attempt to address the generalisability issue such as graph neural networks [101, 507].

The paper is structured as follows: Section 5.2 explores the problem statement from the inverse problem perspective and provides the rationale for machine learned design models, Section 5.3 explains the methodology employed to develop the generalisable structural design model, Section 5.4 presents the step-by-step process of the neural network development process, and Section 5.5 discusses the model's generalisability and prediction variability, along with suggestions for further research.

5.2 Problem statement

5.2.1 A novel perspective

The inverse problem perspective for structural design as shown in Figure 5.1 consists out of two operations, the *forward* and *inverse problem* (shown as the bottom and top ellipses, respectively) and three sets of data: *observations*, *known priors* and *causal factors* (shown as rectangles from left to right, respectively). One of the underpinning features of the inverse problem perspective is the clear demarcation between structural analysis and structural design, a distinction often re-iterated in engineering philosophy [4, 11], yet

never linked to the corresponding nature of forward and inverse problems, respectively.

Both the forward (structural analysis) and the inverse problem (structural design) rely on *known priors*, shown centrally in Figure 5.1, which can be thought of as constraints set by a design brief such as load and span requirements. During design they inform and regularise the search space of *causal factors* (model parameters such as section properties and topologies), and in analysis they allow the evaluation of *observations* (utilisation ratios such as ultimate (ULS) and serviceability limit states (SLS) [508]). Unlike traditional inverse problems, the observations are not measured physically, yet are expressed theoretically based on the utilisation ratios that could be measured from a compliant design solution which the set of causal factors correspond with; inverse problems are not defined by the physicality of the observations.

Within this context, the application of machine learning in structural engineering can be split into three categories based on the type of operations the machine learned components replace. These categories help distinguish between fundamentally different types of machine learning applications that occur within the context of structural engineering and are identifiable across different decades of the literature:

a) *ML forward operators*: machine learned components that aid or accelerate solving the forward problem (structural analysis) to inform or validate design decisions. Examples include neural network like models as quick re-analysis tools for optimum design (1991) [509] and machine learning models to determine the buckling behaviour and model decomposition of thin-walled members required for structural analysis (2023) [510].

b) *ML optimisation solvers*: machine learned components entirely motivated by the traditional iterative solution process to arrive at structural designs. Examples include "neural dynamic models" developed as an alternative structural design optimisation technique (1995) [511] and a physics informed neural energy-force network that replaces both the structural design and analysis steps (2023) [512].

c) *ML inverse operators*: machine learned components which solve the inverse problem (structural design) by mapping a set of structural utilisations and known priors to model parameters directly. Examples include estimating cross-sectional properties for simple trusses directly based on known optimum examples using neural networks (1994) [98] and approximating topological optimised structures in real-time using convolutional neural networks (2022) [513].

These three categories can also be differentiated visually as shown in Figure 5.2. It is worth noting that the field of ML forward operators has likely received the most research attention in the form of "surrogate models" [514, 515]. In this respect, machine learned optimisation solvers and inverse operators are less common. Furthermore, the machine learned forward operators and optimisation solvers identified above typically require some form of iteration to achieve structural design; machine learned inverse operators on the other hand can be non-iterative [98, 513]. The ability to provide real-time design feedback is of particular interest to address the limitations of current iterative structural design approaches. To this end, and in support of the inverse problem perspective, this paper will focus on developing a non-iterative structural design model for continuous beam systems.

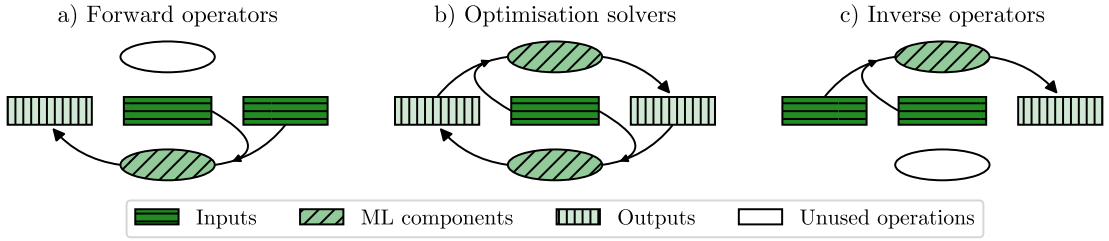


Figure 5.2: Types of machine learning (ML) components from the inverse problem perspective. Shapes in each sub-figure; ellipses: inverse problem (top), forward problem (bottom); rectangles: observations (left), known priors (middle), causal factors (right).

5.2.2 Design problem: continuous beam systems

Continuous beam systems arise in structural engineering when rigid connections between members are required or unavoidable due to design or material considerations. The support fixity and structural connectivity render the system statically indeterminate. This poses a challenge from a design perspective, since the compliance of cross-sectional properties cannot be evaluated without knowledge of their magnitudes; this results in an iterative design process, especially for complex design scenarios with heterogeneous loading and span conditions [465].

Figure 5.3 highlights the design problem for continuous beam systems from the inverse problem perspective. The known priors, which are shown centrally as the design brief, include the number of members m in the system indexed by i with span length L_i from vector $\mathbf{L} = [L_i]_{0 \leq i < m}$, subjected to uniformly distributed loads (UDLs) ω_i from vector $\boldsymbol{\omega} = [\omega_i]_{0 \leq i < m}$ (representing the total, factored load including self-weight). These known priors and the utilisation ratios $\mathbf{u}$ of the members, shown on the left in Figure 5.3, are needed to evaluate the causal factors, shown on the right as the cross-section property vector $\mathbf{P} = [\mathbf{P}_i]_{0 \leq i < m}$.

The design problem is complicated due to the existence of c potentially critical load arrangements J indexed by j from set $\mathbf{J} = [J_j]_{0 \leq j < c}$ shown at the bottom of Figure 5.3. The size c of $\mathbf{J}$ was studied in [441]. Each of these load arrangements cause different structural responses such as bending moments $\mathbf{M}$, and will give rise to a matrix of utilisation ratios $u_{i,j}$ to form matrix $\mathbf{u} = [u_{ij}]_{0 \leq i < m, 0 \leq j < c}$ that can be evaluated with structural analysis to check for structural compliance ($u_{i,j} \leq 1.0$). Instead of repeatedly assuming cross-section properties $\mathbf{P}$ and conducting structural analysis calculations until the matrix of utilisation ratios $\mathbf{u}$ are compliant, a machine learned inverse operator relies solely on the known priors and the utilisation ratios $\mathbf{u}$ to directly evaluate the cross-section properties $\mathbf{P}$.

For the purpose of the continuous beam system considered in this work, several assumptions will be made: members are made out of S355 steel, are considered laterally restrained (and hence not susceptible to lateral instability), Timoshenko-Ehrenfest beam theory is used to model this system and the structure will be analysed elastically yet designed against ULS plastic cross-section property checks as allowed by Eurocode EN 1993-1-1 5.4.2 (2) [466]. The cross-sectional properties to be evaluated include the major axis second moment of area I, the major axis shear area A_z and the major axis plastic

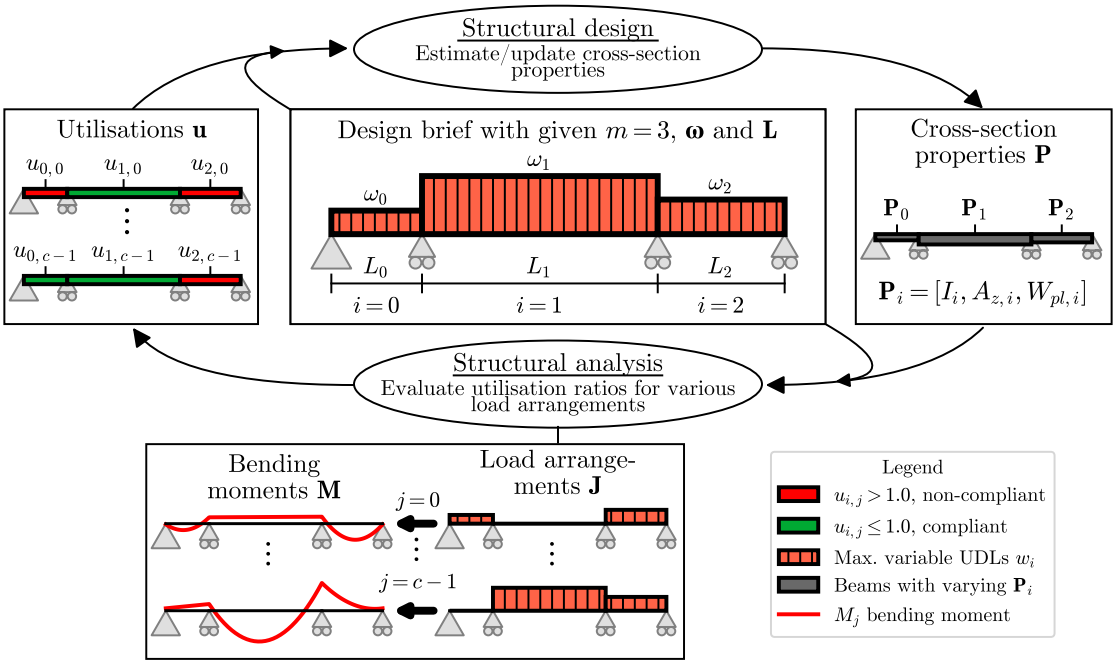


Figure 5.3: Design process of a continuous beam system from the inverse problem perspective.

section modulus W_{pl} for each member i. Together they form the member-based cross-sectional property vector $\mathbf{P}_i$:

$$\mathbf{P_i} = [I_i, A_{z,i}, W_{\text{pl},i}] \tag{5.1}$$

The structural analysis operation in Figure 5.3 is defined by a forward operator:

$$\mathbf{u} = O_{\text{forw}}(m, \boldsymbol{\omega}, \mathbf{L}, \mathbf{P}) \tag{5.2}$$

and similarly the structural design operation by an inverse operator:

$$O_{\text{inv}}(m, \boldsymbol{\omega}, \mathbf{L}, \mathbf{u}) = \mathbf{P} \tag{5.3}$$

where both O_{forw} and O_{inv} rely on the same known priors, the design brief information m, $\boldsymbol{\omega}$ and $\mathbf{L}$ that define the structural system and design problem.

5.2.3 The need for machine learned inverse operators

Defining an explicit non-iterative inverse operator for Equation 5.3 is challenging due to the difficulty of inverting the forward operator and is directly linked to the ill-posed nature common across most inverse problems [181]. A quantitative evaluation of the extent of ill-posedness in structural design is not obvious, however it is possible to describe why the structural design problem shown in Figure 5.3 is ill-posed, namely due to the infinite number of viable solutions, and:

a) The physical limitations introduced by yielding, buckling, serviceability that arise when combining the forward model with structural codes, resulting in a discontinuous relationship between the observations and causal factors.

b) The indeterminacy of the continuous beam system which increases with the number of members of the system.

The ill-posedness of structural design has further been demonstrated in Section 3.3. It is worth noting that any structural analysis forward operators themselves are approximations of the true behaviour of structures, and dealing with this associated uncertainty is a key challenge in design. For example, engineers need to decide if the assumptions and simplifications of structural analysis models, such as the material response (e.g. perfectly elastic) and underlying beam theory (e.g. Euler–Bernoulli theory), are representative of the structure's true behaviour.

The difficulty of inverting a forward operator can be shown mathematically. Typically, the O_{forw} operator contains two steps. The first step, defined by $O_{\text{forw},1}$ would evaluate the structural response of the system when subjected to a set of external forces $\boldsymbol{\omega}$ in terms of deflections and internal forces, and the second step, defined by $O_{\text{forw},2}$, would take these structural response observations to evaluate the utilisation ratios based on design codes. Consider for example building a $O_{\text{forw},1}$ operator using the stiffness matrix method to evaluate the internal force vector $[f_p]_i$ for member i defined as:

$$[f_p]_i = [V_{1,i}, M_{1,i}, V_{2,i}, M_{2,i}]^\intercal \tag{5.4}$$

where V, and M represent the internal shear forces and bending moments within member i at the start (index 1) and end of the member (index 2). Let us also assume, for simplicity, that the members consist out of steel with E and G for the Youngs and shear modulus, respectively, with a maximum yield stress of σ_y. In this case, the internal forces $[f_p]_i$ for each member could be evaluated using a simplified Timoshenko-Ehrenfest beam theory for a *single* load arrangement by Equation 5.5. To achieve this, $[k_{pq}]_i$ is defined as the local stiffness matrix shown in Equation 5.6, $[K_{pq}]$ as the global stiffness matrix in Equation 5.7, $[d_q]$ as the nodal displacement vector in Equation 5.8 with $F_p([\omega_i])$ as the external force vector, where rows and columns of all matrices are indexed by p and q, respectively:

$$[f_p]_i = [k_{pq}]_i [d_q] \tag{5.5}$$

$$[k_{pq}]_i = \frac{EI_i}{L_i^3(1-\varphi)} \begin{bmatrix} 12 & 6L_i & -12 & 6L_i \\ 6L_i & 4L_i^2 & -6L_i & 2L_i^2 \\ -12 & -6L_i & 12 & -6L_i \\ 6L_i & 2L_i^2 & -6L_i & 4L_i^2 \end{bmatrix}, \quad \varphi = \frac{12EI_i}{A_z G L_i^2} \tag{5.6}$$

$$[K_{pq}] = [k_{pq}]_0 + [k_{pq}]_1 + \ldots + [k_{pq}]_{m-1} = \sum_{i=0}^{m-1} [k_{pq}]_i \tag{5.7}$$

$$[d_q] = [K_{pq}]^{-1} [F_p([\omega_i])] \tag{5.8}$$

These operations can be succinctly written to transform the cross-section vector $\mathbf{P}$ with help of the known priors m, $\boldsymbol{\omega}$, $\mathbf{L}$ into the internal forces vector for each member i:

$$O_{\text{forw},1,i} = [f_p]_i = [k_{pq}]_i \left[\sum_{i=0}^{m-1} [k_{pq}]_i \right]^{-1} \left[F_p([\omega_i]) \right] \tag{5.9}$$

Inverting this equation to yield O_{inv} is difficult since it would require separating or decomposing the individual cross-section properties $\mathbf{P}_i$ out of the stiffness matrices $[k_{pq}]_i$. This cannot be done without, at minimum, making some assumptions about the relative proportions of the cross-section properties from one member to another. Inverting the second step of the forward operator $O_{\text{forw},2}$ poses further challenges. Suppose $O_{\text{forw},2}$ transforms the internal member forces $[f_p]_i$ to evaluate the governing (critical) utilisation ratios indexed by r for t design checks for a single load arrangement J. For example, using the steel design code EN 1993-1-1 [466]:

$$\begin{gathered} u_i = O_{\text{forw},2,i}(O_{\text{forw},1,i}) = \max([u_{ir}]_{0\leq i<m,\, 0\leq r<t}) = \max(u_{i,0}, u_{i,1}, \cdots, u_{i,t-1}) \\ \text{where:} \\ u_{i,0} = \frac{V_{1,i}}{A_{z,i}\,\sigma_y/\sqrt{3}}, \quad u_{i,1} = \frac{M_{1,i}}{W_{\text{pl},i}\,\sigma_y} \\ u_{i,2} = \frac{V_{2,i}}{A_{z,i}\,\sigma_y/\sqrt{3}}, \quad u_{i,3} = \frac{M_{2,i}}{W_{\text{pl},i}\,\sigma_y} \\ \vdots \\ u_{i,t-1} \text{ for other compliance checks} \end{gathered} \tag{5.10}$$

The difficulty here is that the governing utilisation ratio can change according to the known priors of the problem statement. This means that an individual equation for each possible critical design check would need to be derived. For example different design equations exist for the same structural check depending on the type of cross-section (Class 1 vs. Class 4) [466] a final design solution might contain, which is not known ahead of time. Note also that the equations above do not even consider the serviceability limit state, the multiple load arrangements $\mathbf{J}$ which may be critical, nor the need to sufficiently discretise individual beam members.

It is because of the challenges identified above that machine learned inverse operators are particularly appealing, since they can approximate a relationship between a set of variables that may be difficult to encode explicitly [516]. Given a dataset generated by the O_{forw} operator that maps a set of cross-section properties $\mathbf{P}$ to compliant utilisations ratios $\mathbf{u}$, one can train a probabilistic machine learning model $O^{\dagger}_{\text{inv}}$ with parameters $\boldsymbol{\theta}$ to map the set of bounded utilisation ratios $\mathbf{u}$ back to the cross-sectional properties $\mathbf{P}$ with known priors m, $\boldsymbol{\omega}$ and $\mathbf{L}$:

$$O^{\dagger}_{\text{inv}}(\boldsymbol{\theta}, m, \boldsymbol{\omega}, \mathbf{L}, \mathbf{u}) \approx \mathbf{P} \tag{5.11}$$

By generating a dataset of valid structural designs with the help of existing optimisation approaches that contain the forward operator, a supervised machine learning model

can be trained to learn the mapping of known priors and utilisations to cross-sectional properties directly. This represents a fundamental shift from traditional approaches employed in structural design that rely on engineering expertise and computationally expensive structural analysis or optimisation models at the point of design application. Machine learned inverse operators create non-iterative structural design models for which there currently exist no explicitly defined equivalents. Instead of focusing on accelerating forward models, computational resources can be invested in generating a dataset using physically complex yet realistic modelling assumptions. These machine learned structural design models aim to provide significantly greater generalisability than typical rules of thumb employed in design whilst still providing real-time feedback, benefit non-expert stakeholders whose own decision making relies on structural design outcomes and improves design knowledge permanence which can be difficult to attain due to industry turnover.

5.3 Methodology

5.3.1 Choosing an appropriate machine learning model archetype

The aim of the inverse operator $O^{\dagger}_{\text{inv}}$ is to predict the cross-section property vector defined by Equation 5.1 numerically; therefore $O^{\dagger}_{\text{inv}}$ will be a regression model. This restricts the types of supervised machine learning models of interest. The complexity and size of the design space are likely to demand a large dataset size discouraging the use of instance-based models such as the k-nearest neighbour algorithm that store similarity measurements in memory [517]. Similarly, support and relevance vector machines become impractical for datasets containing more than 3000 samples [518]. The non-linearity of the design problem voids the applicability of linear regression models, and decision trees (including the ensembled variants such as random forests) perform better at classification tasks [80].

These reasons motivated the use of neural networks, in particular multilayer neural networks (MLPs), a choice which is supported by evidence that suggests neural networks outperform other data-driven approximation algorithms in structural engineering applications [99]. Although various archetypes exist ranging from convolutional (CNNs), recurrent (RNNs) and graph-based types (GNNs), MLPs are commonly used in literature [98, 501], and the results within this work could prove useful as a comparative performance measure for more advanced deep learning architectures [100, 504, 506] in future studies.

Multilayer neural networks have a fixed-dimensional input vector $\mathbf{x}_0$ of size n that map to the output vector $\mathbf{x}_D$ of size o with D layers. In this study, a network has $D-1$ hidden layers of height H each indexed by d and are defined by $f_d(\mathbf{x}_d)$, which contains a (non-linear) activation function a_d with weight matrix $\mathbf{w}_d$ and bias vector $\mathbf{b}_d$. The weight matrices and bias vectors of each layer form the model's parameters $\boldsymbol{\theta} = (\mathbf{w} = [\mathbf{w}_d]_{0\leq d<D}, \mathbf{b} = [\mathbf{b}_d]_{0\leq d<D})$. Multiple hidden layers give form to the neural network f through a function composition defined as:

$$\begin{gathered} f : \mathbb{R}^n \to \mathbb{R}^o \\ f(\mathbf{x}_0) \to \mathbf{x}_D : f_{D-1} \circ \ldots \circ f_1 \circ f_0(\mathbf{x}_0) \\ f_d(\mathbf{x}_d) = a_d(\mathbf{w}_d \mathbf{x}_d + b_d) \end{gathered} \tag{5.12}$$

The exact choice of architecture in terms of depth D, height H and activation functions a_d of the network as indicated in Equation 5.12 will require experimentation to achieve acceptable performance with a good bias-variance trade-off [519]. More importantly though, the features used for the input vector $\mathbf{x}_0$ will require careful consideration to create a generalisable inverse operator $O^{\dagger}_{\text{inv}}$ as set out in Equation 5.11.

5.3.2 Selecting appropriate neural network features

Feature selection, the process of choosing appropriate inputs, is essential for a machine learning model to generalise well to unseen data points. Unnecessary or irrelevant features can cause a model to learn a relationship with target variables that are not representative of the physical behaviour of the system, and thereby lead to worse results when interpolating within or extrapolating beyond the training set.

Previous studies of neural network based design models selected features relevant to the singular topology of the structural system at hand [98, 99]. Such approaches expose the largest limitation of multilayer neural networks: the fixed-dimensionality of the input vector [101]. These models may perform well for the particular topology they were trained against, yet the same model tends to perform worse or may not be applicable for differently sized structural systems, which severely limits their utility.

To address this limitation, this work takes advantage of a recently developed concept known as the *influence zone* developed in Chapter 4. The influence zone k_{max} is a measure of the extent to which surrounding design information is relevant for the utilisation evaluation of members. Whilst k_{max} differs for each member within a continuous structural system as shown in Figure 5.4, for well defined design constraints and error thresholds, the maximum value of k_{max} within continuous beam systems converges towards a non-negative integer. The influence zone of member g is found when the following two conditions are met:

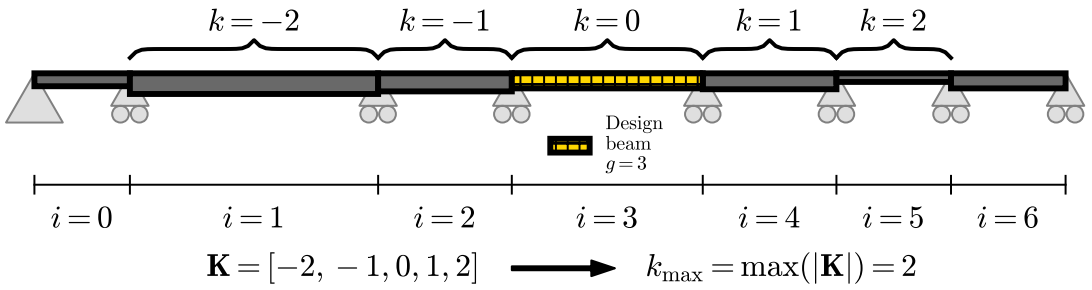


Figure 5.4: A figurative influence zone of $k_{\text{max}}=2$ for design beam $g=3$ within a $m=7$ continuous beam system with $\epsilon_{\text{max}}=0.02$ limit.

$$\left| 1 - \frac{u_{g,\text{cap}}}{u_{g,\text{true}}} \right| \leq \epsilon_{\max}$$
$$u_{g,\text{cap}} = \max \left(\sum_{i=-k_{\max}}^{k_{\max}} \mathbf{u}_{g,i,j}(\boldsymbol{\omega}, \mathbf{L}, \mathbf{P}, \mathbf{J}) \right) \tag{5.13}$$

In Equation 5.13, $\epsilon_{\max}$ represents the maximum error threshold due to the difference between $u_{g,\text{cap}}$, the captured utilisation ratio of the design beam g for a given value of $k_{\max}$, and $u_{g,\text{true}}$, the true utilisation ratio of the design beam g if the contribution of all members of the continuous beam system had been considered. $\mathbf{u}_{g,i,j}$ is the utilisation ratio contribution function towards the design beam g by member i based on the UDLs $\boldsymbol{\omega}$, spans $\mathbf{L}$, structural properties $\mathbf{P}$ and load arrangements $\mathbf{J}$. If the requirement for $\epsilon_{\max}$ is sufficiently relaxed, the maximum influence zone $k_{\max}$ can be determined for any potential continuous beam system arising under the specified design constraints as shown in Chapter 4. This is extremely useful to ensure the relevant inputs are fed to a machine learning model. The influence zone thereby acts as a mechanics-driven feature selection process, and provides the basis to generalise to a continuous beam system of arbitrary size m.

5.3.3 Structuring features for arbitrary system size m

Zero-padding, the process of adding zero-valued inputs, arises in the context of convolutional neural networks to allow trained kernel filters to parse through the edges and corners of an input space [519]. This technique can also be applied to continuous beam systems to conceptualise a design model that parses over a structural system to make localised predictions for each member i. If the design information, here the UDLs $\boldsymbol{\omega}$ and span $\mathbf{L}$ that fall within the influence zone are provided as inputs to the network, then this would result in an input vector $\mathbf{x}_0$ of size $n = 4k_{\max} + 2$, as shown in Figure 5.5 for member $i = 3$ and $i = 0$. These inputs should, based on the principle of the influence zone, contain the relevant information to predict the cross-section properties of member d with an accuracy of up to $\epsilon_{\max}$.

It is now conceivable that the same neural network could be used to make a prediction for any other member using a fixed-dimensional input vector $\mathbf{x}_0$ by structuring the inputs relative to the position of the design beam's influence zone. This would include end-span beams by using zero-padding as shown in Figure 5.5 for member $i = 0$. Zero-padding in this instance is also logically consistent, since it corresponds with a beam that does not in fact exist; that is a beam of zero length L and zero UDL load ω. Therefore, instead of structuring the neural network based on the absolute position of a beam within the entire continuous beam system (as indexed by i), the inputs are structured relatively to the influence zone of a design beam g to predict the cross-section properties of that design beam $\mathbf{P}_g$.

Whilst such an approach will require m forward passes (inferences) to predict the cross-section properties of an m sized system (one prediction per beam), it enables the same neural network to be applied to continuous beam systems of any size m for which

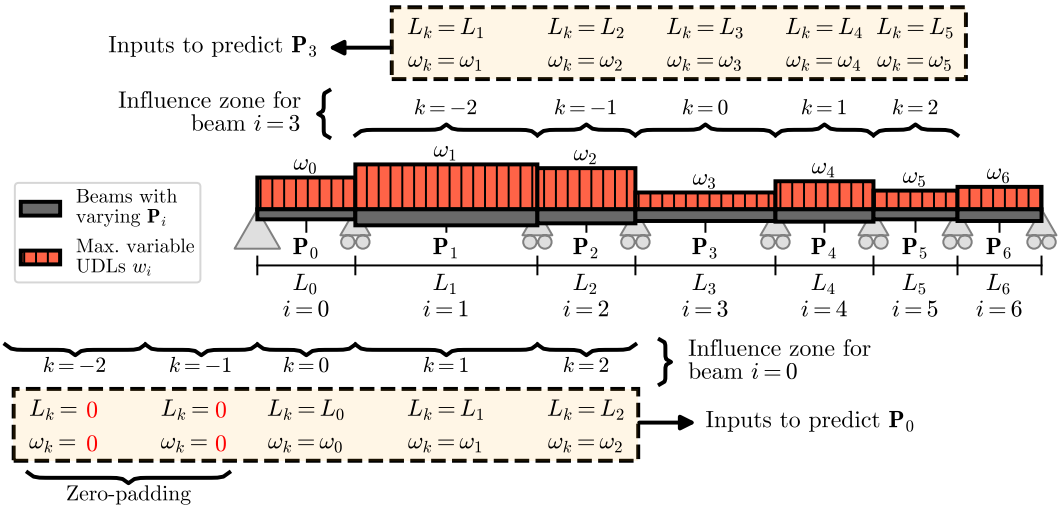


Figure 5.5: An illustration demonstrating the structuring of the neural network inputs using influence zones and zero-padding with $k_{\max} = 2$, leading to $n = 4k_{\max} + 2 = 10$ inputs.

the maximum influence zone value $k_{\max}$ that determined the size of the input vector $\mathbf{x}_0$ applies to. Based on the principle of influence zones, the neural network will be able to make predictions for continuous beam systems of size greater than the fixed-dimensional input vector size $m > 2k_{\max} + 1$, since any information outside the influence zone should by definition not be relevant (for an assumed $\epsilon_{\max}$). On the other hand, zero-padding allows the same neural network to predict along system edges as well as continuous beam systems of sizes smaller than the influence zone.

5.3.4 Generating an appropriate dataset

As explained previously, the maximum influence zone $k_{\max}$ size depends on the design constraints and an assumed error threshold $\epsilon_{\max}$. These design constraints can be defined by setting minimum and maximum ranges on the known priors, UDLs $\boldsymbol{\omega}$ and spans $\mathbf{L}$, as well as the cross-section properties within vector $\mathbf{P} = [I, A_z, W_{\text{pl}}]$:

$$\begin{aligned} \omega_{\min} &< \omega_i &&< \omega_{\max} \\ L_{\min} &< L_i &&< L_{\max} \\ I_{\min} &< I_i &&< I_{\max} \\ A_{z,\min} &< A_{z,i} &&< A_{z,\max} \\ W_{\text{pl},\min} &< W_{\text{pl},i} &&< W_{\text{pl},\max} \end{aligned} \tag{5.14}$$

Constraints for each of these variables were chosen generously to cover the entire range of potential continuous beam systems that arise in structural design (from fixed framed multi-storey buildings to continuous bridge decks). Table 5.1 highlights the ranges chosen for the UDLs and spans, along with the interval at which these inputs were sampled at using a random uniform distribution.

Property	Min.	Interval	Max.
ω [kN/m]	5	5	325
L [m]	0.5	0.5	20.0

Table 5.1: Ranges and intervals of know priors used for the influence zone evaluation and data generation.

Although arbitrary cross-section property combinations could have been chosen for I, A_z and W_{pl}, using cross-section properties from an explicitly defined set ensures the predicted cross-section properties are physically realistic. Initially, the standardised UB cross-sections from BS EN 10365 [467] were considered. However, the minimum and maximum cross-section properties from this set were not sufficient for the lightest and heaviest loading conditions possible under the design constraints set by Table 5.1. For this reason, a set of custom I-sections were generated and used exclusively for all members.

These custom I-sections were generated by averaging the geometrical ratios between the web depth d_{w}, flange thickness t_{f}, flange breadth b_{f} and the web thickness t_{w} that arise in BS EN 10365 [467]. Aside from ensuring that they share commonalities with the UB BS EN 10365, this process also ensured at minimum Class 2 sections [466] to allow the use of plastic cross-section properties. 1000 individual cross-sections were generated that ensured equal spacing across these ratios. The resulting granularity (as opposed to the 91 within BS EN 10365) meant that the utilisation ratio precision achievable during data-generation was significantly higher. The custom I-sections and associated cross-section properties are shown in Table 5.2.

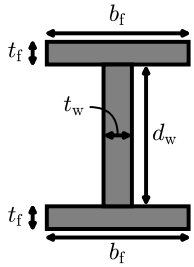
Diagram	Property	UB BS EN10365			Custom I-sections		
		Min	Mean	Max	Min	Mean	Max
	t_{w} [mm]	4.0	13.0	36.1	3.0	24.0	45.0
	d_{w} [mm]	112	550	928	30.3	243	455
	t_{f} [mm]	6.8	21.4	65.0	4.8	38.6	72.4
	b_{f} [mm]	76.0	220	421	55.5	444	833
	$d_{\text{w}}/t_{\text{w}}$	23.9	43.9	59.9	43.9	43.9	43.9
	$t_{\text{f}}/t_{\text{w}}$	1.19	1.61	1.90	1.61	1.61	1.61
	$b_{\text{f}}/t_{\text{w}}$	8.7	18.5	27.5	18.5	18.5	18.5
	A_{z} [cm^2]	4.47	87.9	334	4.1	329.2	922
	I [cm^4]	473	233×10^3	1.25×10^6	305	3.34×10^6	15.5×10^6
	W_{pl} [cm^3]	84.2	6110	28.0×10^3	49.4	44.9×10^3	167×10^3

Table 5.2: Cross-section properties comparison between Universal Beams (UB) from BS EN 10365:2017 and custom generated I-sections. Note in particular that mean dimension ratios ($d_{\text{w}}/t_{\text{w}}$ etc.) are identical for both groups of cross-sections.

Together, these efforts ensure that the dataset on which the neural network is trained on covers sufficient breadth in terms of the input and output space to generalise for a wide variety of continuous beam systems. The dataset generated based on the aforementioned design constraints, the concept of influence zones, and the technique of zero-padding were

chosen with the aim to maximise the generalisability of the inverse operator for any system size m, UDLs $\boldsymbol{\omega}$ and spans $\mathbf{L}$. This leaves only the utilisation ratios $\mathbf{u}$ as the remaining input variable in Equation 5.11. Instead of passing utilisation ratios as explicit inputs to the network, it was decided that the dataset will be generated so that all beams closely correspond to the target utilisation ratio u_{target}. The network will therefore implicitly learn the u_{target} from the data itself.

The dataset was generated by designing continuous beam systems of size $m = 2k_{\max}+1$ with each member having a span L and UDL ω value drawn from a random uniform distribution based on the discretised ranges and intervals specified in Table 5.1. These heterogeneous structural systems were modelled and optimised using third-party software (Rhino3D©, Grasshopper© and Karamba3D© [476]) after having identified the influence zone $k_{\max}$ for the design constraints in Table 5.1 and Table 5.2. The beams were optimised for minimum depth against ULS cross-section checks from EN 1993-1-1 6.2 [466] using a coupled analysis and design procedure [465] with a target utilisation ratio of $u_{\text{target}} = 0.99$.[1]

5.3.5 Neural network training procedure

The generalised neural network structure developed in this work is shown in Figure 5.6. Identifying an appropriate architecture in terms of height H, depth D and activation functions a_d requires experimentation. The choice of loss function J to compare predicted targets $\widehat{\mathbf{x}}_D$ against true targets $\mathbf{x}_D$ also form part of the experimentation process.

Loss functions and performance metrics

In this study, four different loss functions were investigated as shown in Table 5.3. These include the Mean Absolute Error (MAE) and Mean Square Error (MSE) loss functions that are commonly used for regression models. One limitation associated with both is that their derivates (in respect to predicted targets) back-propagate the model parameters $\boldsymbol{\theta}$ with no regards what the relative size of the error is in relation to the magnitude of the output variables I, A_z and W_{pl}.

This is problematic given the orders of magnitude difference between the largest and smallest section properties of the custom I-sections as shown in Table 5.2. An error of 100 cm^4 for I would cause the same back propagation adjustment using MAE or MSE regardless if the true second moment of area value target is 305 cm^4 or 305×10^5 cm^4. As a consequence, both MAE and MSE would prioritise minimising the absolute error, which mathematically favours target values of large magnitudes at the expense of smaller ones.

To address the above mentioned issue, percentage-based versions of both MAE and MSE were tested, defined in Table 5.3 as the Mean Absolute Percentage Error (MAPE) and the Mean Squared Percentage Error (MSPE). Whilst MAPE is commonly used, MSPE is not tested in practice. Both MAPE and MSPE ensure that during back-propagation, the optimiser updates model parameters in proportion to the relative deviation between predicted $\widehat{\mathbf{x}}_D$ and true outputs $\mathbf{x}_D$, which should be a better performance criterion to address the orders of magnitude difference in the output space that arise in these particular continuous beam systems.

[1] An overview of data generation model is presented in Appendix B.

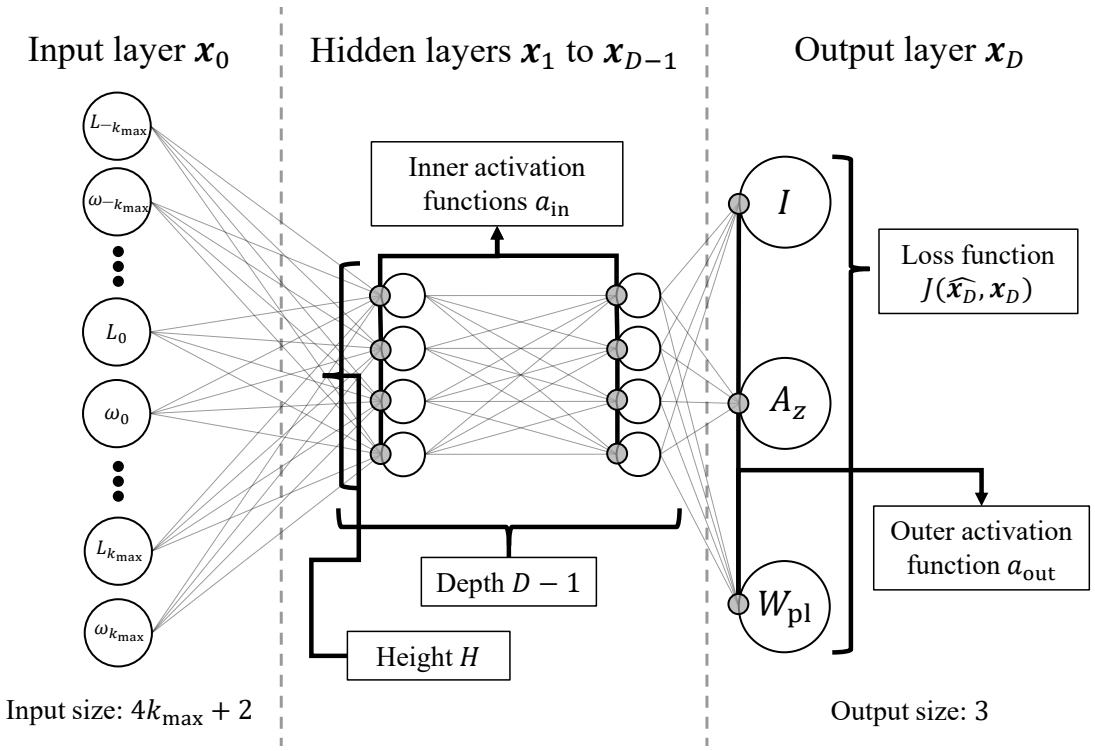


Figure 5.6: Generalised neural network structure with known priors from the influence zone $k_{\max}$ as the input layer $\mathbf{x}_0$ and cross-section properties of the beam as the output layer $\mathbf{x}_D$.

Regardless of the choice of loss function, MAPE will be used as a comparison metric between different networks. However to study the dispersion of prediction errors, an accuracy metric M will also be evaluated with minimum, 0.5%, 2.5%, 50% (median), 97.5%, 99.5% and maximum percentile values. This will allow the evaluation of the 95% and 99% confidence intervals (CI) and help identify the range of over and under prediction of outputs, which is important in the context of safe structural design:

$$M(\widehat{\mathbf{x}_D}, \mathbf{x}_D) = \frac{1}{p}\sum_{i=1}^{p}\frac{\widehat{x_{D,i}}}{x_{D,i}} = \frac{1}{3}\left(\frac{\widehat{I}}{I} + \frac{\widehat{A_z}}{A_z} + \frac{\widehat{W_{\text{pl}}}}{W_{\text{pl}}}\right). \tag{5.15}$$

Activation functions

The activation functions tested in this work are listed in Table 5.4, and includes the commonly used rectified linear unit (ReLU) function amongst others [519]. A distinction is drawn between the inner activation functions a_{in} within the hidden layers, and the outer activation function a_{out}, that evaluate the target values $\mathbf{x}_D$. All inputs and outputs were scaled between 0 and 1 by dividing the values by the maximum magnitude of the features and targets within the training and validation set, respectively. Therefore, it is important to choose only output activation functions compatible with the scaled values of the targets as reflected by the range of a_{out} functions listed in Table 5.4.

Name	Formula	Derivative
Mean Absolute Error	$J_{\text{MAE}} = \frac{1}{p}\sum_{i=1}^{p}\lvert\widehat{x_{D,i}} - x_{D,i}\rvert$	$J_{\text{MAE}}{}' = \frac{1}{p}\sum_{i=1}^{p}\begin{cases}\text{-1}, & \widehat{x_{D,i}} < x_{D,i}\\ 1, & \widehat{x_{D,i}} > x_{D,i}\end{cases}$
Mean Squared Error	$J_{\text{MSE}} = \frac{1}{2p}\sum_{i=1}^{p}(\widehat{x_{D,i}} - x_{D,i})^2$	$J_{\text{MSE}}{}' = \frac{1}{p}\sum_{i=1}^{p}\widehat{x_{D,i}} - x_{D,i}$
Mean Absolute Percentage Error	$J_{\text{MAPE}} = \frac{1}{p}\sum_{i=1}^{p}\left\lvert\frac{\widehat{x_{D,i}} - x_{D,i}}{x_{D,i}+\epsilon}\right\rvert$	$J_{\text{MAPE}}{}' = \frac{1}{p}\sum_{i=1}^{p}\begin{cases}\frac{\text{-1}}{\lvert x_{D,i}+\epsilon\rvert}, & \widehat{x_{D,i}} < x_{D,i}\\ \frac{1}{\lvert x_{D,i}+\epsilon\rvert}, & \widehat{x_{D,i}} > x_{D,i}\end{cases}$
Mean Squared Percentage Error	$J_{\text{MSPE}} = \frac{1}{2p}\sum_{i=1}^{p}\frac{(\widehat{x_{D,i}} - x_{D,i})^2}{\lvert x_{D,i}+\epsilon\rvert}$	$J_{\text{MSPE}}{}' = \frac{1}{p}\sum_{i=1}^{p}\frac{\widehat{x_{D,i}} - x_{D,i}}{\lvert x_{D,i}+\epsilon\rvert}$

Table 5.3: Loss functions to be tested with p predicted targets $\widehat{x}_D$, p true targets x_D and small ϵ to avoid division by zero errors.

Name	Formula	a_{in}	a_{out}
Rectified linear unit (ReLU)	$a_{\text{ReLU}} = \max(\mathbf{wx}+\mathbf{b}, 0)$	✓	✓
Sigmoid	$a_{\text{sigm}} = 1/(1+e^{\mathbf{wx}+\mathbf{b}})$	✓	✓
Hyperbolic tangent	$a_{\tanh} = \tanh(\mathbf{wx}+\mathbf{b})$	✓	
Exponential	$a_{\exp} = e^{\mathbf{wx}+\mathbf{b}}$		✓

Table 5.4: Table of inner a_{in} and outer a_{out} activation functions to be tested with weight vector $\mathbf{w}$, bias vector $\mathbf{b}$ and layer vector $\mathbf{x}$.

Height and depth analysis

The appropriate size of a neural network in terms of height H and depth D was found by finding a suitable trade-off between under and over fitting the model parameter space. The design complexity of continuous beam systems will likely be reflected in deeper and wider neural networks than those considered in previous literature [99] due to the large number of load arrangements that may be critical, the numerous design criteria that govern the design, and the variety of viable cross-sections. For this reason, a wide range of heights and depths were tested. The size of the networks were denoted by a simple syntax based on the architecture of the hidden layers. For example, "50-50-50" refers to a neural network with three hidden layers with 50 nodes each.

Other neural network parameters and hyperparameters

Given the large dataset size and computational resources required for training, a simple hold-out strategy was deemed appropriate as opposed to other validation strategies [520], and hence the final dataset was randomised and split into training, validation and testing sets using a 70%, 15%, 15% split, respectively. The testing set was only used once after an appropriate neural network architecture was found experimentally. A robustness

check with various initialiser seeds was carried out on the final architecture. Other neural network training aspects, such as optimisers, types of initialisers, learning rates and batch-sizes were chosen empirically based on MAPE performance and qualitative comparison of learning behaviour. The options/ranges for each of these are summarised in Table 5.5. All stochastic elements were controlled through explicit initialiser seeds.

Neural network training aspect	Options/range considered	Selected
Optimiser	SGD, RMSprop, Adam, Nadam	Nadam
Learning rate	$\alpha = [0.0001, 1.0000]$	$\alpha = 0.0005$
Initialiser	Gaussian with $\mu = 0$, $\sigma = [0.00, 1.00]$	$\sigma = 0.05$
Batch-size	$[128, 8192]$	1024

Table 5.5: Options and/or ranges of neural network learning parameters and hyperparameters tested, along with selected parameters for all training runs presented in results.

5.3.6 Summary of methodology

The following procedure was adopted to develop the machine learned inverse operator:

1. Evaluate the maximum influence zone $k_{\max}$ for the continuous beam system using the procedure from Chapter 4 based on the design constraints specified in Tables 5.1 and 5.2.

2. Design continuous beam systems of size $m = 2k_{\max} + 1$ using a coupled analysis and design approach [465] with a target utilisation ratio $u_{\text{target}} = 0.99$. Each beam within the continuous beam system will correspond with one data point, with zero-padding for edge or near-edge beams as shown in Figure 5.5. Finally, split and normalise the data into a training, validation and testing set as explained in Section 5.3.5.

3. Develop the neural network model using the following steps:

    3.1. Assume a standard 50-50 architecture and test out the various combinations of loss and activation functions as identified Table 5.3 and Table 5.4, respectively, based on 100k training data points.

    3.2. Test various height H and depth D variations as explained in Section 5.3.5 based on 100k training data points.

    3.3. For the best architecture (height, depth and activation function), test the performance against different training set sizes.

4. Evaluate the performance of the final neural network against the testing dataset and conduct a robustness test using various initialiser seeds for the weights and biases.

5.4 Model development and results

5.4.1 Influence zone size estimation

The maximum influence zone $k_{\max}$ of continuous beam systems subject to design constraints specified by Tables 5.1 and 5.2 was established. Using the procedure from Chapter 4, 25 random UDL and span distributions were generated for a $m = 17$ sized system and designed against ULS checks from EN 1993-1-1 [466] using the custom I-sections specified in Table 5.2 with a target utilisation ratio $u_{\text{target}} = 0.99$. This led to the creation of 10,625 continuous beams ($25 \times 25 \times 17$). Each beam's influence zone value was evaluated using an error threshold of $\epsilon_{\max} = 0.02$. This threshold was selected based on the expected MAPE performance achievable with the multi-layered neural network, with the results shown in Figure 5.7. The results indicate that the average and maximum influence zone size is $k_{\max} = 1.75$ and $k_{\max} = 5$, respectively. This suggests the system size required for the dataset generation is $m = 2k_{\max} + 1 = 11$, and the required input layer size is $4k_{\max} + 2 = 22$. This influence zone evaluation took 5 hours of computation time.

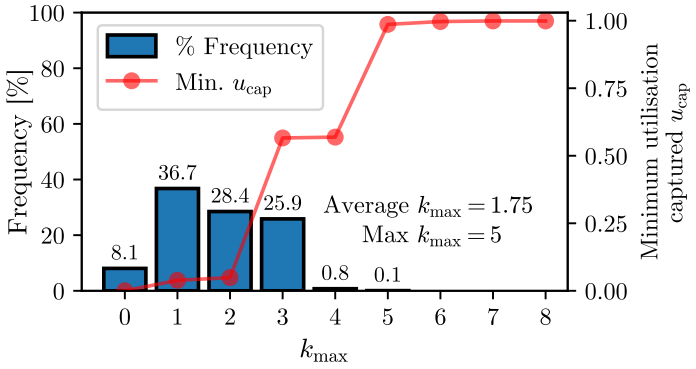


Figure 5.7: Influence zone results for a $m = 17$ system with $\epsilon_{\max} = 0.02$ based on the design constraints established by Tables 5.1 and 5.2 using the methodology from Chapter 4.

5.4.2 Data generation, visualisation and pre-processing

Drawing from uniform distributions for spans and UDL values identified in Table 5.1, two datasets were created. The first consists out of 266 unique UDL $\boldsymbol{\omega}$ and span $\mathbf{L}$ permutations of $m = 11$ sized continuous beam systems, and the second out of 251 unique permutations. A coupled analysis and design optimisation approach [465] with a target utilisation ratio $u_{\text{target}} = 0.99$ based on ULS cross-section checks [466] and all critical load arrangements [441] was implemented to find the appropriate custom I-section from Table 5.2 for each beam within the system. This process resulted in 1,471,327 individual data-points ($11 \times (266^2 + 251^2)$) that took 3.5 days to generate.

The distribution of utilisation ratios achieved for the specified target utilisation ratio $u_{\text{target}} = 0.99$ are shown in Figure 5.8a). Since the design space is limited to discretized cross-section properties the utilisation target ratio $u_{\text{target}} = 0.99$ was rarely met exactly.

Therefore, a sub-selection of this dataset took place, discarding all of the beams that fell outside of utilisation ratio range $0.97 \leq u < 1.00$.

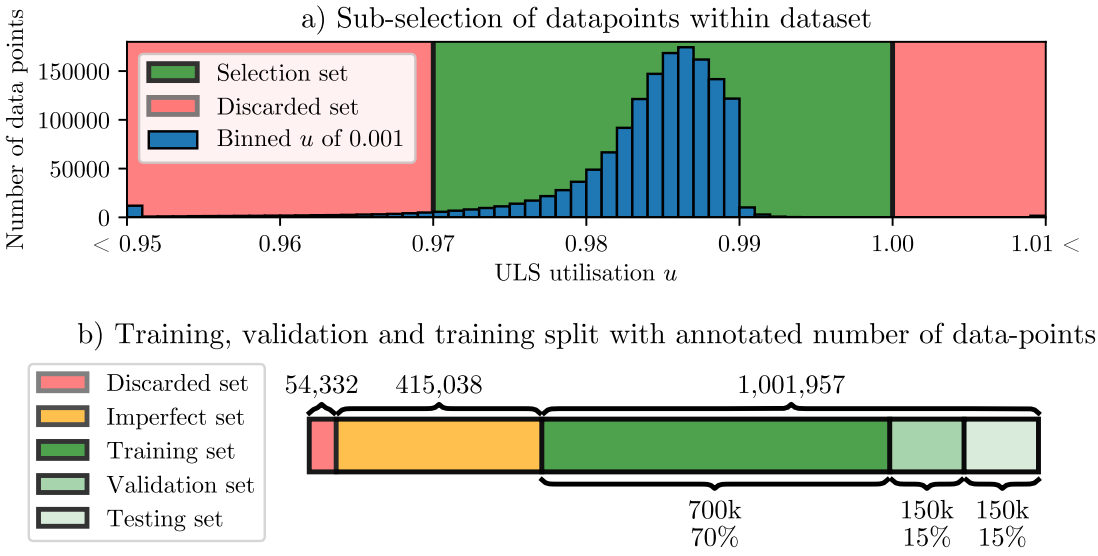


Figure 5.8: Sub-selection of data points from the initial 1,471,327 dataset based on a ULS utilisation ratio u range of 0.97-1.00, with values collated in bins of size 0.001. The 1000k CBeamXP dataset was drawn from the green sets.

Although only 54,322 data-points belonged to the discarded set, the dataset was further stripped of all data points which had beam members within their system that belonged to the discarded set, even if those beams themselves fell within the selected utilisation ratio range. These data points are defined as the "Imperfect set" in Figure 5.8a). This process removed another 415,038 data-points. This left 1,001,957 data points (1,471,327-415,038-54,322), each representing a beam within a $m = 11$ member system and the surrounding design information from the influence zone of valid structural designs under ULS conditions. This set was randomised, further stripped of another 1957 data points, to yield a dataset size of exactly 1 million (1000k).

This 1000k dataset was named CBeamXP: Continuous Beam Cross-section Predictors [6] and represents ULS compliant beam systems of system size $m = 11$ with utilisation ratios between $0.97 \leq u < 1.00$. The CBeamXP dataset was split into a training, validation and testing set using a 70%, 15%, 15% split as shown in Figure 5.8b). Histograms of the spans, UDLs, utilisation ratios and cross-section indices (corresponding to one of the 1000 custom I-sections in ascending stiffness order) are shown in Figure 5.9. For pre-processing, all inputs and outputs were divided by the maximum value within the training and validation set.

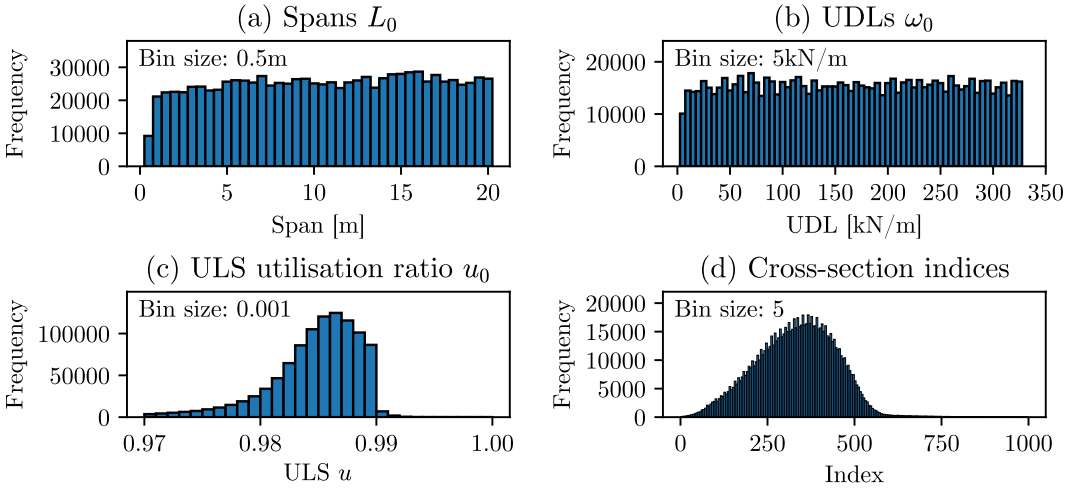


Figure 5.9: Frequency distributions for various descriptor variables of the CBeamXP dataset. Spans and UDL values are uniformly distributed, whilst the selected cross-section indices of the optimised beam systems follow a normal distribution.

5.4.3 Network architecture development results

Loss and activation function variations

The neural network development began by evaluating the MAPE performance of 50-50 architectures for different loss functions J, inner a_{in} and outer a_{out} activation functions. These networks were trained for 1000 epochs using 100k datapoints from the training set, yet validated against the entire 150k validation set. The total training time was 5 hours with results shown in Table 5.6.

	J_{MAE}			J_{MSE}		
	$a_{\text{out,ReLU}}$	$a_{\text{out,sigm}}$	$a_{\text{out,exp}}$	$a_{\text{out,ReLU}}$	$a_{\text{out,sigm}}$	$a_{\text{out,exp}}$
$a_{\text{in,ReLU}}$	0.105	0.164	0.143	0.116	0.168	0.146
$a_{\text{in,sigm}}$	0.558	0.158	0.168	0.568	0.250	0.243
$a_{\text{in,tanh}}$	0.205	0.143	0.133	0.274	0.160	0.147
	J_{MAPE}			J_{MSPE}		
	$a_{\text{out,ReLU}}$	$a_{\text{out,sigm}}$	$a_{\text{out,exp}}$	$a_{\text{out,ReLU}}$	$a_{\text{out,sigm}}$	$a_{\text{out,exp}}$
$a_{\text{in,ReLU}}$	**0.072**	**0.090**	**0.091**	**0.089**	**0.093**	**0.089**
$a_{\text{in,sigm}}$	0.707	0.129	0.124	0.707	0.147	0.137
$a_{\text{in,tanh}}$	0.205	**0.083**	**0.083**	0.235	**0.082**	**0.087**

Table 5.6: Validation MAPE metrics at epoch 1000 for different combinations of loss J, inner a_{in} and outer a_{out} activation functions for a 50-50 architecture using 100k training and 150k validation data points. MAPE values of less than 0.100 (10%) are in bold.

The results clearly indicate that the percentage-based loss functions J_{MAPE} and J_{MSPE} typically outperform their non-percentage-based counter-parts. Performance between ei-

ther J_{MAPE} and J_{MSPE} was relatively similar, with all MAPE values of less than 10% (0.100) bolded. J_{MAPE} was chosen as the loss function for this investigation due to it being more commonly used. The five best inner and outer activation function combinations in Table 5.6 under J_{MAPE} (bolded values) were subsequently qualitatively analysed.

This qualitative analysis highlighted that using ReLU as an outer activation function allows the prediction of null cross-section properties. This is an invalid prediction since the model operates on the basis that a beam with some minimum cross-section properties must exist in the context of this structural system. A similar limitation applied for the sigmoid activation function which asymptotically approaches the value of one at positive infinity. This limits the network's ability of predicting cross-sections larger than those found within the training and validation dataset. For these reasons, the $a_{\text{out,exp}}$ function was selected for this particular network architecture since $a_{\text{out,exp}}$ does not result in zero-valued cross-section properties and also does not impose an upper limit on the outputs. From the remaining viable networks, the $a_{\text{in,ReLU}}$ and $a_{\text{out,exp}}$ architecture converged the quickest and was therefore chosen for further development in this study.

Height and depth variations

The architecture of the hidden layers (height H and depth D) needs to be sufficiently expressive to reflect the design complexity of continuous beam systems, and need to avoid under- and over-fitting the model. Therefore, comparison between training and validation performance is needed. The 100k training and 150k validation sets from section 5.4.3 were re-used for this purpose. Figure 5.10 compares the performance of various networks containing two hidden layers of varying heights at epoch 1000. The combined training time of these networks was 22 hours. The 600-600 network was identified as the point at which the performance transitioned from under-fitting to slight over-fitting. Figures 5.10b) and c) further indicate the accuracy profiles for both training and validation, respectively. Note that the maximum validation accuracy values greatly exceed the value of at least 19 for all networks, regardless of height, meaning the network predicted cross-sectional properties 19 times larger than the target value.

The "optimal" number of hidden layers for height 600 was investigated, with results shown in Figure 5.11. This resulted in a combined training time of 12 hours. Networks with more than three hidden layers showed no major improvements in either training or validation performance except in minimum training accuracy. However, this was not associated with an improvement in minimum validation accuracy as shown in Figure 5.11c). For these reasons, a depth of three hidden layers was deemed appropriate.

Dataset size variations

Figure 5.12 shows the change in performance as a function of the training dataset size, from 25k to 700k data points, with the same 150k validation dataset as in the previous sections. The combined training time was 1.5 days. Except for slight variations in the minimum and maximum accuracy values, the performance of the neural network naturally improved with a larger training set.

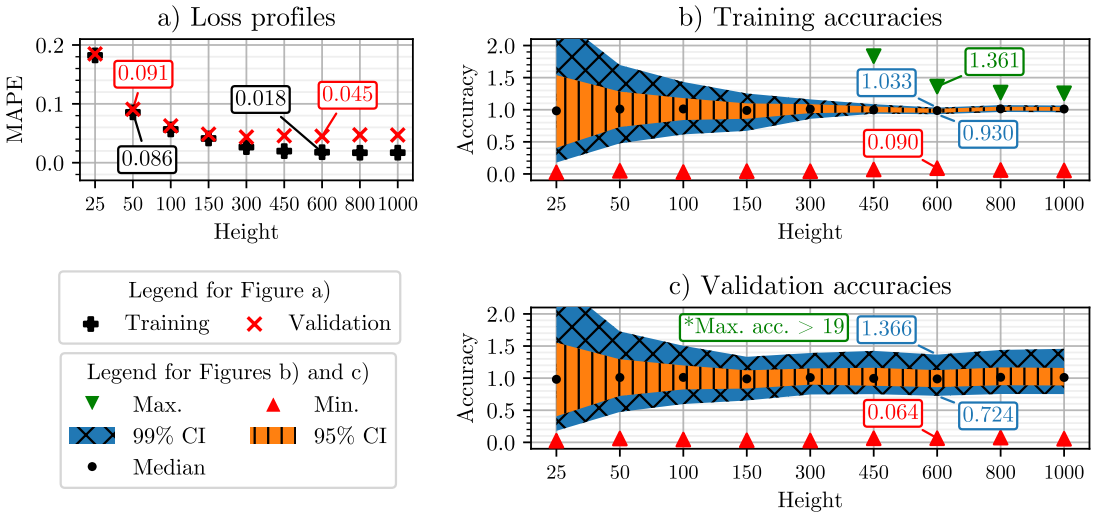


Figure 5.10: Loss and accuracy profiles for $a_{\text{in,ReLU}}$ and $a_{\text{out,exp}}$ networks at epoch 1000 with J_{MAPE} with two hidden layers of equal height. Training set size of 100k and validation set size of 150k.

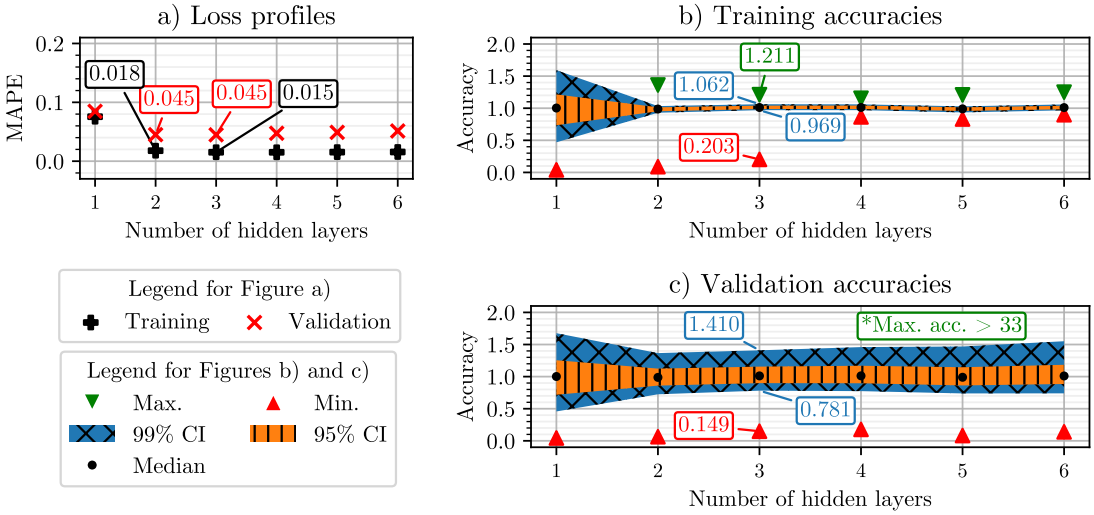


Figure 5.11: Loss and accuracy profiles for $a_{\text{in,ReLU}}$ and $a_{\text{out,exp}}$ networks at epoch 1000 with J_{MAPE} with hidden layers of height $H = 600$. Training set size of 100k and validation set size of 150k.

5.4.4 Model performance: testing and robustness

The final neural network model consists of a 600-600-600 architecture with $a_{\text{in,ReLU}}$ and $a_{\text{out,exp}}$ activation functions trained using the J_{MAPE} loss function based on a training and validation dataset size of 700k and 150k data points, respectively, with learning graphs shown in Figure 5.13. The neural network at epoch 1000 was also evaluated against the testing set created in Figure 5.8, and checked for robustness by re-training the same

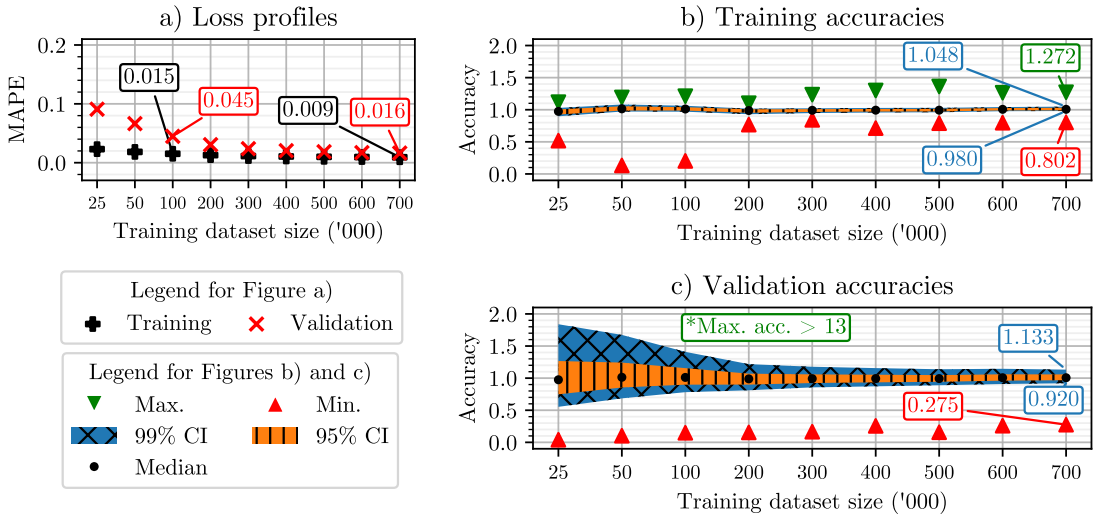


Figure 5.12: Loss and accuracy profiles for 600-600-600 $a_{\text{in,ReLU}}$ and $a_{\text{out,exp}}$ network at epoch 1000 with J_{MAPE} for various training dataset sizes, with a validation set size of 150k.

network a further 9 times using different kernel initialiser seeds, which took 4.5 days. The general model performance results and standard deviations $\sigma_{\text{initialiser}}$ due to these different initialiser seeds is summarised in Table 5.7.

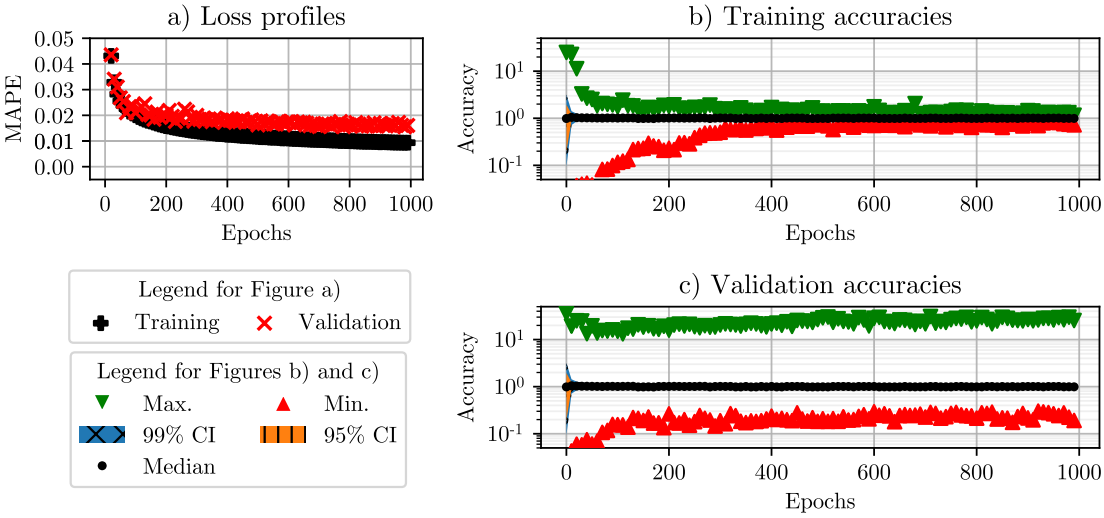


Figure 5.13: Loss and accuracy profiles for 600-600-600 $a_{\text{in,ReLU}}$ and $a_{\text{out,exp}}$ network at epoch 1000 with J_{MAPE}, 700k training and 150k validation sets. Note use of logarithmic y-axis to show the full range of maximum validation accuracies.

The similar performance between the testing and validation set in Table 5.7 strongly suggests that the model is likely to generalise well to new data-points. The impact of changing initialiser seed is minimal except for the minimum and maximum accuracy val-

Dataset	Data Type	MAPE	Accuracy Percentiles						
			Min	0.5%	2.5%	Median	97.5%	99.5%	Max
Training	Results	0.009	0.802	0.980	0.990	1.007	1.030	1.048	1.272
	$\sigma_{\text{initialiser}}$	0.000	0.161	0.013	0.010	0.007	0.010	0.014	0.138
Validation	Results	0.016	0.275	0.920	0.968	1.006	1.058	1.133	26.811
	$\sigma_{\text{initialiser}}$	0.001	0.058	0.014	0.010	0.007	0.012	0.018	4.977
Testing	Results	0.016	0.313	0.917	0.967	1.006	1.058	1.138	12.282
	$\sigma_{\text{initialiser}}$	0.001	0.070	0.012	0.010	0.007	0.012	0.018	2.189

Table 5.7: Loss and accuracy profiles for 600-600-600 $a_{\text{in,ReLU}}$ and $a_{\text{out,exp}}$ network at epoch 1000 with J_{MAPE}, 700k training, 150k validation sets and 150k testing set.

ues.

5.5 Discussion

5.5.1 Model generalisability

One of the fundamental objectives of this work was to develop a machine learned structural design model capable of generalising beyond the system size $m = 11$ it was trained on. To achieve this, the influence zone concept was leveraged with zero-padding to theoretically allow the neural network to make localised predictions for continuous beams of arbitrary system size m. To test this, over 1000 additional testing data-points were generated using the same methodology as described in Section 5.3.4 and sub-selection process as shown in Section 5.4.2 for each system size $1 \leq m \leq 20$ (including $m = 11$). The MAPE and accuracy performance are shown in Figure 5.14.

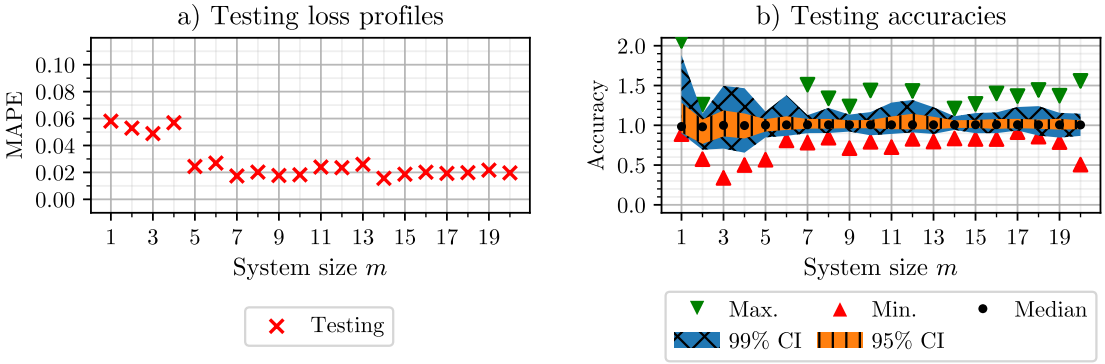


Figure 5.14: Generalisability performance of the final 600-600-600 neural network in terms of MAPE and accuracy for unseen structures of varying system size m.

Figure 5.14 indicates that the machine learned inverse operator demonstrates strong generalisation capability for continuous beam system sizes $m \geq 5$ with MAPE $\approx 2\%$. System sizes $m < 5$ saw a slightly deteriorating MAPE values of $5\% - 6\%$. These are

encouraging results given that the neural network was never trained on system sizes less or greater than $m = 11$. The greatest variations in performance were typically in the maximum and minimum accuracy values; in fact the model performed often better in terms of maximum performance for system size other than $m = 11$.

These results provide merit to the novel implementation of the influence zone concept from Chapter 4 as a mechanics driven feature selector and using zero-padding to build machine learned inverse operators capable of generalising to differently sized continuous structural systems. This opens the possibility of investigating the applicability of this methodology for two or three dimensional frames. Furthermore, these results also provide a solution to the limitation of fixed-dimensional input vectors of multi-layer neural networks [101].

In recent years, other researchers have investigated the development of generalisable machine learning models; most of these efforts have focused on machine learned forward operators [101, 507, 521]. Within the realm of structural design inverse operators, researchers have noted that the question of generalisability remains typically under-investigated [100]. Whilst previous works studied the ability of neural networks to generalise under different boundary conditions [503, 504], this work distinguishes itself on generalising across differently sized systems. Combining the underlying techniques behind these studies may allow one to train a generalisable model of arbitrary size and arbitrary boundary conditions.

5.5.2 Performance variability

This investigation also differentiated itself from previous works by measuring the variability of predictions in terms of accuracies. Notably, this allowed one to identify the range of over- and under-predictions, which are not captured by average loss function metrics such as MAE or MAPE. Despite gradual improvement within the 95% and 99% confidence intervals, the final performance graph in Figure 5.13 indicates that the confidence intervals of the validation set lag those of the training set. The same can also be said for the testing set, especially for maximum and minimum accuracies as shown in Table 5.7.

To identify potential causes of this divergence in performance between the training and testing set, custom box plots of testing accuracies were generated for a number of variables that describe the dataset, ordered based on ascending deciles (D_0 to D_{10}). By evaluating the standard deviation of each decile's accuracy values, and taking the standard deviation of those standard deviations $\sigma(\sigma_{\text{Deciles}})$, one can quantify numerically which variable causes the greatest dispersion of the accuracy values. These results are shown in Figure 5.15.

By studying Figure 5.15 in detail, it was identified that the total load variable $\omega_0 \times L_0$ caused the greatest $\sigma(\sigma_{\text{Deciles}})$ dispersion as seen in Figure 5.15f). Figure 5.15f) also showed the most identifiable demarcation between low and high accuracy results. The prediction variability of cross-section properties of a beam is the worst when the combined product of both the UDL load ω_0 and span L_0 fell in the lowest Decile ($< D_1$). This pattern can also be identified by studying heat-maps of the average and maximum MAPE performance which occurred at each ω_0 and span L_0 combination within the dataset as shown in Figure 5.16.

Using structural engineering intuition, one infers that the design of short and lightly loaded spans is more likely to be influenced by the UDLs of the surrounding members

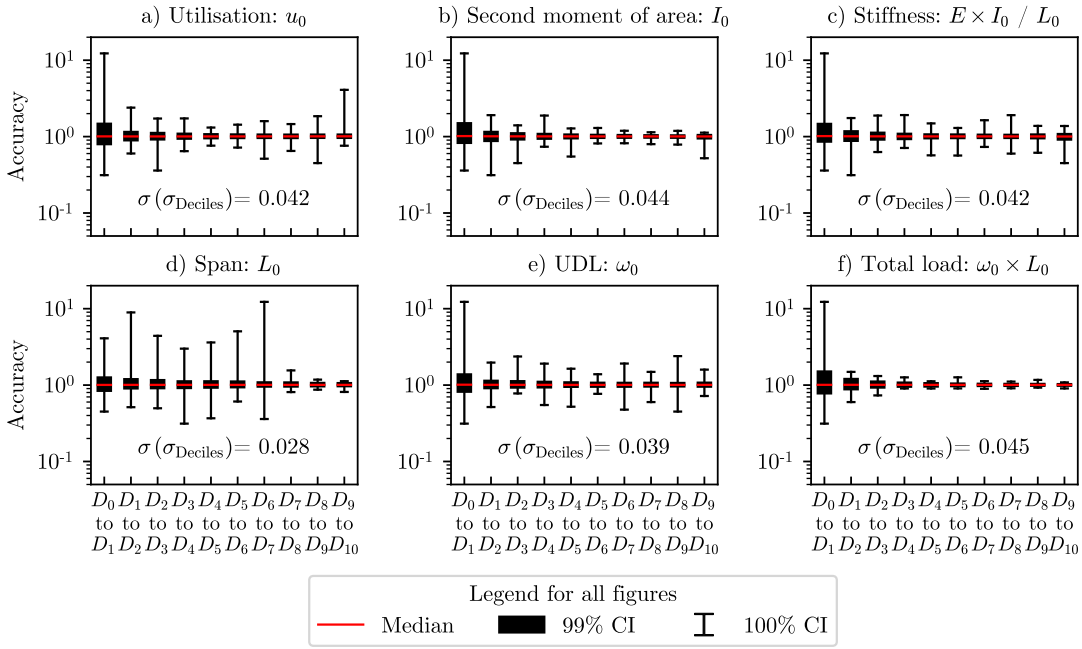


Figure 5.15: Custom box plots of accuracies vs. variables in binned deciles. Total load f) correlates with the greatest dispersion $\sigma(\sigma_{\text{Deciles}}) = 0.045$, which can also be identified visually.

within a continuous system. Whilst the influence zone concept ensures the pertinent design information is contained within the inputs, providing that information solely in the form of an input vector may not be sufficient to make accurate predictions under all circumstances. The fact that there were also wide prediction variabilities for the smallest deciles for both the second-moment of area and stiffness values as shown in Figures 5.15b) and c) suggests that exposing the machine learning model to additional physics knowledge (other than influence zones) of the structural system may lead to further improvements.

In one of the early studies, Berke et al. [501] noted that despite achieving relatively low prediction errors on average, neural network predictions can occasionally vary significantly. In more recent works, the presence of large error predictions (over >40%) were noted and manually removed from the final reported average prediction error metric [503]. The results from this study highlight that this variability issue needs to be further addressed. So far, the authors have identified only a single study that investigated error variability when evaluating machine learning performance for civil and structural engineering applications [99]. The use of the accuracy metric along with its minumum, maximum, 95% and 99% metrics could provide a framework to study error variability in more detail.

5.5.3 Other neural network performance observations

Despite using a simple multi-layer neural network for the structural design inverse operator $O^{\dagger}_{\text{inv}}$, the development procedure successfully lowered the validation error from MAPE values of $\approx$ 10% in Table 5.6 to 1.6% in Table 5.7, a performance that was matched

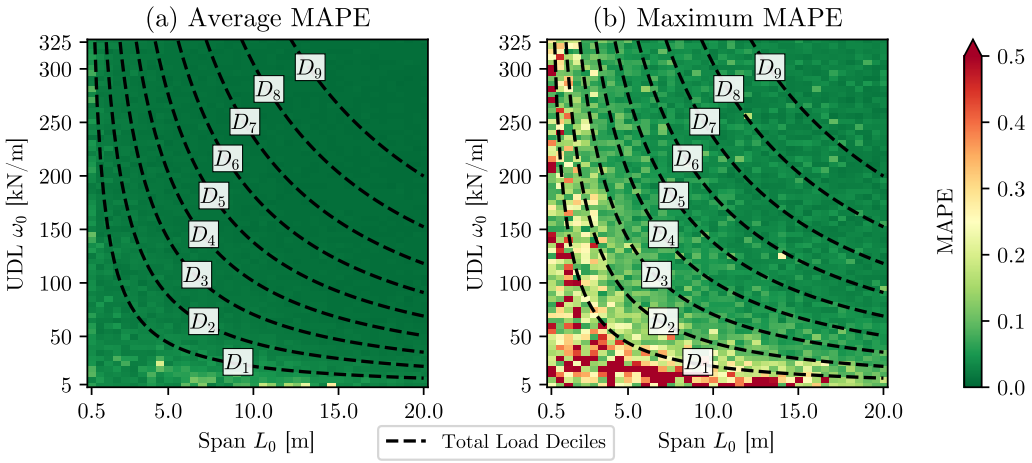


Figure 5.16: Total load heatmaps which evaluates the a) average MAPE and b) maximum MAPE values for all UDL ω_0 and span L_0 combinations. Notice how maximum MAPE errors occur at low total load combinations ($\omega_0 \times L_0$) within the first decile up to D_1.

by the testing dataset as well. This was attributable to numerous factors, a notable one being the use of the percentage based loss function, which as anticipated in Section 5.3.5, was more suitable for the dataset given the orders of magnitude differences in the targets. The use of the exponential output activation function a_{exp} may also have positively contributed to dealing with target values that vary greatly in magnitude.

The lack of literature on machine learned structural design models for continuous beam systems means that a direct comparison of the 1.6% MAPE performance is not possible at present. However, one can compare this performance with performance metrics of structural design models developed for different applications. For example, the network developed in this work outperformed previous multi-layer neural network regression models; an early concrete beam prediction model achieved a MAPE value of 10.17% [97], whilst a cross-section predictor of aerospace components averaged out at a MAPE value of 5% [501]. The network presented in this study also performed well when compared to more advanced network architectures such as convolutional neural networks for topologically optimised truss structures that achieved voxel value errors of 5.63% [503]. Comparison with further works that developed machine learned structural design models was not possible for studies which reported performance with non-percentage based metrics such as MAE [98] or MSE [99, 100, 513].

This study also differentiates itself by the quantity of data it was trained on (up to 700,000 data-points), which based on Figure 5.12 helped improve validation performance. Early works from the 1990s had training set sizes smaller than 100 data-points [97, 98, 501], and even more recent literature only trained using 600 [99], 12,000 [100] 28,000 [503] or just under 40,000 [504] data-points. Whilst large datasets significantly increase computational cost, the combination of big data and more advanced neural network architectures may improve performance further, both in terms of average error and prediction variability.

5.5.4 Limitations and scope for future works

There are multiple limitations that restrict practical use of the proposed design model. The first is the fact that the structural systems within the dataset were designed against ULS constraints only, and made other assumptions on the nature of the design problem listed in Section 5.2.2. The generalisability of the model, specifically for system sizes $m < 5$ also requires further work, and the issue of prediction variability will also require additional investigation in terms of either model architectures or generating larger datasets. Furthermore, there likely exist a wide range of mathematical techniques from inverse problems that could aid in developing and assessing operators for structural design problems, by for example estimating the Lipschitz coefficient of the mapping. These limitations provide a clear basis for further works in the future.

On another note, Table 5.8 summarises the total computation time required for the entirety of the results section. Whilst the computation time could have been accelerated through parallelisation, improved computation resources and simplification of metric evaluations algorithms, the purpose of Table 6.9 is to indicate the relative proportion of time spent at each stage. Greater computational resources may allow investigations using alternative validation strategies such as k-fold cross-validation [520] and automated hyperparameter selection procedures [522] that could result in improved performance.

Section	Stage	Computation time		Proportion [%]
		In hours [h]	In days [D]	
Section 5.4.1	Influence zone evaluation	5	0.2	33
Section 5.4.2	Data generation and pre-processing	84	3.5	
Section 5.4.3	Loss and activation function study	5	0.2	27
Section 5.4.3	Height and depth study	34	1.4	
Section 5.4.3	Dataset size study	36	1.5	
Section 5.4.4	Testing and robustness study	108	4.5	40
Total computation time:		272	11.3	100

Table 5.8: Computation time for each neural network development stage.

A significant portion of the computation effort was spent simply generating the datapoints for training, validation and testing. In light of encouraging reproducibility studies [62] and to encourage research that improves the predictive capability of the machine learned structural design model presented here, the CBeamXP dataset along with an associated python-based neural network training script are made available at an open-source data repository [6].

5.6 Conclusions

This work developed a new neural network based structural design model to predict cross-section property requirements of continuous beam systems non-iteratively. The major contributions of this investigation include:

- Framing structural design as an inverse problem, and using this novel perspective to

identify three distinct types of machine learning applications. One of these types, machine learned inverse operators, were investigated in this work to develop a non-iterative structural design model. This presents a fundamental shift from traditional design approaches.

- Developing a non-iterative structural design model for continuous beam systems of arbitrary member size through the novel use of influence zones from Chapter 4 to provide a mechanics-driven feature selection process that enhanced the model's generalisability.
- Achieving a mean absolute percentage error of 1.6% which was lower than machine learned structural design models from comparative literature [97, 501, 503]. This performance was attributable to the careful consideration of the network architecture in terms of height and depth of the hidden layers, the selection of loss and activation functions that were appropriate to address the challenges posed by continuous beam system, and a dataset size of 700,000 data points.
- Identifying the importance of measuring and reducing prediction error variability. In this study the 99% confidence interval for testing accuracy was between 91.7% and 113.8%. Reducing prediction variability is a significant knowledge gap in literature, especially in regards to machine learning applications within safety critical systems such as structural design.

The CBeamXP dataset generated in this work containing one million data-points along with an associated python-based neural network training script were published at an open-source data repository [6]. Aside from allowing results to be reproduced, sharing this data will hopefully encourage future research towards machine learned structural design models that improve the mean absolute percentage error, generalisability, or prediction variability achieved in this investigation.

Data statement

The dataset generated and used in this work as well as a neural network training script are available at an open-source data repository [6].

Appendix

5.A Neural network training script

```
"""
This python script uses the CBeamXP dataset to build and train a
neural network structural design model. It can also be used to
replicate the results from the Gallet et al. (2024) research article
"Machine learning for structural design models of continuous beam
systems via influence zones" (doi.org/x).

This script takes three inputs:
SIZE_TRAIN: Defines the size of the training dataset,
needs to be an integer between 1 to 700,000.
EPOCHS: Defines the number of epochs to train the neural network for.
MODEL_TYPE: By default creates a user defined neural network. The
architecture can be defined in the
step3_generate_NN_model() function. One can also select
pre-built options 'Gallet-50-50' and 'Gallet-600-600-600'.

This script directly compares the results of any neural network
against those evaluated in the Gallet et al. (2023) research article.
"""


# --- Set inputs
# Creates a user defined architecture, defined in the
    step3_generate_NN_model() function
SIZE_TRAIN=1*10**5; EPOCHS=5; MODEL_TYPE='user_defined_model'

# To replicate 'Gallet_50_50' results:
# SIZE_TRAIN=1*10**5; EPOCHS=1000; MODEL_TYPE='Gallet_50_50'

# To replicate 'Gallet_600_600_600' results:
# SIZE_TRAIN=7*10**5; EPOCHS=1000; MODEL_TYPE='Gallet_600_600_600'


# --- Import modules
import os
import math
import pickle
import pandas as pd
import numpy as np

import tensorflow as tf
import tensorflow.keras as keras
import tensorflow_probability as tfp
import tensorflow.math as tfm

import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
from matplotlib.patches import Patch


# --- Main script
```

```
def main():
    valid_config = step1_set_global_settings_and_check_inputs(SIZE_TRAIN, EPOCHS, MODEL_TYPE)

    if valid_config:
        datasets = step2_extract_and_preprocess_datasets(size_train=SIZE_TRAIN)
        init_model = step3_generate_NN_model(model_type=MODEL_TYPE)
        trained_model, MAPE, accuracy = step4_train_NN_model(init_model, datasets, epochs=EPOCHS)
        step5_compare_results_with_Gallet_et_al_2023(trained_model, MAPE, accuracy)
        step6_save_model(trained_model, MAPE, accuracy)
        step7_plot_training_results(MAPE, accuracy)

# --- Define Mean Absolute Percentage Error (MAPE) loss function
class lossMAPE(keras.losses.Loss):
    def __init__(self, name='MAPE', eps=10.**-8):
        super().__init__(name=name)
        self.eps = eps

    def call(self, y_true, y_pred):
        return tfm.abs((y_true-y_pred) / (y_true + self.eps))

    def get_config(self):
        return {}

    @classmethod
    def from_config(cls, config):
        return cls(**config)


# --- Configures global settings and checks inputs
def step1_set_global_settings_and_check_inputs(SIZE_TRAIN, EPOCHS, MODEL_TYPE):
    # To replicate results from Gallet et al. (2023), the following commands are required
    keras.mixed_precision.set_global_policy('float32') # Use <Policy "float32">
    tf.config.set_visible_devices([], 'GPU') # Force CPU training, GPU training may differ
    tf.random.set_seed(0) # Set global random tensorflow seed to zero

    # Disables debugging logs; script output more easily interpreted
    os.environ['TF_CPP_MIN_LOG_LEVEL'] = '1' # tensorflow INFO messages are not printed
    #pd.set_option('display.max_columns', None)

    # Simple operation to check device name
    test_op = tf.constant(0.0)

    # Report progress
    print('\n--- Step 1: Set global settings and check inputs ---')
    print('Mixed precision policy: {}'.format(keras.mixed_precision.global_policy()))
    print('Training on CPU: {}'.format('CPU' in test_op.device and 'GPU' not in test_op.device))
```

```
97
98      # Check that inputs are valid
99      valid_config = True
100     if (MODEL_TYPE not in ['Gallet_50_50',
101                            'Gallet_600_600_600',
102                            'user_defined_model']):
103         valid_config = False
104         print('Invalid MODEL_TYPE')
105     elif EPOCHS < 1 or type(EPOCHS) is not int:
106         valid_config = False
107         print('Invalid EPOCHS')
108     elif SIZE_TRAIN < 1 or SIZE_TRAIN > 700000 or type(SIZE_TRAIN) is not
        int:
109         valid_config = False
110         print('Invalid SIZE_TRAIN')
111
112     if valid_config == True:
113         print('Inputs: Valid \n')
114
115     return valid_config
116
117
118 # --- Extracts and pre-processes datasets
119 def step2_extract_and_preprocess_datasets(size_train=100000,
120                                           size_val=150000,
121                                           size_test=150000):
122     # Load and pre-process dataset
123     dataset_file_name = 'CBeamXP_dataset.csv'
124     size_train = size_train # Max points available: 700,000
125     size_val = size_val # Max points available: 150,000
126     size_test = size_test # Max points available: 150,000
127
128     # Column names of features and targets
129     columns_features = ['UDL_-5_[kN/m]', 'UDL_-4_[kN/m]', 'UDL_-3_[kN/m]',
130     'UDL_-2_[kN/m]', 'UDL_-1_[kN/m]', 'UDL_0_[kN/m]',
131     'UDL_1_[kN/m]', 'UDL_2_[kN/m]', 'UDL_3_[kN/m]',
132     'UDL_4_[kN/m]', 'UDL_5_[kN/m]', 'span_-5_[m]',
133     'span_-4_[m]', 'span_-3_[m]', 'span_-2_[m]',
134     'span_-1_[m]', 'span_0_[m]', 'span_1_[m]',
135     'span_2_[m]', 'span_3_[m]', 'span_4_[m]',
136     'span_5_[m]']
137     columns_targets = ['I_[cm4]', 'Az_[cm2]', 'Wpl_[cm3]']
138
139     # Load entire dataset into a dataframe
140     dataset_all = pd.read_csv(dataset_file_name, delimiter=',')
141
142     # Split dataset based on type: training, validation, testing
143     dataset_train = dataset_all[dataset_all['dataset_type'].values == '
        training']
144     dataset_val = dataset_all[dataset_all['dataset_type'].values == '
        validation']
145     dataset_test = dataset_all[dataset_all['dataset_type'].values == '
        testing']
146
147     # Split each dataset into inputs (x) and outputs (y) for training
148     x_train = dataset_train[columns_features].iloc[:size_train]
149     y_train = dataset_train[columns_targets].iloc[:size_train]
```

```
    x_val = dataset_val[columns_features].iloc[:size_val]
    y_val = dataset_val[columns_targets].iloc[:size_val]
    x_test = dataset_test[columns_features].iloc[:size_test]
    y_test = dataset_test[columns_targets].iloc[:size_test]

    # Extract maximum values for each column from
    # the combined training and validation datasets
    max_features = pd.concat([x_train, x_val]).max()
    max_targets = pd.concat([y_train, y_val]).max()

    # Standardise all data-points between 0-1
    # by dividing all features by their maximums
    x_train = x_train / max_features; y_train = y_train / max_targets
    x_val = x_val / max_features; y_val = y_val / max_targets
    x_test = x_test / max_features; y_test = y_test / max_targets

    # Report progress
    tmp_str = ('--- Step 2: Dataset loaded, split and pre-pocessed ---\n'
               'Total datapoints available: {:,}\n'
               'Max training dataset size: 700,000\n'
               'Training dataset size: {:,} \n'
               'Validation dataset size: {:,} \n'
               'Testing dataset size: {:,} \n'
               'Number of features: {} \n'
               'Number of targets: {} \n')

    print(tmp_str.format(dataset_all.shape[0], x_train.shape[0], x_val.
    shape[0],
    x_test.shape[0], x_train.shape[1], y_train.shape[1]))

    return (x_train, y_train, x_val, y_val, x_test, y_test)


# --- Generate neural network (NN) model
def step3_generate_NN_model(model_type='Gallet_50_50'):
    # Define kernel (weights and biases) initialiser with explicit seeds
    def return_init(seed):
        return keras.initializers.RandomNormal(mean=0.0, stddev=0.05, seed=
    seed)

    # Generate model using Sequential model API
    # Model type can be based on Gallet et al. (2023) or user defined
    architectures
    if model_type == 'Gallet_50_50':
        init_model = keras.models.Sequential(name='Gallet_50_50')
        init_model.add(keras.layers.Dense(50, activation='relu',
    input_shape=(22,), kernel_initializer=return_init(0)))
        init_model.add(keras.layers.Dense(50, activation='relu',
    kernel_initializer=return_init(1)))
        init_model.add(keras.layers.Dense(3, activation='exponential',
    kernel_initializer=return_init(1)))
    elif model_type == 'Gallet_600_600_600':
        init_model = keras.models.Sequential(name='Gallet_600_600_600')
        init_model.add(keras.layers.Dense(600, activation='relu',
    input_shape=(22,), kernel_initializer=return_init(0)))
        init_model.add(keras.layers.Dense(600, activation='relu',
    kernel_initializer=return_init(1)))
```

```
199         init_model.add(keras.layers.Dense(600, activation='relu',
    kernel_initializer=return_init(2)))
200         init_model.add(keras.layers.Dense(3, activation='exponential',
    kernel_initializer=return_init(2)))
201     elif model_type == 'user_defined_model':
202         # ------------ Change this code to test different architectures
    ------------
203         init_model = keras.models.Sequential(name='user_defined_model')
204         init_model.add(keras.layers.Dense(25, activation='relu',
    input_shape=(22,), kernel_initializer=return_init(0)))
205         init_model.add(keras.layers.Dense(3, activation='relu',
    kernel_initializer=return_init(1)))
206         #
    ------------------------------------------------------------------------

207     else:
208         print('Error: MODEL_TYPE not recognised, no init_model created.')
209
210     # Compile initliased model with optimisers, loss function and metrics
211     init_model.compile(keras.optimizers.Nadam(learning_rate=0.0005), loss=
    lossMAPE())
212
213     # Show model summary
214     print('--- Step 3: Create and compile initialised model ---')
215     init_model.summary()
216     print('') # Add new-line after model summary
217
218     return init_model
219
220
221 # --- Train neural network (NN) model
222 def step4_train_NN_model(model, datasets, epochs=10):
223     # Split datasets into individual objects
224     x_train, y_train, x_val, y_val, x_test, y_test = datasets
225
226     # Custom callback to evaluate accuracy percentiles
227     # Accuracy percentiles need to be evaluated on epoch_end, not batch_end
    .
228     # On batch_end evaluation averages percentile values over batches;
    whilst not
229     # an issue for MAPE values, this would significantly distort perenctile
     values.
230     # Hence the need for this custom call-back.
231     class AccuracyPercentilesCallback(keras.callbacks.Callback):
232         def __init__(self, model, x_train, y_train, x_val, y_val):
233             self.model = model
234             self.x_train = x_train; self.y_train = y_train
235             self.x_val = x_val; self.y_val = y_val
236
237             self.history = {'Min': [], '0.5%': [],
238                             '2.5%': [], '50%': [],
239                             '97.5%': [], '99.5%': [],
240                             'Max': [], 'val_Min': [],
241                             'val_0.5%': [], 'val_2.5%': [],
242                             'val_50%': [], 'val_97.5%': [],
243                             'val_99.5%': [], 'val_Max': []}
244
```

```
    def on_epoch_end(self, epoch, logs=None):
        pred_y_train = self.model.predict(self.x_train, batch_size=10**4, verbose=0)
        pred_y_val = self.model.predict(self.x_val, batch_size=10**4, verbose=0)

        acc = np.mean((pred_y_train / self.y_train.to_numpy()),axis=1)
        val_acc = np.mean((pred_y_val / self.y_val.to_numpy()),axis=1)

        accPerc = np.percentile(acc,[0, 0.5, 2.5, 50, 97.5, 99.5, 100])
        val_accPerc = np.percentile(val_acc,[0, 0.5, 2.5, 50, 97.5, 99.5, 100])

        self.history['Min'].append(accPerc[0])
        self.history['0.5%'].append(accPerc[1])
        self.history['2.5%'].append(accPerc[2])
        self.history['50%'].append(accPerc[3])
        self.history['97.5%'].append(accPerc[4])
        self.history['99.5%'].append(accPerc[5])
        self.history['Max'].append(accPerc[6])
        self.history['val_Min'].append(val_accPerc[0])
        self.history['val_0.5%'].append(val_accPerc[1])
        self.history['val_2.5%'].append(val_accPerc[2])
        self.history['val_50%'].append(val_accPerc[3])
        self.history['val_97.5%'].append(val_accPerc[4])
        self.history['val_99.5%'].append(val_accPerc[5])
        self.history['val_Max'].append(val_accPerc[6])

    accuracy_callback = AccuracyPercentilesCallback(model, x_train, y_train, x_val, y_val)

    # Fit the model and evaluate losses
    print('--- Step 4: Train model ---')
    model_history = model.fit(x_train, y_train, validation_data=(x_val, y_val), epochs=epochs, verbose=2, batch_size=1024, shuffle=True, callbacks=accuracy_callback)
    print('') # Force a new-line
    return model, model_history.history, accuracy_callback.history


# --- Compare results with Gallet et al. (2023)
def step5_compare_results_with_Gallet_et_al_2023(trained_model, MAPE, accuracy):
    # Define function to extracts results from losses and accuracy objects
    def extract_results(epoch, MAPE, accuracy):
    decimals = 3
    epoch_results = {
        'Epochs': epoch+1,
        'MAPE Loss': round(MAPE['loss'][epoch], decimals),
        'Val. MAPE Loss': round(MAPE['val_loss'][epoch], decimals),
        'Train Min Acc': round(accuracy['Min'][epoch], decimals),
        'Train 0.5% Acc': round(accuracy['0.5%'][epoch], decimals),
        'Train 2.5% Acc': round(accuracy['2.5%'][epoch], decimals),
        'Train Median Acc': round(accuracy['50%'][epoch], decimals),
        'Train 97.5% Acc': round(accuracy['97.5%'][epoch], decimals),
        'Train 99.5% Acc': round(accuracy['99.5%'][epoch], decimals),
        'Train Max Acc': round(accuracy['Max'][epoch], decimals),
```

```
        'Val. Min Acc': round(accuracy['val_Min'][epoch], decimals),
        'Val. 0.5% Acc': round(accuracy['val_0.5%'][epoch], decimals),
        'Val. 2.5% Acc': round(accuracy['val_2.5%'][epoch], decimals),
        'Val. Median Acc': round(accuracy['val_50%'][epoch], decimals),
        'Val. 97.5% Acc': round(accuracy['val_97.5%'][epoch], decimals),
        'Val. 99.5% Acc': round(accuracy['val_99.5%'][epoch], decimals),
        'Val. Max Acc': round(accuracy['val_Max'][epoch], decimals),
    }
    return epoch_results

    # Load model MAPE file
    with open('saved_models/Gallet_50_50_MAPE', 'rb') as file_MAPE:
        Gallet_50_50_MAPE = pickle.load(file_MAPE)

    # Load model accuracy file
    with open('saved_models/Gallet_50_50_accuracy', 'rb') as file_accuracy:
        Gallet_50_50_accuracy = pickle.load(file_accuracy)

    # Load model with custom objects
    Gallet_50_50_model = keras.models.load_model(
        './saved_models/Gallet_50_50',
        custom_objects= {'lossMAPE': lossMAPE})

    # Load model MAPE file
    with open('saved_models/Gallet_600_600_600_MAPE', 'rb') as file_MAPE:
        Gallet_600_600_600_MAPE = pickle.load(file_MAPE)

    # Load model accuracy file
    with open('saved_models/Gallet_600_600_600_accuracy', 'rb') as
    file_accuracy:
        Gallet_600_600_600_accuracy = pickle.load(file_accuracy)

    # Load model with custom objects
    Gallet_600_600_600_model = keras.models.load_model(
        './saved_models/Gallet_600_600_600',
        custom_objects= {'lossMAPE': lossMAPE})

    # Extract results and save as DataFrame
    user_results = pd.DataFrame.from_dict(extract_results(len(MAPE['loss'])
    -1, MAPE, accuracy), orient='index', columns=['Model trained now'])

    Gallet_50_50_results = pd.DataFrame.from_dict(extract_results(len(
    Gallet_50_50_MAPE['loss'])-1, Gallet_50_50_MAPE, Gallet_50_50_accuracy)
    , orient='index', columns=['Gallet_50_50'])

    Gallet_600_600_600_results = pd.DataFrame.from_dict(extract_results(len
    (Gallet_600_600_600_MAPE['loss'])-1, Gallet_600_600_600_MAPE,
    Gallet_600_600_600_accuracy), orient='index', columns=['
    Gallet_600_600_600'])

    # Merge results into a single DataFrame
    all_models = pd.merge(user_results, Gallet_50_50_results, left_index=
    True, right_index=True)
    all_models = pd.merge(all_models, Gallet_600_600_600_results,
    left_index=True, right_index=True)

    # Display results
```

```
343     print('--- Step 5: Compare results against Gallet et al. (2023) ---')
344     print(all_models)
345     print('') # # Add new-line after results table
346 
347 
348 # Save the model and MAPE/accuracy files
349 def step6_save_model(trained_model, MAPE, accuracy):
350     print('--- Step 6: Model saved ---')
351     trained_model.save('./saved_models/user_defined_model')
352     with open('./saved_models/user_defined_model_accuracy', 'wb') as
    file_accuracy:
353         pickle.dump(accuracy, file_accuracy)
354     with open('./saved_models/user_defined_model_MAPE', 'wb') as
    file_losses:
355         pickle.dump(MAPE, file_losses)
356 
357     print('Check local "saved_models" directory for saved model, MAPE and
    accuracy dictionary files.\n')
358 
359 
360 # Create and display training results
361 def step7_plot_training_results(MAPE, accuracy):
362     # Create figure, add subplots and adjust padding/margins
363     fig_width=7; fig_height=4.0
364     fig = plt.figure(figsize=(fig_width, fig_height))
365     fig.subplots_adjust(left=0.09, right=0.98,
366                         bottom=0.15, top=0.82,
367                         hspace=1.0, wspace=1.1)
368     fig.suptitle('Learning graphs, compare with Gallet et al. (2023) (doi.
    org/x)', x=0.5, y=0.95)
369 
370     spec = fig.add_gridspec(nrows=2, ncols=5)
371 
372     # Generate individual axes to draw on
373     ax0 = fig.add_subplot(spec[0:1,0:2])
374     ax1 = fig.add_subplot(spec[0,2:5])
375     ax2 = fig.add_subplot(spec[1,2:5])
376 
377     # Epoch list for x-axis values
378     epoch_list = range(1, len(MAPE['loss'])+1)
379 
380     # ax0 plot - Plot loss profiles
    ------------------------------------------------------
381     ax0.set_title('a) Loss profiles', fontsize=11)
382     ax0.set_ylabel('MAPE')
383     ax0.set_xlabel('Epochs')
384     ax0.set_ylim(bottom=-0.10, top=1.10)
385     ax0.scatter(epoch_list, MAPE['loss'], marker='P',
386     color='black', label='Training', zorder=5)
387     ax0.scatter(epoch_list,  MAPE['val_loss'], marker='x',
388     color='red', label='Validation', zorder=5)
389     ax0.legend(ncol=2, loc='center', bbox_to_anchor=(0.45, -0.9),
390     title='Legend for Figure a)', columnspacing=1.0, handletextpad=0.5)
391 
392     # ax1 plot - Plot training accuracy percentiles
    -------------------------------------
393     ax1.set_title('b) Training accuracy', fontsize=11)
```

```
    ax1.set_ylabel('Accuracy')
    ax1.set_xlabel('Epochs')
    ax1.set_ylim(bottom=-0.10, top=2.10)
    ax1.scatter(epoch_list, accuracy['Max'],
    color='green', marker='v', zorder=15, label='Max.')
    ax1.fill_between(epoch_list, accuracy['0.5%'],
    accuracy['99.5%'], hatch='xx', zorder=5, label='99% CI')
    ax1.scatter(epoch_list, accuracy['50%'],
    color='black', marker='.', zorder=10, label='Median')
    ax1.scatter(epoch_list, accuracy['Min'],
    color='red', marker='^', zorder=15, label='Min.')
    ax1.fill_between(epoch_list, accuracy['2.5%'],
    accuracy['97.5%'], hatch='||', zorder=8, label='95% CI')

    ax1.legend(title='Legend for Figures b) and c)',
    ncol=2, loc='center', bbox_to_anchor=(-0.542, -1.7))

    # ax2 plot - Plot validation accuracy percentiles
    ----------------------------------
    ax2.set_title('c) Validation accuracy', fontsize=11)
    ax2.set_ylabel('Accuracy')
    ax2.set_xlabel('Epochs')
    ax2.set_ylim(bottom=-0.10, top=2.10)
    ax2.scatter(epoch_list, accuracy['val_Max'],
    color='green', marker='v', zorder=15, label='Max.')
    ax2.fill_between(epoch_list, accuracy['val_0.5%'],
    accuracy['val_99.5%'], hatch='xx', zorder=5, label='99% CI')
    ax2.scatter(epoch_list, accuracy['val_50%'],
    color='black', marker='.', zorder=10, label='Median')
    ax2.scatter(epoch_list, accuracy['val_Min'],
    color='red', marker='^', zorder=15, label='Min.')
    ax2.fill_between(epoch_list, accuracy['val_2.5%'],
    accuracy['val_97.5%'], hatch='||', zorder=8, label='95% CI')

    # Report completed plot
    print('--- Step 7: Learning graph generated ---')
    print('Learning graph can be compared with training plots\nfound in
    Gallet et al. (2023) (doi.org/x).\n')
    plt.show()


# --- Run as main script only
if __name__ == "__main__":
main()
```

Script 5.1: Neural network training script

Chapter 6

Article: "Towards verifiable continuous structural design models using physics informed neural networks"

Foreword commentary

This work titled "Towards verifiable continuous structural design models using physics informed neural networks" is presented here as a journal article in preparation for a potential *Engineering Applications of Artificial Intelligence* submission, and builds on the research results from Chapter 5. Preliminary research findings from this approach were originally presented on March 23rd, 2023 at the *Institution of Structural Engineers' Young Researchers' Conference* found in Appendix C. The final results presented here continue to demonstrate the inverse problem perspective established in Chapter 3, and tests the viability of physics informed neural networks for structural design models. The investigation is critical in identifying some effective avenues for future research highlighted in Chapter 7. The work was entirely conducted by Adrien Gallet with Danny Smyl providing supervision and reviewing the final draft.

Abstract

Since their inception, physics informed neural networks have shown advantages over traditional data-driven models by reducing the need of large training dataset sizes and embedding physical information in the training process that typically render predictions more consistent with the laws of physics. This work investigates the viability and advantages of physics informed neural network (PINN) formulations for structural design models. To achieve this, local stiffness relations based on Timoshenko–Ehrenfest beam theory are used to constrain the output space for structural size predictions of a continuous beam system. The study presented here had two aims: the first was to investigate if such PINNs provide an effective way of improving the performance characteristics of a structural design

model when compared to networks trained using standard architecture with a data-only loss function. The second aim was to test the viability of a verifiable neural network that could indicate the physical consistency of a prediction during inference. The results indicate that PINNs do improve the performance under testing conditions, but that the correlation between data-losses and physics-losses is weak, limiting the verifiability of the model during inference. Various future avenues of research to develop a truly verifiable architecture are suggested in the conclusions.

6.1 Introduction

The use of physics informed neural networks (PINNs) in civil and structural engineering is beginning to gain attention since their introduction between 2017 to 2019 [5, 90, 91]. A PINN differentiates itself from standard artificial neural networks by supplementing a standard loss function, such as Mean Average Error (MAE) or Mean Average Percentage Error (MAPE) with a physical equation that relates the features and targets of the network based on physical laws that are known to apply to the problem at hand.

One of the earliest applications of PINNs in civil and structural engineering literature was published in 2022 for a geometry identification problems in the field of continuum mechanics, a specific type of inverse problem [523]. In this study, the authors trained a neural network to minimise the difference between the predicted to measured boundary displacement and stress fields based on coordinate information, and thereby update the estimated material and geometric parameters of the square-shaped matrix material based residual of equilibrium equations. A different study in 2023 developed a PINN based topology optimisation (PINNTO) framework to numerically determine the displacement fields, and thereby entirely replace the Finite Element Analysis found in conventional structural topology optimisation approaches [524]. In this work, coordinate information are also used as inputs, and the optimised solution corresponds to the displacement field that minimises the total potential energy. A further work published in 2023 incorporated PINNs within a graph neural network structure to minimise the error in equilibrium equations to structurally analyse the response of frames subject to nodal forces [94]. Lastly, in early 2024, a unified solver-free structural optimiser was developed, in which a physical loss function based on complementary energy equations was minimised to determine the optimal truss structures [512]. Their results compared favourably with low errors when compared to traditional FEA based optimisation approaches.

These examples demonstrate that PINNs have been successfully applied for various applications. From the perspective of structural analysis and design, and the various machine learned operators discussed in Chapter 5 (see specifically Figure 5.2), two of the aforementioned pieces of work focused on building *ML optimisation solvers* using PINNs, in which the goal is to replace traditional finite elements solvers to arrive at optimised designs [512, 524], one piece of work used PINNs to build a *ML forward operator* to help analyse structures [94], and only one investigated the application of PINNs for a *ML inverse operator*, albeit not for the purpose of structural design [523].

The investigation presented here is to focus on how PINNs could potentially be deployed to improve the performance of machine learned inverse operators, specifically focusing on comparing the performance of a physics-informed neural network to the data-only

networks developed in Chapter 5. There are two main objectives based on the current limitations of the inverse operator achieved in the previous investigation: the first is to attempt to reduce the error and accuracy dispersion profiles by incorporating known physical information into the operator. This is substantially different to previous research conducted [512, 524], which created data-less operators only based on well defined physics equations, but closer to what was achieved in previous PINN based ML inverse operator research [523]. In that regard, more evidence for identifying structural design as an inverse operator is presented [140]. The second objective is to theorise and test a potential verifiable architecture of the inverse operator through which the physical validity of the predicted results could be evaluated during inference of the model, namely outside of training and validation. This is a substantial novel contribution to the literature.

Section 6.2 introduces the general methodology on how such a PINN based inverse operator could be built using local stiffness relations, Section 6.3 presents the results from the training runs, Section 6.4 discusses the outcomes and Section 6.5 summarises the areas for future works.

6.2 Methodology

6.2.1 Physics informed neural networks

The general objective of supervised neural networks is to solve a generalised optimization problem by minimizing the empirical loss function defined by Equation 6.1:

$$\arg\min \frac{1}{n} \sum_{i=1}^{n} L\left(\boldsymbol{y}_i, f\left(\boldsymbol{x}_i; \boldsymbol{\theta}\right)\right) + \lambda \Omega\left(\boldsymbol{\theta}\right) \tag{6.1}$$

where $\boldsymbol{y}_i$ are the target variables from a dataset containing n data points, $f\left(\boldsymbol{x}_i; \boldsymbol{\theta}\right)$ is the predicted target variable based on predictors $\boldsymbol{x}_i$ and $\boldsymbol{\theta}$ are the weights and biases of the model. L is simply a loss measured between the true targets $\boldsymbol{y}_i$ and predicted targets $f\left(\boldsymbol{x}_i; \boldsymbol{\theta}\right)$, such as Mean Square Error (MSE) or the Mean Absolute Percentage Error (MAPE).

PINNs differentiate themselves from traditional neural networks with the last term $\lambda \Omega\left(\boldsymbol{\theta}\right)$, which is representative of a regulariser that penalises model parameters $\boldsymbol{\theta}$ that deviate significantly from pre-set value (typically 0) to ensure a non-complex model where no network nodes dominate the predictions. In PINNs, $\Omega\left(\boldsymbol{\theta}\right)$ of the regularisation term is replaced by physical equations , which create a functional relationship between the input and output variables based on known physical phenomena. Hence, in addition to the data loss term, there is a physics loss term, which evaluates if the inputs and output variables construe to physically meaningful solutions.

Normally, the physics loss equations are based on the well established ordinary or partial differential equations (ODE/PDE) such as the one-dimensional heat equation:

$$u_t = k u_{xx} + h(x, t) \tag{6.2}$$

where the variables x and t would be inputs to the network, u the singular output, and terms u_t and u_{xx} being evaluated through auto-differentiation during learning. By then

generating a physics loss function in the form of:

$$\text{MSE}_{\text{Physics}} = \left(ku_{xx} + h(x,t) - u_t\right)^2. \tag{6.3}$$

One can evaluate the magnitude of prediction error, and back-propagate these errors through the network to achieve learned behaviour. The progressive minimisation of the physics loss function represents a convergence towards physically realistic results, and hence can help the neural network during training on how to correctly update the parameters in respect to the loss landscape (or loss surface).

The difficulty in developing a PINN for a structural design model is that structural design is not solely characterised by an ODE/PDE equation. This investigation will however attempt to relate the input to the output space established in Chapter 5 (where inputs were span and UDL values, and outputs the cross-section properties of a continuous beam element), by using non-discretised local stiffness relations.

6.2.2 Local stiffness matrix relations

The relationship between the deformation and internal forces within a member can be expressed via the Timoshenko-Ehrenfest stiffness matrix equations. Suppose we have the continuous beam system indexed by i with ω_i UDLs, L_i spans, with each beam having cross-section properties P_i as shown in Figure 6.1. In this case, the deformation vectors can be related to the internal reaction forces with end-force adjustments (due to the UDLs) using the local stiffness matrix relations as shown in eqation 6.4:

$$\begin{bmatrix} V_{z,1} \\ M_{yy,1} \\ V_{z,2} \\ M_{yy,2} \end{bmatrix} + \begin{bmatrix} \frac{-\omega L}{2} \\ \frac{-\omega L^2}{12} \\ \frac{-\omega L}{2} \\ \frac{\omega L^2}{12} \end{bmatrix} = \frac{EI}{L^3(1+\phi)} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & (4+\phi)L^2 & -6L & (2-\phi)L^2 \\ -12 & -6L & 12 & -6L \\ 6L & (2-\phi)L^2 & -6L & (4+\phi)L^2 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \theta_1 \\ \delta_2 \\ \theta_2 \end{bmatrix} \tag{6.4}$$

$$\text{where } \phi = \tfrac{12EI}{A_z G L^2}$$

with shear forces V_z and bending moments M_{yy}, Youngs Modulus E, shear modulus G, the second moment of area I, shear area A_z and rotations θ and displacements δ at the start and end of each continuous beam member numbered as location 1 and 2 respectively as shown in Figure 6.1c). Note that these locations (1 and 2) always correspond to areas over supports in this work, and that non-discretised members will be used in this investigation, meaning each local stiffness correlation will correspond to a single prismatic beam. Since there are no deflections occurring over the supports, $\delta_1 = 0$ and $\delta_2 = 0$ and Equation 6.4 can be simplified to the following:

$$\begin{bmatrix} V_{z,1} \\ M_{yy,1} \\ V_{z,2} \\ M_{yy,2} \end{bmatrix} + \begin{bmatrix} \frac{-\omega L}{2} \\ \frac{-\omega L^2}{12} \\ \frac{-\omega L}{2} \\ \frac{\omega L^2}{12} \end{bmatrix} = \frac{EI}{L^3(1+\phi)} \begin{bmatrix} 6L & 6L \\ (4+\phi)L^2 & (2-\phi)L^2 \\ -6L & -6L \\ (2-\phi)L^2 & (4+\phi)L^2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}. \tag{6.5}$$

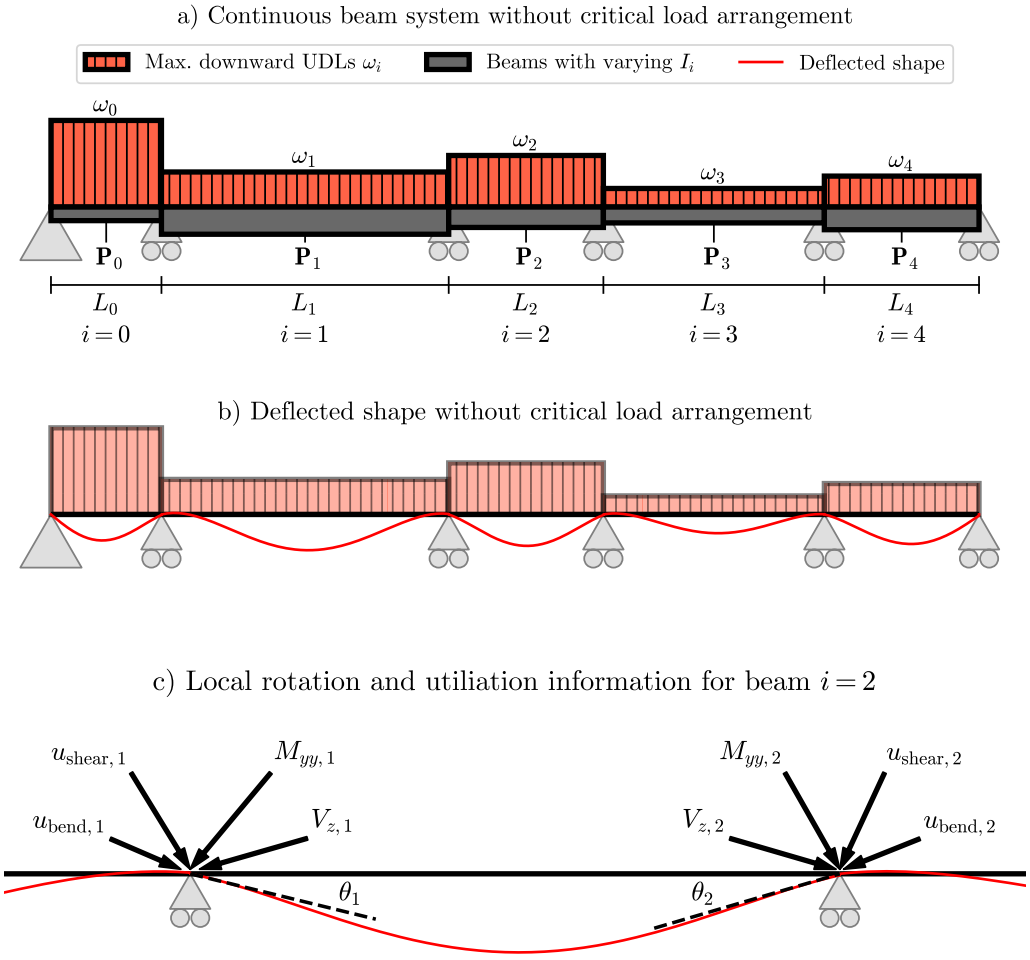


Figure 6.1: Continuous beam system subjugated to uniformly distributed loads with b) deflected shape and c) resulting rotations θ_1 and θ_2. Physical response information in terms of bending u_{bend} and shear u_{shear} utilisation ratios and bending moments M_{yy} and shear force V_z can be identified over each support.

Equation 6.5 can therefore be broken down into the following four equations:

$$
\begin{aligned}
V_{z,1} &= \frac{EI_{yy}}{L^3(1+\phi)}(6L\theta_1 + 6L\theta_2) + \frac{\omega L}{2} \\
M_{yy,1} &= \frac{EI_{yy}}{L^3(1+\phi)}\left((4+\phi)L^2\theta_1 + (2-\phi)L^2\theta_2\right) + \frac{\omega L^2}{12} \\
V_{z,2} &= \frac{EI_{yy}}{L^3(1+\phi)}(-6L\theta_1 - 6L\theta_2) + \frac{\omega L}{2} \\
M_{yy,2} &= \frac{EI_{yy}}{L^3(1+\phi)}\left((2-\phi)L^2\theta_1 + (4+\phi)L^2\theta_2\right) - \frac{\omega L^2}{12}.
\end{aligned}
\tag{6.6}
$$

The neural network developed in Chapter 5 related span L_i and UDL ω_i inputs to cross-sectional area output values, not internal forces V_z and M_{yy}. Using Eurocode 3 [466], it is possible to evaluate the shear and bending utilisation factors using the cross-sectional properties of an element,

$$\begin{aligned} \frac{|V_z|}{A_z \frac{f_y}{\sqrt{3}}} &= u_{\text{shear}} \\ \frac{|M_{yy}|}{W_{\text{pl},yy}\, f_y} &= u_{\text{bend}} \end{aligned} \tag{6.7}$$

which can be re-arranged:

$$\begin{aligned} |V_{z,\text{Ed}}| &= u_{\text{shear}}\, A_z \frac{f_y}{\sqrt{3}} \\ |M_{yy,\text{Ed}}| &= u_{\text{bend}}\, W_{\text{pl},yy} f_y. \end{aligned} \tag{6.8}$$

Consequently, it would then be possible to create a functional, physical relationship between input and output information by making the following substitutions into Equation 6.6:

$$\begin{aligned} u_{\text{shear},1}\, A_z \frac{f_y}{\sqrt{3}} &= \left| \frac{EI_{yy}}{L^3(1+\phi)}(6L\theta_1 + 6L\theta_2) + \frac{\omega L}{2} \right| \\ u_{\text{bending},1}\, W_{\text{pl},yy} f_y &= \left| \frac{EI_{yy}}{L^3(1+\phi)} \left((4+\phi)L^2\theta_1 + (2-\phi)L^2\theta_2\right) + \frac{\omega L^2}{12} \right| \\ u_{\text{shear},2}\, A_z \frac{f_y}{\sqrt{3}} &= \left| \frac{EI_{yy}}{L^3(1+\phi)}(-6L\theta_1 - 6L\theta_2) + \frac{\omega L}{2} \right| \\ u_{\text{bending},2}\, W_{\text{pl},yy} f_y &= \left| \frac{EI_{yy}}{L^3(1+\phi)} \left((2-\phi)L^2\theta_1 + (4+\phi)L^2\theta_2\right) - \frac{\omega L^2}{12} \right|. \end{aligned} \tag{6.9}$$

Note that the absolute signs are only needed because utilisation ratios are expressed as non-negative values. Finally, the re-arranged form of these equations becomes:

$$\begin{aligned} A_z &= \frac{\sqrt{3}}{u_{\text{shear},1}\, f_y} \left| \frac{EI_{yy}}{L^3(1+\phi)}(6L\theta_1 + 6L\theta_2) + \frac{\omega L}{2} \right| \\ W_{\text{pl},yy} &= \frac{1}{u_{\text{bending},1}\, f_y} \left| \frac{EI_{yy}}{L^3(1+\phi)} \left((4+\phi)L^2\theta_1 + (2-\phi)L^2\theta_2\right) + \frac{\omega L^2}{12} \right| \\ A_z &= \frac{\sqrt{3}}{u_{\text{shear},2}\, f_y} \left| \frac{EI_{yy}}{L^3(1+\phi)}(-6L\theta_1 - 6L\theta_2) + \frac{\omega L}{2} \right| \\ W_{\text{pl},yy} &= \frac{1}{u_{\text{bending},2}\, f_y} \left| \frac{EI_{yy}}{L^3(1+\phi)} \left((2-\phi)L^2\theta_1 + (4+\phi)L^2\theta_2\right) - \frac{\omega L^2}{12} \right|. \end{aligned} \tag{6.10}$$

The usefulness of Equation 6.10 can be understood in context of the previous structural design model developed in Chapter 5, namely these equations relate portion of the

input space of the network (span and UDL values) and the associated outputs (the cross section properties of an element within a continuous beam system) as long as the following additional terms are known or are predicted:

- Yield strength of steel f_y, Young's modulus E and shear modulus G.
- Utilisation ratios $u_{\text{shear},1}$ $u_{\text{shear},2}$, $u_{\text{bend},1}$ and $u_{\text{bend},2}$.
- Rotations θ_1 and θ_2.

6.2.3 Data and physics loss functions

It is now possible to formulate a data and physics loss function, L_{data} and L_{phy} respectively to train a physics informed neural network. Both these loss functions are built assuming the same feature and target space as that from the machine learned model from Chapter 5 are provided. As a refresher, these included the span value L_0 and UDLs ω_0 from the beam to be designed along with the spans and UDLs of the 5 adjacently connected elements as inputs (leading to a total of 22 inputs), and the shear area A_z, second moment of area I, and the plastic modulus W_{pl} as outputs.

In general, the combined loss function is defined as:

$$L = L_{\text{data}} + \lambda L_{\text{phy}} \tag{6.11}$$

where λ is introduced to control the extent to which physical constraints should be applied on the parameter space of the network $\boldsymbol{\theta}$. The data loss function L_{data} adopted here is equivalent to the MAPE loss function established in the previous chapter, and is chosen due to its outperformance compared to other loss functions previously tested. Consequently, L_{data} is defined as:

$$L_{\text{data}} = \frac{1}{3}\left(\left|\frac{\widehat{I} - I}{I + \epsilon}\right| + \left|\frac{\widehat{A_z} - A_z}{A_z + \epsilon}\right| + \left|\frac{\widehat{W_{\text{pl}}} - W_{\text{pl}}}{W_{\text{pl}} + \epsilon}\right|\right). \tag{6.12}$$

where $\widehat{I}$, $\widehat{A_z}$, $\widehat{W_{\text{pl}}}$ each represent the predicted cross-section properties. The physics loss function L_{phy} on the other hand will compare predicted cross-section property values (denoted by "pred") against the the physics derived cross-section properties introduced by Equation 6.10, whilst still being constructed as MAPE loss functions:

$$\begin{aligned} L_{\text{phy}} \rightarrow L_{\frac{\text{pred}-\text{phy}}{\text{pred}}} = \frac{1}{4}\Bigg(&\left|\frac{\widehat{A_z} - A_{z,\text{phy},1}}{\widehat{A_z} + \epsilon}\right| + \left|\frac{\widehat{W_{\text{pl}}} - W_{\text{pl,phy},1}}{\widehat{W_{\text{pl}}} + \epsilon}\right| \\ &+ \left|\frac{\widehat{A_z} - A_{z,\text{phy},2}}{\widehat{A_z} + \epsilon}\right| + \left|\frac{\widehat{W_{\text{pl}}} - W_{\text{pl,phy},2}}{\widehat{W_{\text{pl}}} + \epsilon}\right|\Bigg) \end{aligned} \tag{6.13}$$

where the physics derived cross-section properties $A_{z,\text{phy},1}$, $W_{\text{pl,phy},1}$, $A_{z,\text{phy},2}$ and $W_{\text{pl,phy},2}$ are defined by:

$$
\begin{aligned}
A_{z,\text{phy},1} =& \frac{\sqrt{3}}{u_{\text{shear},1}\, f_y} \left| \frac{E\widehat{I}}{L^3(1+\phi)}(6L\theta_1 + 6L\theta_2) + \frac{\omega L}{2} \right| \\
W_{\text{pl,phy},1} =& \frac{1}{u_{\text{bend},1}\, f_y} \left| \frac{E\widehat{I}}{L^3(1+\phi)} \left((4+\phi)L^2\theta_1 + (2-\phi)L^2\theta_2\right) + \frac{\omega L^2}{12} \right| \\
A_{z,\text{phy},2} =& \frac{\sqrt{3}}{u_{\text{shear},2}\, f_y} \left| \frac{E\widehat{I}}{L^3(1+\phi)}(-6L\theta_1 - 6L\theta_2) + \frac{\omega L}{2} \right| \\
W_{\text{pl,phy},2} =& \frac{1}{u_{\text{bend},2}\, f_y} \left| \frac{E\widehat{I}}{L^3(1+\phi)} \left((2-\phi)L^2\theta_1 + (4+\phi)L^2\theta_2\right) - \frac{\omega L^2}{12} \right| \\
& \text{where } \phi = \tfrac{12E\widehat{I}}{\widehat{A_z}GL^2}
\end{aligned}
\tag{6.14}
$$

It is worth drawing attention to the fact that the physics derived cross-section properties $A_{z,\text{phy},1}$, $A_{z,\text{phy},2}$, $W_{\text{pl,phy},1}$ and $W_{\text{pl,phy},2}$ as shown in Equation 6.14 are based on both the UDL ω and span L of the target beam, along with the *predicted* second moment of area $\widehat{I}$ and shear area $\widehat{A_z}$ values. In other words, there is now a partial physical relationship between the input and output space. In end effect, the transformation from Equation 6.4 to 6.14 has yielded a physical equation which evaluates what the physically consistent $A_{z,\text{phy},1}$, $A_{z,\text{phy},2}$, $W_{\text{pl,phy},1}$ and $W_{\text{pl,phy},2}$ values should be for a specific beam, and any deviation from those can be minimised with the L_{phy} loss function established by Equation 6.13.

As discussed previously, in order to be able to minimise L_{phy}, the remaining physical information needs to be known. This includes the Youngs and shear modulus E and G, the various utilisation factors $u_{\text{shear},1}$ $u_{\text{shear},2}$, $u_{\text{bend},1}$ and $u_{\text{bend},2}$, and the rotations θ_1 and θ_2. In this study, these values were initially extracted from structural analysis simulation runs based on all UDLs applied (that is no pattern loading). For the first set of networks with a *standard architecture*, these additional values were simply provided to the physics loss function L_{phy} as known data, since this means the output space does not need to be changed, and therefore allows direct comparison with the results achieved in Chapter 5 to determine the advantages of the proposed PINN formulation. A second *verifiable architecture* will however also be tested in which the $u_{\text{shear},1}$, $u_{\text{shear},2}$, $u_{\text{bend},1}$ and $u_{\text{bend},2}$, and the two rotations θ_1 and θ_2 will be predicted in addition to the three cross-section parameters I, A_z, W_{pl}, resulting in a total output space of nine values. This allows the physical validity of the predictions to be tested during inference discussed in Section 6.4.

Physics loss functions of standard architectures

It is in fact possible to arrive at three further physics loss functions L_{phy}. The first one introduced in equation 6.13 above compares predicted cross section values against physics derived ones, and subsequently divides by the predicted values to evaluate a percentage-like error estimation. However, it is relatively easy to build variations of the function:

$$L_{\frac{\text{true}-\text{phy}}{\text{true}}} = \frac{1}{4}\left(\left|\frac{A_z - A_{z,\text{phy},1}}{A_z + \epsilon}\right| + \left|\frac{W_{\text{pl}} - W_{\text{pl,phy},1}}{W_{\text{pl}} + \epsilon}\right| + \left|\frac{A_z - A_{z,\text{phy},2}}{A_z + \epsilon}\right| + \left|\frac{W_{\text{pl}} - W_{\text{pl,phy},2}}{W_{\text{pl}} + \epsilon}\right|\right) \tag{6.15}$$

$$L_{\frac{\text{phy}-\text{pred}}{\text{phy}}} = \frac{1}{4}\left(\left|\frac{A_{z,\text{phy},1} - \widehat{A_z}}{A_{z,\text{phy},1} + \epsilon}\right| + \left|\frac{W_{\text{pl,phy},1} - \widehat{W_{\text{pl}}}}{W_{\text{pl,phy},1} + \epsilon}\right| + \left|\frac{A_{z,\text{phy},2} - \widehat{A_z}}{A_{z,\text{phy},2} + \epsilon}\right| + \left|\frac{W_{\text{pl,phy},2} - \widehat{W_{\text{pl}}}}{W_{\text{pl,phy},2} + \epsilon}\right|\right) \tag{6.16}$$

$$L_{\frac{\text{phy}-\text{true}}{\text{phy}}} = \frac{1}{4}\left(\left|\frac{A_{z,\text{phy},1} - A_z}{A_{z,\text{phy},1} + \epsilon}\right| + \left|\frac{W_{\text{pl,phy},1} - W_{\text{pl}}}{W_{\text{pl,phy},1} + \epsilon}\right| + \left|\frac{A_{z,\text{phy},2} - A_z}{A_{z,\text{phy},2} + \epsilon}\right| + \left|\frac{W_{\text{pl,phy},2} - W_{\text{pl}}}{W_{\text{pl,phy},2} + \epsilon}\right|\right). \tag{6.17}$$

Each of these PINN formulations differentiate themselves slightly. Whilst Equation 6.13 evaluates the percentage loss by dividing by the predicted cross-sectional properties, Equation 6.15 achieves this by dividing by the true cross-sectional values, and Equations 6.16 and 6.17 by the physics-evaluated properties via Equation 6.14. Furthermore, the numerators across all of these formulations differ, by either comparing the physics derived properties from Equation 6.14 with either predicted properties as in Equations 6.13 and 6.16, or true properties as is the case for Equations 6.15 and 6.17. Given the novelty of using the local stiffness relations to physically constrain a structural design model, all four variations will be tested in the standard architectures to determine which of these physics-loss functions L_{phy} perform best.

Physics loss functions for verifiable networks

One advantage of the $L_{\frac{\text{pred}-\text{phy}}{\text{pred}}}$ shown in Equation 6.13 is that it relies primarily on predicted cross-section values. If additionally, all further response information was predicted by the network, that is the various utilisation ratios $u_{\text{shear},1}$, $u_{\text{shear},2}$, $u_{\text{bend},1}$ and $u_{\text{bend},2}$, and the two rotation values θ_1 and θ_2 for each individual beam, then it should be possible to evaluate the physics derived cross-section properties using the input and output vectors alone. In this case, $\widehat{A_{z,\text{phy},1}}$, $\widehat{A_{z,\text{phy},2}}$, $\widehat{W_{\text{pl,phy},1}}$ and $\widehat{W_{\text{pl,phy},2}}$ are evaluated by:

$$
\begin{aligned}
\widehat{A_{z,\text{phy},1}} &= \frac{\sqrt{3}}{\widehat{u_{\text{shear},1}}\, f_y} \left| \frac{E\widehat{I}}{L^3(1+\phi)}(6L\widehat{\theta_1} + 6L\widehat{\theta_2}) + \frac{\omega L}{2} \right| \\
\widehat{W_{\text{pl,phy},1}} &= \frac{1}{\widehat{u_{\text{bend},1}}\, f_y} \left| \frac{E\widehat{I}}{L^3(1+\phi)} \left((4+\phi)L^2\widehat{\theta_1} + (2-\phi)L^2\widehat{\theta_2} \right) + \frac{\omega L^2}{12} \right| \\
\widehat{A_{z,\text{phy},2}} &= \frac{\sqrt{3}}{\widehat{u_{\text{shear},2}}\, f_y} \left| \frac{E\widehat{I}}{L^3(1+\phi)}(-6L\widehat{\theta_1} - 6L\widehat{\theta_2}) + \frac{\omega L}{2} \right| \\
\widehat{W_{\text{pl,phy},2}} &= \frac{1}{\widehat{u_{\text{bend},2}}\, f_y} \left| \frac{E\widehat{I}}{L^3(1+\phi)} \left((2-\phi)L^2\widehat{\theta_1} + (4+\phi)L^2\widehat{\theta_2} \right) - \frac{\omega L^2}{12} \right|
\end{aligned}
\tag{6.18}
$$

and used in the following verifiable physics loss function:

$$
\begin{aligned}
L_{\text{phy,verifiable}} = \frac{1}{4} \Bigg(& \left| \frac{\widehat{A_z} - \widehat{A_{z,\text{phy},1}}}{\widehat{A_z} + \epsilon} \right| + \left| \frac{\widehat{W_{\text{pl}}} - \widehat{W_{\text{pl,phy},1}}}{\widehat{W_{\text{pl}}} + \epsilon} \right| \\
& + \left| \frac{\widehat{A_z} - \widehat{A_{z,\text{phy},2}}}{\widehat{A_z} + \epsilon} \right| + \left| \frac{\widehat{W_{\text{pl}}} - \widehat{W_{\text{pl,phy},2}}}{\widehat{W_{\text{pl}}} + \epsilon} \right| \Bigg).
\end{aligned}
\tag{6.19}
$$

In this case, the only required values would be the material properties, namely the Yield strength f_y, Young's modulus E and shear modulus G. These can be easily established, as for example would be the case using the *CBeamXP* dataset [6].

6.2.4 Standard and verifiable neural network architectures

Two distinct physics loss functions are to be tested, a standard and verifiable physics loss function, and each will require a different output size. The standard physics loss functions from Equations 6.13, 6.15, 6.16 and 6.17 will be tested with the exact same architectures as those established in Chapter 5, namely a 50-50 and 600-600-600 architectures with 22 inputs that encompass the influence zone, the $a_{\text{in,ReLU}}$ and $a_{\text{out,exp}}$ activation functions. The 600-600-600 was specifically chosen due to its optimal performance profile evaluated empirically in Section 5.4.3. The only difference will be the addition of the λL_{phy} term to enforce the physical constraints on the learning parameters. The verifiable loss function from Equation 6.19 will contain the exact same internal architecture (that is the same hidden layer structure), yet be tested tested on an expanded output size of size 9 to be able to predict the utilisation ratios and rotations of the local beam element to be designed. These predicted values can then be used in Equation 6.19 to evaluate the $L_{\text{phy,verifiable}}$. Different outer activation functions will be applied to certain outputs to match their ranges as shown in Figure 6.2.

Overview of neural network training details

Both architectures will require extraction of the utilisation ratio and rotation information during training. This was easily achieved by using the CBeamXP dataset to model

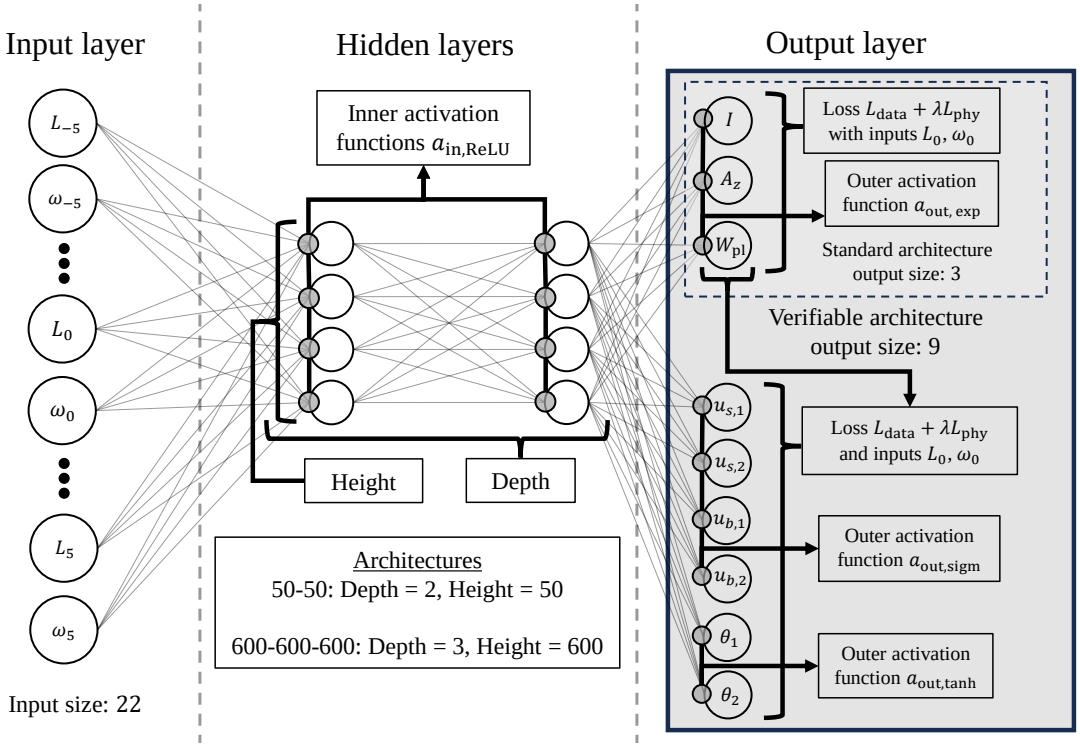


Figure 6.2: Overview of the standard and verifiable neural network architectures to be investigated with hidden layer sizes 50-50 and 600-600-600. In the standard architecture, physical information (utilisation ratios u and rotations θ) are passed directly into the physics function L_{phy} as true values to allow comparison with previous research results. In the verifiable architecture, the physical information itself is predicted to allow for verification at inference. Notice that u_{shear} and u_{bending} have been abbreviated to u_{s} and u_{b} for visualisation purposes.

the individual systems in *Grasshopper* [32] with *Karamba3D* [476]. The utilisation ratios and rotations over each support were then extracted for each individual continuous system each containing 11 beams. All other neural network training details were kept identical to those from the previous machine learning model developed in Chapter 5, which were obtained based on empirical tests, and are briefly summarised in Table 6.1 below.

Neural network training option	Chosen parameter
Optimiser	Nadam
Learning rate	$\alpha = 0.0005$
Initialiser	Gaussian with $\mu = 0$, $\sigma = 0.05$
Batch-size	1024

Table 6.1: Overview of neural network learning parameters and hyperparameters during standard and verifiable PINN architecture testing.

6.3 Results

Various L_{phy} loss functions were tested using standard architectures only. The purpose of these tests was to identify which physics loss functions are best suited for the problem at hand. Additionally, in order to study the impact of dataset size (since PINNs have been shown to require less data to achieve the similar performance in the past [5]), 50-50 architectures were tested with three separate training dataset sizes, whilst varying the λ factor from 0 to 1000 in increasing orders of magnitudes. These results are shown numerically in Tables 6.2, 6.3, 6.4, and graphically in Figure 6.3.

6.3.1 Performance comparison of various L_{phy} loss function formulations for standard architectures

The following three Tables 6.2, 6.3 and 6.4 compare the performance of data-driven models only (when $\lambda = 0$) against PINNs with increasing physics constraints. In each table L_{data} presented the data-derived MAPE for each network, with both training and validation results indicated, with the specified physics loss function $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$, $L_{\frac{\text{phy}-\text{true}}{\text{true}}}$, $L_{\frac{\text{pred}-\text{phy}}{\text{phy}}}$ or $L_{\frac{\text{true}-\text{phy}}{\text{phy}}}$ errors shown as well.

10k training dataset size

	Training with $L_{\text{data}} + \lambda L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$				Training with $L_{\text{data}} + \lambda L_{\frac{\text{phy}-\text{true}}{\text{true}}}$			
	L_{data}		$L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$		L_{data}		$L_{\frac{\text{phy}-\text{true}}{\text{true}}}$	
λ	Train.	Val.	Train.	Val.	Train.	Val.	Train.	Val.
0	0.166	0.181	0.394	0.417	0.166	0.181	0.288	0.358
0.0001	0.160	0.177	0.352	0.375	0.165	0.181	0.286	0.355
0.001	0.203	0.219	0.423	0.472	0.166	0.180	0.288	0.355
0.01	0.149	0.169	0.290	0.361	0.166	0.180	0.270	0.353
0.1	0.184	0.199	0.281	0.374	0.208	0.224	0.244	0.400
1	0.233	0.244	0.199	0.343	0.299	0.316	0.289	0.465
10	0.345	0.345	0.223	0.346	0.411	0.428	0.323	0.544
100	0.312	0.340	0.175	0.325	0.459	0.471	0.349	0.575
1000	0.331	0.340	0.186	0.322	0.459	0.500	0.340	0.564
	Training with $L_{\text{data}} + \lambda L_{\frac{\text{pred}-\text{phy}}{\text{phy}}}$				Training with $L_{\text{data}} + \lambda L_{\frac{\text{true}-\text{phy}}{\text{phy}}}$			
	L_{data}		$L_{\frac{\text{pred}-\text{phy}}{\text{phy}}}$		L_{data}		$L_{\frac{\text{true}-\text{phy}}{\text{phy}}}$	
λ	Train.	Val.	Train.	Val.	Train.	Val.	Train.	Val.
0	0.166	0.181	0.597	0.459	0.166	0.181	1.132	0.745
0.0001	1.018	0.983	1.806	1.889	0.816	0.816	16.18	17.83
0.001	1.132	1.053	1.483	1.996	25.76	25.08	1.402	3.997
0.01	42.56	41.88	5.867	8.135	50.72	49.70	1.425	3.862
0.1	12.93	12.49	7.234	13.73	9.803	8.957	1.148	2.011
1	69.49	68.50	4.229	8.497	15.39	14.69	1.354	2.626
10	101.7	101.4	2.281	4.123	80.00	79.14	1.120	1.535
100	55.50	54.56	2.826	4.307	70.21	69.56	1.074	1.680
1000	79.90	78.62	3.520	4.131	153.6	152.0	0.974	1.280

Table 6.2: Performance profiles for standard 50-50 architectures at epoch 1000 with various $L_{\text{data}} + \lambda L_{\text{phy}}$ loss function combinations using 10k training and 150k validation sets.

100k training dataset size

	Training with $L_{\text{data}} + \lambda L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$				Training with $L_{\text{data}} + \lambda L_{\frac{\text{phy}-\text{true}}{\text{true}}}$			
	L_{data}		$L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$		L_{data}		$L_{\frac{\text{phy}-\text{true}}{\text{true}}}$	
λ	Train.	Val.	Train.	Val.	Train.	Val.	Train.	Val.
0	0.086	0.091	0.225	0.179	0.086	0.091	0.197	0.167
0.0001	0.090	0.094	0.287	0.203	0.080	0.082	0.179	0.162
0.001	0.092	0.095	0.203	0.206	0.076	0.081	0.176	0.142
0.01	0.082	0.083	0.181	0.170	0.080	0.084	0.150	0.149
0.1	0.078	0.087	0.121	0.176	0.106	0.106	0.164	0.190
1	0.104	0.112	0.132	0.186	0.166	0.174	0.190	0.267
10	0.190	0.207	0.165	0.223	0.227	0.223	0.234	0.324
100	0.175	0.173	0.131	0.203	0.267	0.260	0.263	0.309
1000	0.243	0.286	0.154	0.224	0.307	0.330	0.289	0.349
	Training with $L_{\text{data}} + \lambda L_{\frac{\text{pred}-\text{phy}}{\text{phy}}}$				Training with $L_{\text{data}} + \lambda L_{\frac{\text{true}-\text{phy}}{\text{phy}}}$			
	L_{data}		$L_{\frac{\text{pred}-\text{phy}}{\text{phy}}}$		L_{data}		$L_{\frac{\text{true}-\text{phy}}{\text{phy}}}$	
λ	Train.	Val.	Train.	Val.	Train.	Val.	Train.	Val.
0	0.086	0.091	0.293	0.283	0.086	0.091	0.361	0.336
0.0001	0.588	0.571	4.831	4.731	0.462	0.462	18.12	18.636
0.001	0.802	0.802	4.189	4.216	1.144	2.939	5.336	4.114
0.01	44.28	44.74	6.339	6.122	10.72	10.91	2.630	2.617
0.1	71.70	72.70	2.838	3.825	53.62	54.34	1.550	1.811
1	22.33	23.687	7.739	7.587	38.61	39.24	1.448	2.500
10	0.989	0.993	0.935	0.954	27.28	27.46	2.166	2.288
100	691.7	695.0	3.856	4.068	599.0	610.8	0.965	0.960
1000	121.5	122.9	2.662	3.240	1.819	1.863	3.051	3.803

Table 6.3: Performance profiles for standard 50-50 architectures at epoch 1000 with various $L_{\text{data}} + \lambda L_{\text{phy}}$ loss function combinations using 100k training and 150k validation sets.

700k training dataset size

	Training with $L_{\text{data}} + \lambda L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$				Training with $L_{\text{data}} + \lambda L_{\frac{\text{phy}-\text{true}}{\text{true}}}$			
	L_{data}		$L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$		L_{data}		$L_{\frac{\text{phy}-\text{true}}{\text{true}}}$	
λ	Train.	Val.	Train.	Val.	Train.	Val.	Train.	Val.
0	0.066	0.065	0.171	0.157	0.066	0.065	0.150	0.133
0.0001	0.065	0.065	0.159	0.159	0.062	0.063	0.125	0.118
0.001	0.057	0.058	0.137	0.129	0.067	0.066	0.150	0.134
0.01	0.064	0.064	0.138	0.135	0.065	0.065	0.136	0.134
0.1	0.060	0.055	0.120	0.117	0.062	0.055	0.117	0.100
1	0.084	0.077	0.134	0.134	0.081	0.072	0.124	0.115
10	0.119	0.112	0.131	0.137	0.126	0.110	0.166	0.143
100	0.139	0.133	0.116	0.121	0.135	0.124	0.179	0.170
1000	0.299	0.184	0.117	0.130	0.152	0.140	0.172	0.170
	Training with $L_{\text{data}} + \lambda L_{\frac{\text{pred}-\text{phy}}{\text{phy}}}$				Training with $L_{\text{data}} + \lambda L_{\frac{\text{true}-\text{phy}}{\text{phy}}}$			
	L_{data}		$L_{\frac{\text{pred}-\text{phy}}{\text{phy}}}$		L_{data}		$L_{\frac{\text{true}-\text{phy}}{\text{phy}}}$	
λ	Train.	Val.	Train.	Val.	Train.	Val.	Train.	Val.
0	0.066	0.065	0.262	0.193	0.066	0.065	0.343	0.258
0.0001	0.582	0.457	8.432	2.182	0.386	0.387	18.68	18.63
0.001	0.702	0.704	2.256	2.471	0.803	0.738	7.411	8.957
0.01	0.703	0.704	1.190	1.166	0.778	0.775	6.887	6.777
0.1	0.494	0.494	0.457	0.462	17.02	29.60	6.022	4.665
1	0.560	0.557	0.270	0.275	0.743	0.741	18.59	18.60
10	13660	6713	142.1	23.33	1.369	1.364	13.91	16.23
100	37.09	31.86	0.954	0.955	0.823	0.827	20.46	18.55
1000	0.564	0.562	0.227	0.227	29.25	46.78	3.435	1.635

Table 6.4: Performance profiles for standard 50-50 architectures at epoch 1000 with various $L_{\text{data}} + \lambda L_{\text{phy}}$ loss function combinations using 700k training and 150k validation sets.

Visual comparison of results

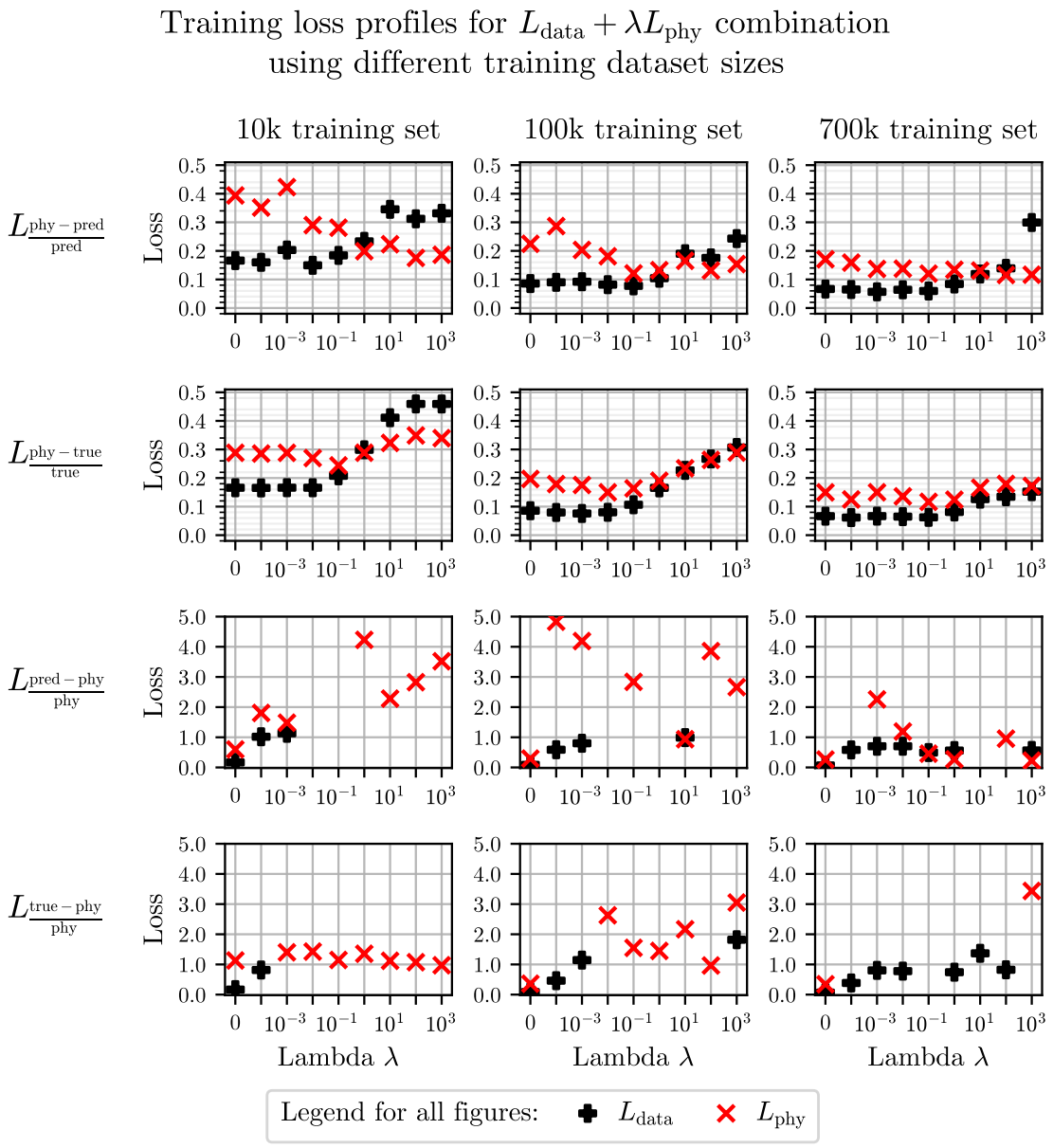


Figure 6.3: Training losses for standard 50-50 architecture at epoch 1000 for various $L_{\text{data}} + \lambda L_{\text{phy}}$ loss functions and training dataset sizes.

The results from Tables 6.2, 6.3 and 6.4 can also be visualised for ease of comparison as shown in Figure 6.3. This figure indicates the impact which increasing the dataset size has on the overall training performance in terms of the pure data loss L_{data} and physics loss L_{physics}. The first observation is the deterioration in performance when the physics loss function uses the physics-estimated cross-section properties within the denominator

as is the case for $L_{\frac{\text{pred}-\text{phy}}{\text{phy}}}$ and $L_{\frac{\text{true}-\text{phy}}{\text{phy}}}$. This may be due to the volatility associated with dividing the losses through the physics estimated cross-section properties, which is a more volatile measure than dividing by the true cross-section properties, such as $L_{\frac{\text{phy}-\text{true}}{\text{true}}}$ or predicted cross-section properties $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$.

Furthermore, increased dataset size generally increased the smoothness of the results, yet from the remaining physics loss function, $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ was the only loss function in which physics losses continued to decrease with increasing λ values. The fact that data losses appear to deteriorate significantly for values larger than $\lambda = 1$ was also an interesting result that will be discussed in further in Section 6.4. For theses reasons, the remaining training runs conducted in this study used $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ as the physics loss function only.

6.3.2 Detailed performance profiles of standard architectures with $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ and 700k training set

Results for λ variations with standard architectures

	Standard 50-50 architecture										
	L_{data}		$L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$		Validation Accuracies Percentiles						
λ	Train.	Val.	Train.	Val.	Min	0.5%	2.5%	Median	97.5%	99.5%	Max
0	0.066	0.065	0.171	0.157	0.032	0.523	0.785	0.994	1.185	1.458	19.93
0.0001	0.065	0.065	0.159	0.159	0.031	0.510	0.803	1.006	1.187	1.454	44.61
0.001	0.057	0.058	0.137	0.129	0.055	0.548	0.799	0.983	1.143	1.402	14.79
0.01	0.064	0.064	0.138	0.135	0.037	0.539	0.804	0.996	1.191	1.489	33.09
0.1	0.060	0.055	0.120	0.117	0.049	0.596	0.835	1.001	1.166	1.440	29.31
1	0.084	0.077	0.134	0.134	0.037	0.579	0.802	1.001	1.279	1.750	26.29
10	0.119	0.112	0.131	0.137	0.041	0.401	0.689	1.006	1.444	2.492	108.9
100	0.139	0.133	0.116	0.121	0.108	0.594	0.774	1.012	1.506	3.053	235.4
1000	0.299	0.184	0.117	0.130	0.064	0.473	0.735	1.008	1.503	3.351	1420
	Standard 600-600-600 architecture										
	L_{data}		$L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$		Validation Accuracies Percentiles						
λ	Train.	Val.	Train.	Val.	Min	0.5%	2.5%	Median	97.5%	99.5%	Max
0	0.009	0.016	0.021	0.030	0.275	0.920	0.968	1.006	1.058	1.133	26.81
0.0001	0.009	0.016	0.021	0.032	0.220	0.891	0.946	0.994	1.036	1.099	15.87
0.001	0.009	0.016	0.022	0.033	0.159	0.896	0.948	0.993	1.035	1.102	26.36
0.01	0.010	0.016	0.022	0.033	0.316	0.896	0.949	0.994	1.037	1.108	7.800
0.1	0.014	0.018	0.034	0.039	0.232	0.884	0.942	0.991	1.032	1.094	8.878
1	0.032	0.027	0.064	0.052	0.178	0.838	0.921	0.997	1.081	1.188	11.90
10	0.034	0.029	0.046	0.057	0.181	0.828	0.916	1.001	1.090	1.237	24.86
100	0.053	0.041	0.051	0.058	0.149	0.816	0.910	1.000	1.110	1.316	531.4
1000	0.128	0.071	0.049	0.056	0.189	0.815	0.908	1.003	1.118	1.447	2043

Table 6.5: Performance profiles for standard 50-50 and 600-600-600 architectures at epoch 1000 with $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ loss function combination using 700k training and 150k validation sets.

An in-depth breakdown of the performance profile of the PINNs using the $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ physics loss function and 700k training set are provided in Table 6.5. These include showing the accuracy percentile profiles achieved on the validation set containing 150k data points for both the 50-50 and 600-600-600 architectures.

Visual representation of results for λ variations for standard architectures

The key trends of this performance profile can also be shown visually with Figures 6.4 and 6.5.

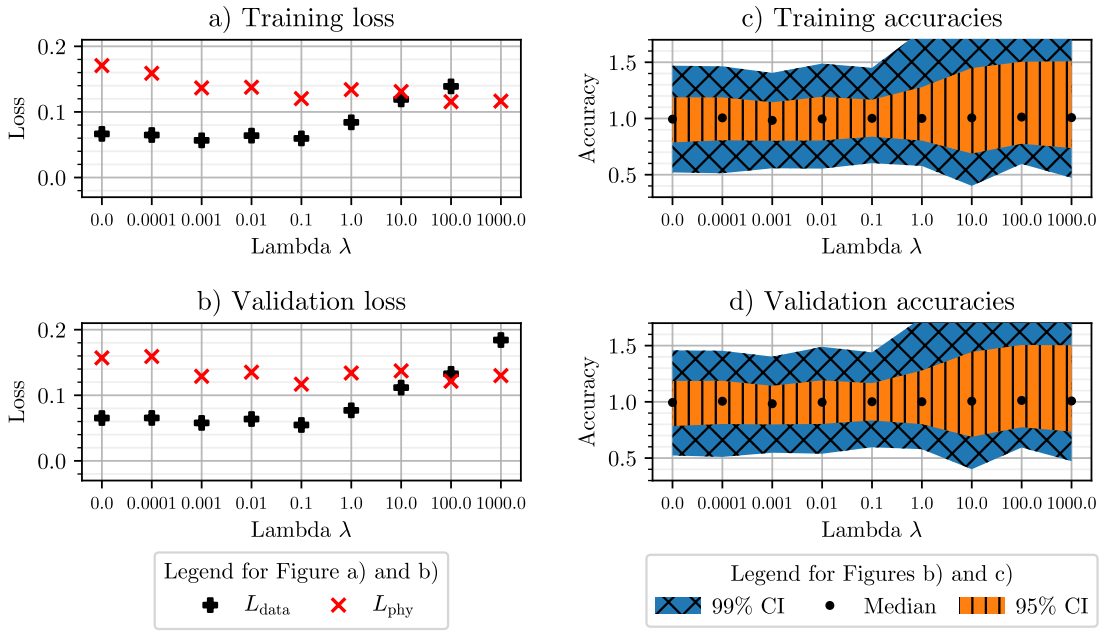


Figure 6.4: Visual performance profiles for the standard 50-50 architecture at epoch 1000 with $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ loss function using 700k training and 150k validation sets.

These results indicate that PINNs formulated with the $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ loss function appear to provide some benefit to smaller/simpler architectures such as with 50-50 hidden layers, demonstrated by the reduction in the physics loss L_{phy} and a tightening of the accuracy percentiles for lambda values up to $\lambda = 1$. For the 600-600-600 architecture, there is no visible advantage using PINNs. In both cases the performance deteriorates for lambda values greater than $\lambda > 1$.

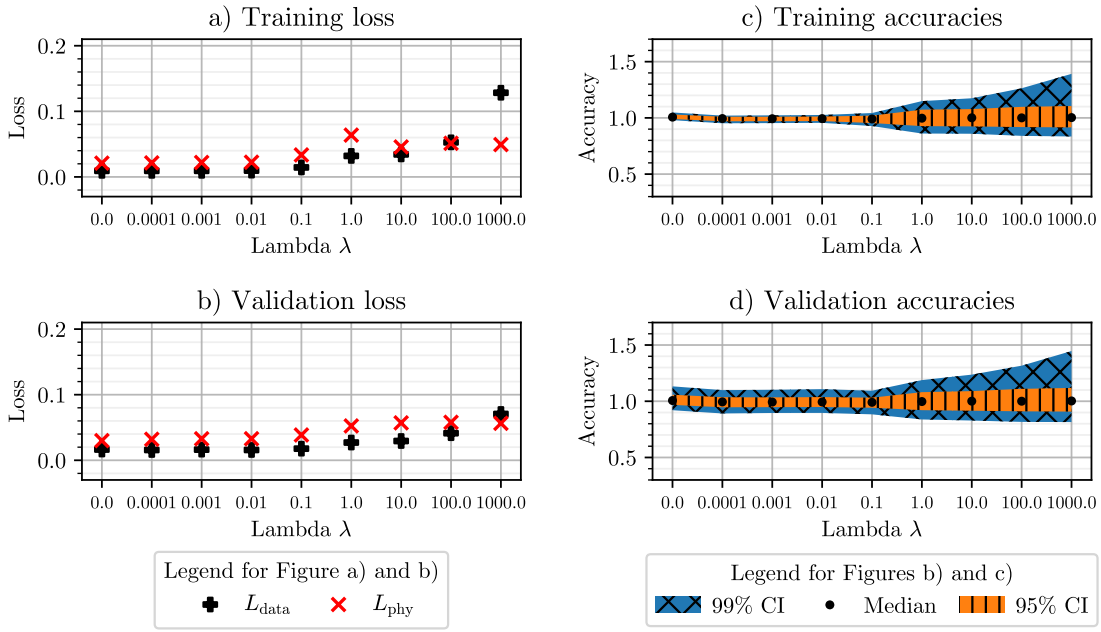


Figure 6.5: Visual performance profiles for the standard 600-600-600 architecture at epoch 1000 with $L_{\frac{\text{phy-pred}}{\text{pred}}}$ loss function using 700k training and 150k validation sets.

Performance comparison for training, validation and testing data for $\lambda = 0.1$ models with full training graphs for standard architectures

Model	Dataset	L_{data}	Accuracy Percentiles						
			Min	0.5%	2.5%	Median	97.5%	99.5%	Max
Standard 50-50 model with $\lambda = 0.0$	Train.	0.066	0.022	0.520	0.787	0.994	1.188	1.470	23.33
	Val.	0.065	0.032	0.523	0.785	0.994	1.185	1.458	19.94
	Test.	0.066	0.032	0.514	0.785	0.994	1.189	1.477	18.43
Standard 50-50 model with $\lambda = 0.1$	Train.	0.060	0.020	0.603	0.838	1.001	1.167	1.450	14.32
	Val.	0.055	0.049	0.596	0.835	1.001	1.166	1.440	29.31
	Test.	0.054	0.070	0.601	0.837	1.001	1.169	1.470	12.91
Standard 600-600-600 model with $\lambda = 0.0$	Train.	0.009	0.802	0.980	0.990	1.007	1.030	1.048	1.272
	Val.	0.016	0.275	0.920	0.968	1.006	1.058	1.133	26.81
	Test.	0.016	0.313	0.917	0.967	1.006	1.058	1.138	12.28
Standard 600-600-600 model with $\lambda = 0.1$	Train.	0.014	0.488	0.927	0.954	0.990	1.020	1.043	5.289
	Val.	0.018	0.232	0.884	0.942	0.991	1.032	1.094	8.878
	Test.	0.018	0.163	0.881	0.942	0.990	1.032	1.095	10.36

Table 6.6: Loss and accuracy profiles for standard architectures at epoch 1000 trained with $\lambda = 0.1$ and the $L_{\frac{\text{phy-pred}}{\text{pred}}}$ with 700k training, 150k validation sets and 150k testing set.

Lastly, a full set of results of accuracy profiles and L_{data} errors trained with $\lambda = 0.1$

and the $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ loss function are shown in Table 6.6, with training profiles shown in Figures 6.6 and 6.7.

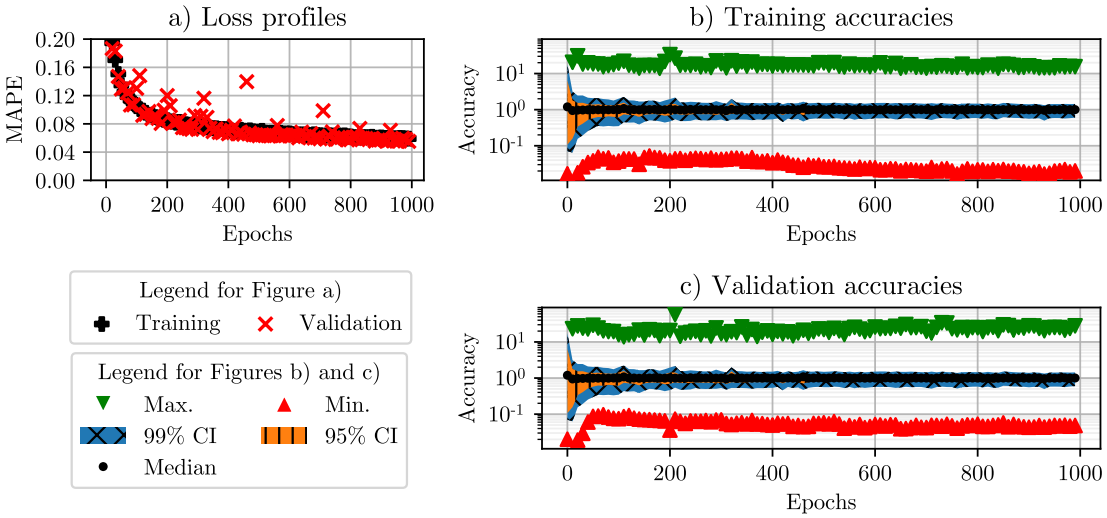


Figure 6.6: Visual performance profiles for standard 50-50 architecture at epoch 1000 with with $\lambda = 0.1$ and the $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ using 700k training and 150k validation sets.

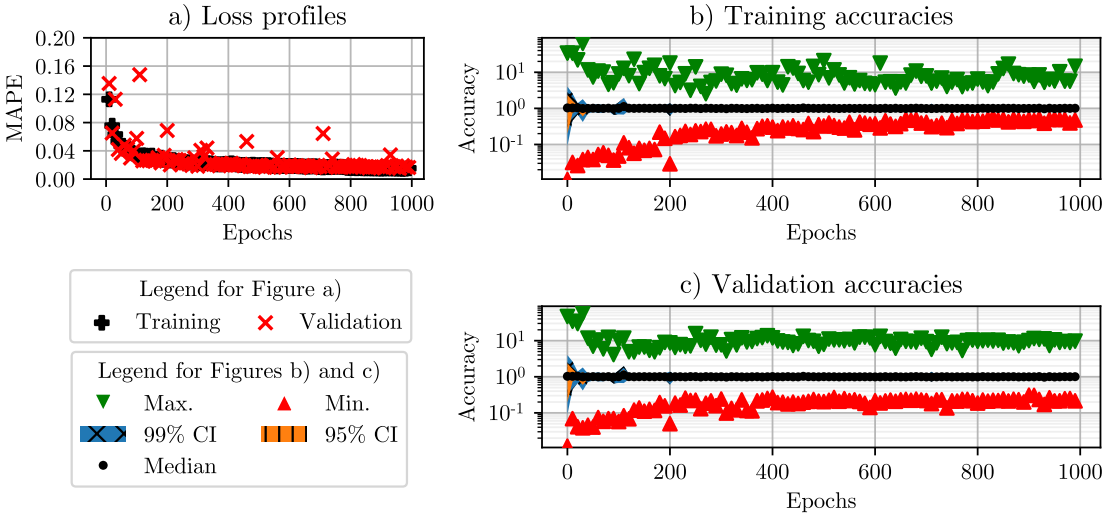


Figure 6.7: Visual performance profiles for standard 600-600-600 architecture at epoch 1000 with with $\lambda = 0.1$ and the $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ using 700k training and 150k validation sets.

As a general observation, which will be discussed fully in Section 6.4, PINNs tend to increase the performance of smaller standard architectures such as 50-50 slightly, yet there is no substantial performance improvement when applied to larger architectures.

6.3.3 Verifiable PINNs with $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$

Presented here are all the results associated with the verifiable architecture trained with the $L_{\text{data}} + \lambda L_{\text{phy,verifiable}}$ loss function combination. Since only one physics function is applicable here, and since there was no advantage to training with reduced dataset sizes, the verifiable results presented here were all training with the 700k dataset.

Results for λ variations with verifiable architectures

	Verifiable 50-50 architecture										
	L_{data}		$L_{\text{phy,verifiable}}$		Validation Accuracies Percentiles						
λ	Train.	Val.	Train.	Val.	Min	0.5%	2.5%	Median	97.5%	99.5%	Max
0	0.064	0.064	224.9	229.0	0.026	0.531	0.788	0.997	1.185	1.444	25.08
0.0001	0.063	0.063	1.347	1.478	0.029	0.520	0.790	1.000	1.180	1.433	27.67
0.001	0.064	0.063	0.452	0.514	0.045	0.528	0.767	0.997	1.170	1.377	18.26
0.01	0.063	0.063	0.186	0.188	0.042	0.503	0.796	0.999	1.185	1.463	63.53
0.1	0.067	0.067	0.097	0.102	0.034	0.542	0.798	0.996	1.195	1.500	47.62
1	0.140	0.141	0.023	0.023	0.024	0.193	0.427	0.987	1.244	1.615	37.52
10	0.694	0.691	0.010	0.011	0.012	0.095	0.203	1.492	2.310	2.752	12.99
100	1.068	1.063	0.006	0.006	0.002	0.031	0.086	1.852	3.036	3.945	23.76
1000	2.388	2.377	0.004	0.004	0.023	0.100	0.226	3.258	7.088	11.73	88.62
	Verifiable 600-600-600 architecture										
	L_{data}		$L_{\text{phy,verifiable}}$		Validation Accuracies Percentiles						
λ	Train.	Val.	Train.	Val.	Min	0.5%	2.5%	Median	97.5%	99.5%	Max
0	0.010	0.014	98.66	109.6	0.272	0.906	0.954	0.992	1.020	1.066	12.41
0.0001	0.010	0.013	0.383	0.316	0.279	0.911	0.956	0.993	1.023	1.068	19.21
0.001	0.010	0.013	0.136	0.161	0.170	0.934	0.976	1.006	1.046	1.102	14.15
0.01	0.009	0.014	0.066	0.058	0.145	0.902	0.953	0.993	1.019	1.062	8.888
0.1	0.010	0.014	0.033	0.034	0.200	0.909	0.954	0.993	1.022	1.069	14.72
1	0.035	0.038	0.013	0.013	0.195	0.645	0.675	0.998	1.034	1.093	25.38
10	0.076	0.076	0.007	0.007	0.000	0.405	0.641	0.987	1.056	1.124	3.099
100	0.863	0.860	0.004	0.004	0.054	0.389	0.558	1.708	2.641	2.893	23.27
1000	2.089	2.085	0.002	0.002	0.051	0.398	0.614	3.002	5.473	6.665	61.02

Table 6.7: Performance profiles of verifiable PINNs with 50-50 and 600-600-600 architectures for various λ values at epoch 1000 with the $L_{\text{phy,verifiable}}$ loss function using 700k training and 150k validation sets.

Visual representation of results for λ variations for verifiable architectures

These results can also be visualised for the verifiable 50-50 architecture as shown in Figure 6.8 and the verifiable 600-600-600 architecture as shown in Figure 6.9. A deteriation of performance for $\lambda > 1$ is once again clearly visible.

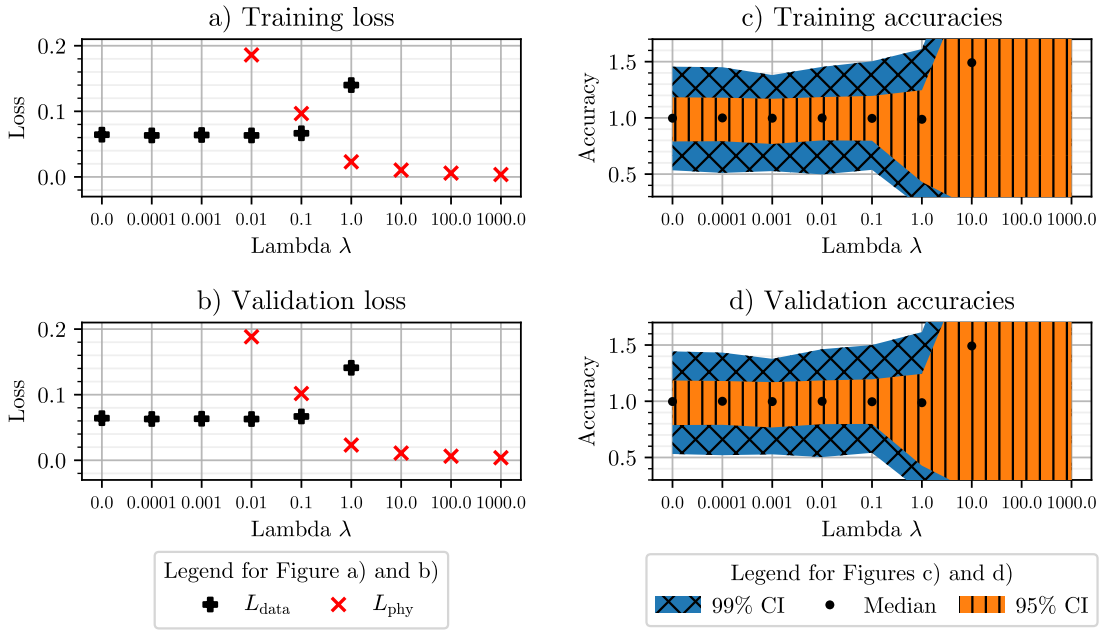


Figure 6.8: Visual performance profiles for verifiable 50-50 architecture at epoch 1000 with various $L_{\text{data}} + \lambda L_{\text{phy,verifiable}}$ loss function using 700k training and 150k validation sets.

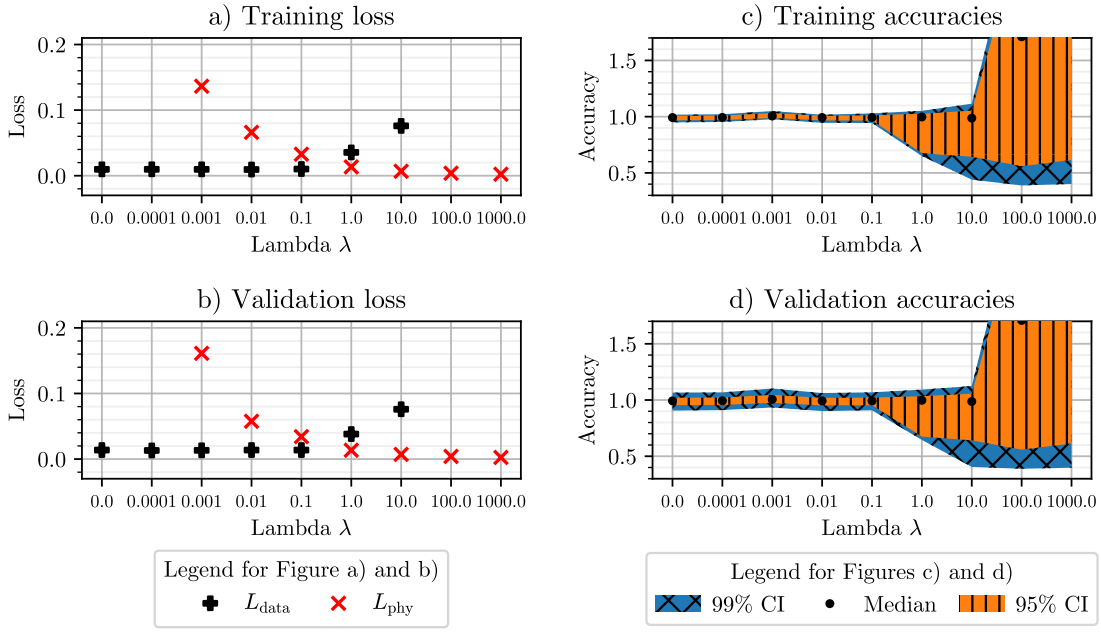


Figure 6.9: Visual performance profiles for verifiable 600-600-600 architecture at epoch 1000 with various $L_{\text{data}} + \lambda L_{\text{phy,verifiable}}$ loss function using 700k training and 150k validation sets.

Performance comparison for training, validation and testing data for $\lambda = 0.1$ models with full training graphs for verifiable architectures

Similarly to before, the best performing PINN in terms of balancing physics and data losses occurred close to $\lambda = 0.1$ values for both the 50-50 and 600-600-600 verifiable architectures. The full performance profile of a data-only architecture ($\lambda = 0$) and the verifiable architecture with $\lambda = 0.1$ is shown in Table 6.8, with performance profiles shown in Figure 6.10 and 6.11.

Model	Dataset	L_{data}	Accuracy Percentiles						
			Min	0.5%	2.5%	Median	97.5%	99.5%	Max
Verifiable 50-50	Train.	0.064	0.023	0.534	0.791	0.997	1.183	1.455	21.83
model with	Val.	0.064	0.026	0.531	0.788	0.997	1.185	1.444	25.09
$\lambda = 0.0$	Test.	0.064	0.025	0.532	0.792	0.997	1.185	1.470	18.63
Verifiable 50-50	Train.	0.067	0.035	0.538	0.797	0.996	1.195	1.503	19.059
model with	Val.	0.067	0.034	0.542	0.798	0.996	1.195	1.500	47.621
$\lambda = 0.1$	Test.	0.066	0.037	0.539	0.796	0.995	1.198	1.510	21.85
Verifiable	Train.	0.010	0.548	0.949	0.968	0.992	1.008	1.018	1.809
600-600-600 model	Val.	0.014	0.272	0.906	0.954	0.992	1.020	1.066	12.42
with $\lambda = 0.0$	Test.	0.013	0.248	0.907	0.954	0.992	1.020	1.066	3.069
Verifiable	Train.	0.010	0.639	0.943	0.965	0.992	1.013	1.029	1.345
600-600-600 model	Val.	0.014	0.200	0.909	0.954	0.993	1.022	1.069	14.73
with $\lambda = 0.1$	Test.	0.013	0.199	0.908	0.954	0.993	1.022	1.070	5.232

Table 6.8: Loss and accuracy profiles for 600-600-600 $a_{\text{in,ReLU}}$ and $a_{\text{out,exp}}$ network at epoch 1000 with $L_{\text{data}} + \lambda L_{\text{phy,verifiable}}$, 700k training, 150k validation sets and 150k testing set.

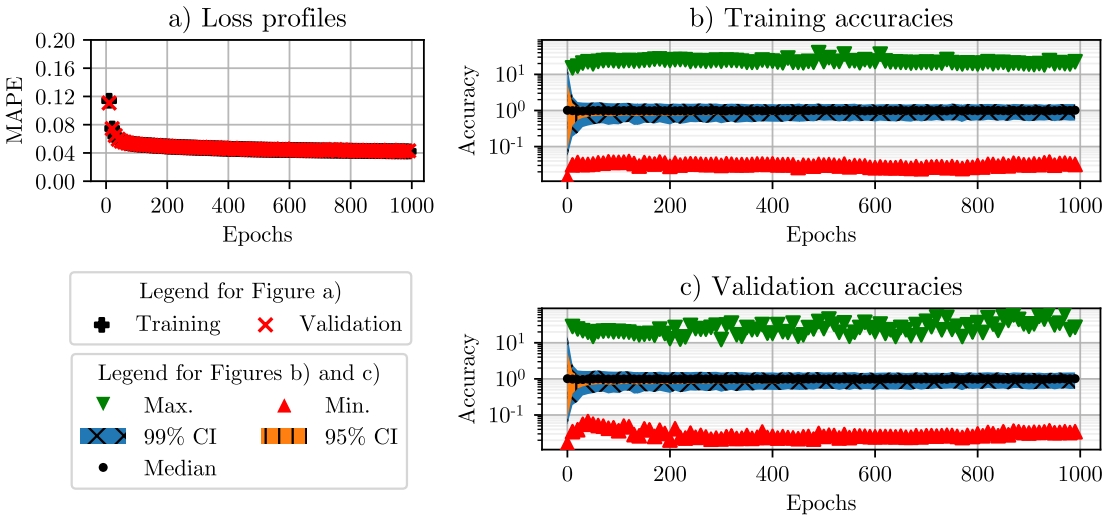


Figure 6.10: Visual performance profiles for verifiable 50-50 architecture at epoch 1000 with $L_{\text{data}} + \lambda L_{\text{phy,verifiable}}$ loss function combinations using 700k training and 150k validation sets with $\lambda = 0.1$.

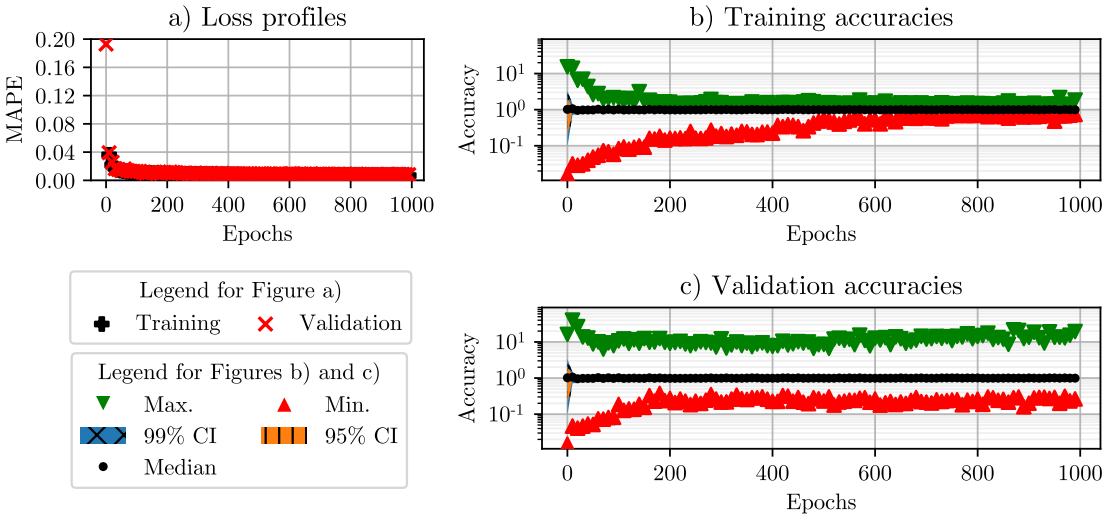


Figure 6.11: Visual performance profiles for verifiable 600-600-600 architecture at epoch 1000 with $L_{\text{data}} + \lambda L_{\text{phy,verifiable}}$ loss function combinations using 700k training and 150k validation sets with $\lambda = 0.1$.

6.4 Discussion

6.4.1 Identifying appropriate λ values

One key theme across all networks trained is that the performance of the PINNs tend to deteriorate when λ exceeds a value of 1. This pattern was clearly observable in Figure 6.3 across most physics loss functions, including $L_{\frac{\text{phy}-\text{true}}{\text{true}}}$ and $L_{\frac{\text{pred}-\text{phy}}{\text{phy}}}$, demonstrating this was not an issue related to the $L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ or $L_{\text{phy,verifiable}}$ loss equations identified by Equations 6.13 and 6.19. The deterioration in performance was identified in both the standard and verifiable architectures, however with slightly different behaviour. In both cases, values with $\lambda > 1$ resulted in a broadening of the accuracy percentiles, which is representative of worse performance. However, whilst both L_{data} and L_{phy} losses worsened with increasing values of λ for standard architectures as shown in Figure 6.6 and 6.7, increasing λ only increased the L_{data} errors, yet decreased the physics-based $L_{\text{phy,verifiable}}$ errors for the verifiable architectures.

This behaviour may be attributable to the fact that the physics loss function used to constrain this structural design problem is ill-posed and highly non-unique, which is a core issue of inverse problems in general [140]. When large λ values are used to constrain this structural design problem, the increased relative magnitude of the L_{phy} error is likely to encourage the neural network parameters to find the closest viable physics solution to the problem, without necessarily finding the one identified by L_{data} that constraints towards the true data-point. This suggests that the L_{data} term is critical in "nudging" the solution to the correct answer. Alternatively, more constraints might be required (which did exist in the original data generation stage of the CBeamXP dataset, namely in terms of class of cross-sections allowed and strict dimensional ratios between the flanges and webs of the

I-sections, see Table 5.2). This highlights the central theme of inverse structural design problems: that physical models alone provide only partial prior information.

This would also ultimately explain why the PINNs with $\lambda = 0.1$ had in general the best performance, since this approached the limit where the physics losses L_{phy} informed the neural network slightly towards physically realistic predictions without dominating the loss magnitude to the extent that it would completely diverge from the true datapoint. λ values between 0.1 and 1.0 might have shown even better performance. This is both suggestive of the limits of using a local stiffness based formulation, but also helps to identify avenues for improvement, by providing either more constraints or perhaps moving towards a global formulation.

Compared to literature, it is worth noting that since the majority of previously identified literature has predominantly dealt with data-less PINN formulations [94, 512, 524] and therefore did not face an issue with choosing an appropriate λ value. The only example of a PINN based ML inverse operator that did use λ values [523] tested a completely different PINN formulation based on a data term that is combined with residual of equilibrium partial differential equations and boundary condition constraints, with λ fixed for the PDE term at a value of one and varied between $\lambda = 1, 3, 10$ for the boundary conditions. The impact of varying λ terms in that particular research was minimal. Further research is required to therefore understand the a correct choice for the λ value for PINNs based on the local stiffness relation presented here.

6.4.2 Performance improvements of standard architectures

One of the fundamental aims of this investigation was to evaluate if the proposed PINN formulation improves the performance of ML structural design model when compared against its data-only driven equivalent (that is when $\lambda = 0$). The results based on the standard architecture with $\lambda = 0.1$ shown in Table 6.5 indicate that the local formulation only decreases the L_{data} error (which is equivalent to the MAPE error reported in Chapter 5) slightly for the standard 50-50 architecture from 6.6% to 5.4% for the testing dataset. Whilst there are also improvements observed in the accuracy percentile values, these improvements are not as large as anticipated.

Furthermore, for the larger standard 600-600-600 architecture, providing the physical constraints actually deteriorated both the L_{data} and accuracy bounds, with only slight improvements measured at the 97.5% and 99.5% percentile bounds. This is also reflective in the testing dataset when subjected to the decile analysis conducted in the previous chapter in Section 5.15 as shown in Figure 6.12 below. Whilst there is some improvement in terms of the standard deviation of standard deviations $\sigma(\sigma_{\text{Deciles}})$, it is still the largest for data within the first decile of total load $\omega_0 \times L_0$. The heatmap shown in Figure 6.13 reveals a similar distribution of errors when compared to the previous chapter shown in Figure 5.16.

Whilst the improvement of the accuracy bounds for the standard 50-50 architecture is an encouraging sign, it is clear that supplementing PINN to a previously designed architecture is not by itself sufficient to drastically improve its performance. This may perhaps be another indication of the limitations of using local stiffness relations only to physically inform the network. It is possible that a global formulation is instead desirable,

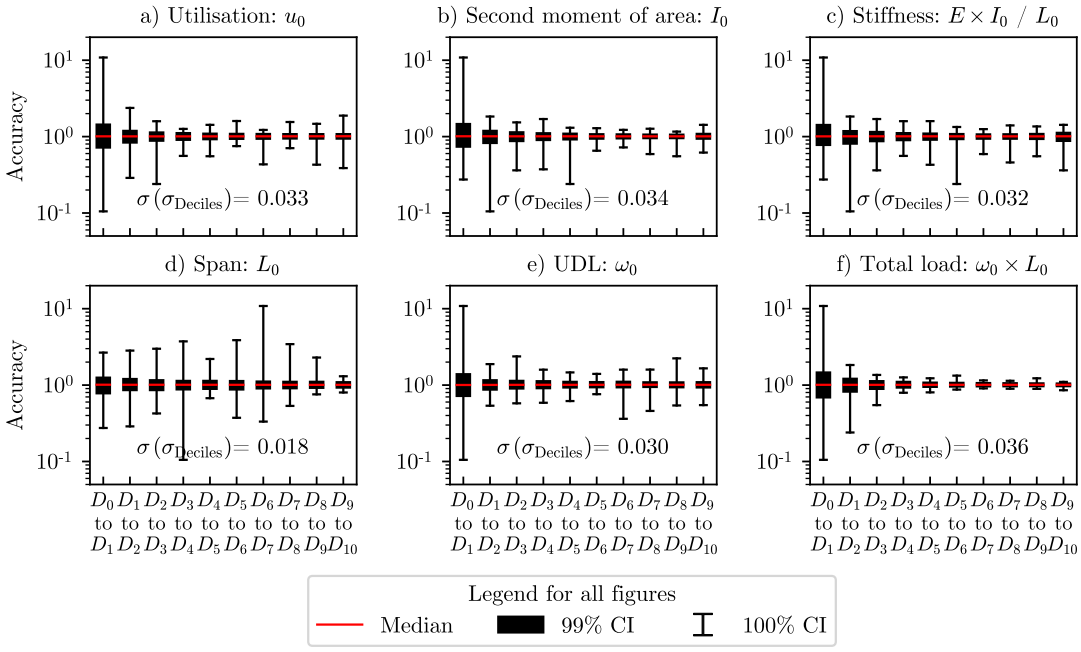


Figure 6.12: Decile analysis of 150k testing dataset with the standard 600-600-600 architecture at epoch 1000 with $L_{\text{data}} + L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ loss function and $\lambda = 0.1$.

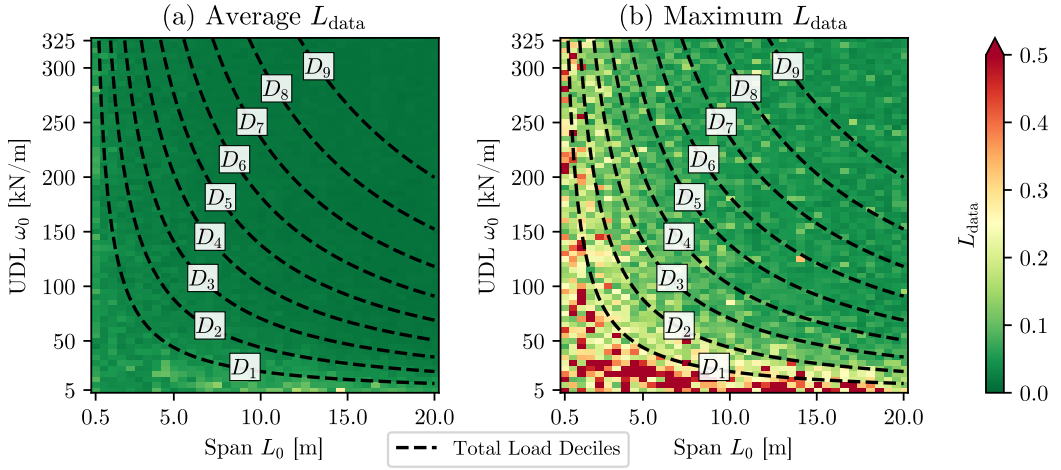


Figure 6.13: Heatmap for 150k testing dataset performance with standard 600-600-600 architecture at epoch 1000 with $L_{\text{data}} + L_{\frac{\text{phy}-\text{pred}}{\text{pred}}}$ loss function and $\lambda = 0.1$.

especially when considering the fact that the influence zone of such a system has been shown to be up to $k_{\max} = 5$ [441]. Global stiffness relations that reflect the adjacent stiffness information of the system would constrain the solution space better, and thereby should help address the error dispersions associated with low total load values.

6.4.3 Viability of verifiable architectures

The second aim of this investigation was to propose a verifiable PINN architecture that would allow a user to calculate the physical consistency of a prediction based on the local stiffness relations at inference. Similar to the standard architectures, the best performing PINNs based on the verifiable architecture was with $\lambda = 0.1$ as clearly demonstrated by Figures 6.8 and 6.9. However, there was no performance improvement (although no deterioration either) in terms of data error L_{data} or accuracy profiles with $\lambda = 0.1$ compared to $\lambda = 0$ as shown in Table 6.8. This is further evidence that the local stiffness relations may be non-optimal for the problem at hand.

Since the verifiable architecture predicts all of the required structural information including utilisation ratios and the rotations at the support of each beam, it is possible to evaluate the difference between the predicted cross-sectional properties of the network with the physics-derived ones via Equation 6.14. This means that some insight on how physically consistent a prediction can now be evaluated for *any* input. Whilst the level of physical consistency is of interest, in an ideal scenario, physical consistency should also correlate with lower prediction error. To this end, the correlation between the 150k training L_{data} and $L_{\text{phy,verifiable}}$ losses for each verifiable 50-50 and 600-600-600 architecture and for all λ values tested were evaluated, with results of this analysis shown in Figure 6.16.

Figure 6.16 indicates that while the correlation between these two sets of errors grows as λ approaches a value of 1, it never exceeds a coefficient of determination of $R^2 = 0.12$. The fact that the correlation does increase within this range is encouraging. Additional trials and a increase range of λ values could result in lower error margins. However, overall, increasing λ values decreases $L_{\text{phy,verifiable}}$ at the cost of increasing L_{data} losses. For similar reasons as explained before, this may simply indicate the limitations of the local stiffness relation employed here.

Finally, a decile analysis of the error dispersion for the verifiable 600-600-600 architecture with $\lambda = 0.1$ when tested against the 150k testing dataset is shown in Figure 6.14. Interestingly, a decrease in dispersion of dispersions $\sigma(\sigma_{\text{Deciles}})$ is identified here, with a maximum value of $\sigma(\sigma_{\text{Deciles}}) = 0.025$, which is less than from the standard architectures as shown in 6.12 and when compared to the previous investigation as presented in Chapter 5, Figure 5.15. Whilst the lowest total load combination $\omega_0 \times L_0$ still contains the largest errors, the verifiable architecture does appear to have clearly contributed to reducing the maximum L_{data} errors when compared to Figure 6.13 and that achieved in Chapter 5, Figure 5.16 with the 150k testing dataset.

6.4.4 Limitations and scope for future works

PINNs based on the local stiffness formulation appear to generate only a small increase in performance when compared their data-only equivalents. The discussion has already identified that the existence of non-unique solutions necessitates the data-term unlike previous investigations in literature with data-less formulations [94, 512, 524]. Furthermore, the local stiffness formulation here does not constrain the solution based on the spans and UDL load values from adjacent members, which with knowledge of the influence zone size of the system $k_{\max} = 5$ would point towards the need for a global formulation instead.

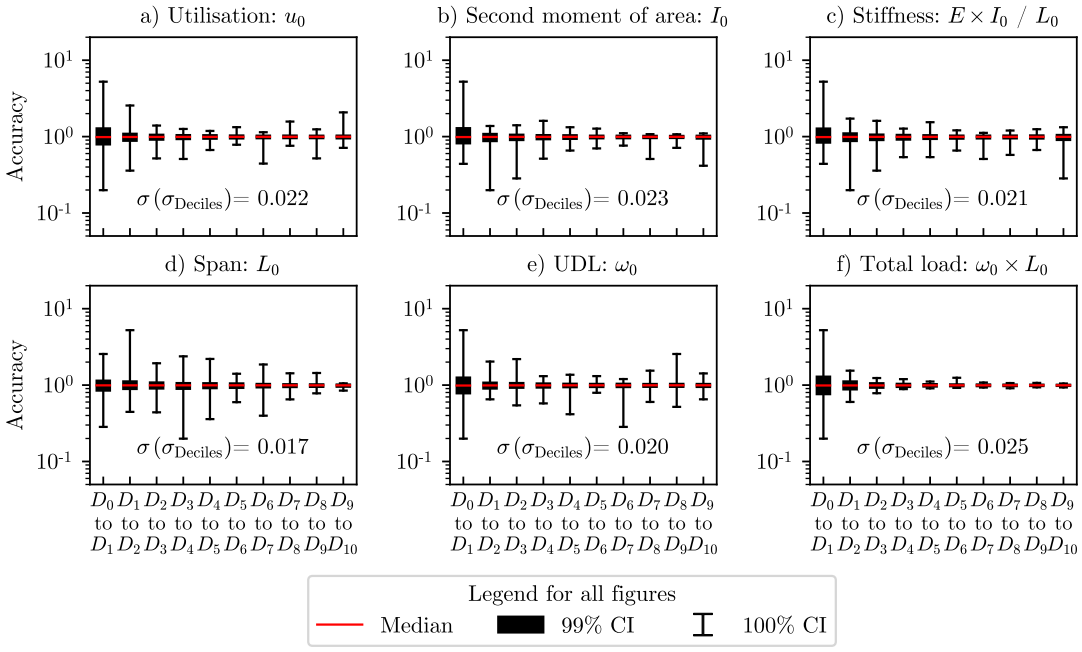


Figure 6.14: Decile analysis of 150k testing dataset with the verifiable 600-600-600 architecture at epoch 1000 with $L_{\text{data}} + L_{\text{phy,verifiable}}$ loss function and $\lambda = 0.1$.

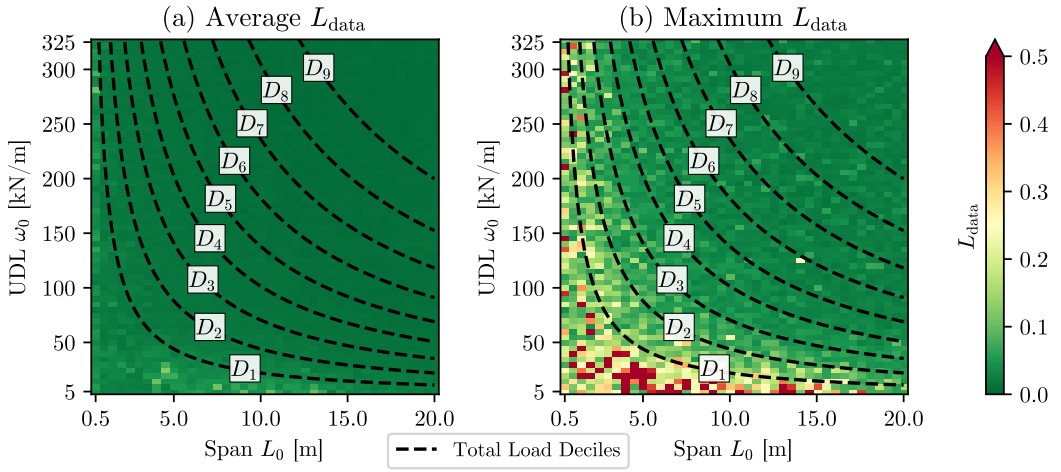


Figure 6.15: Heatmap for 150k testing dataset performance with verifiable 600-600-600 architecture at epoch 1000 with $L_{\text{data}} + L_{\text{phy,verifiable}}$ loss function and $\lambda = 0.1$.

This is identified as the likely scope for future works.

Another limitation of the methodology employed here is that the rotational and utilisation ratio information is evaluated for the full, non-patterned load arrangement, which is never the critical load arrangement. In this investigation, this was chosen to avoid having to potentially factor end-beam reaction forces based on whether the critical load arrangement had an active UDL on the member under investigation, however future studies could

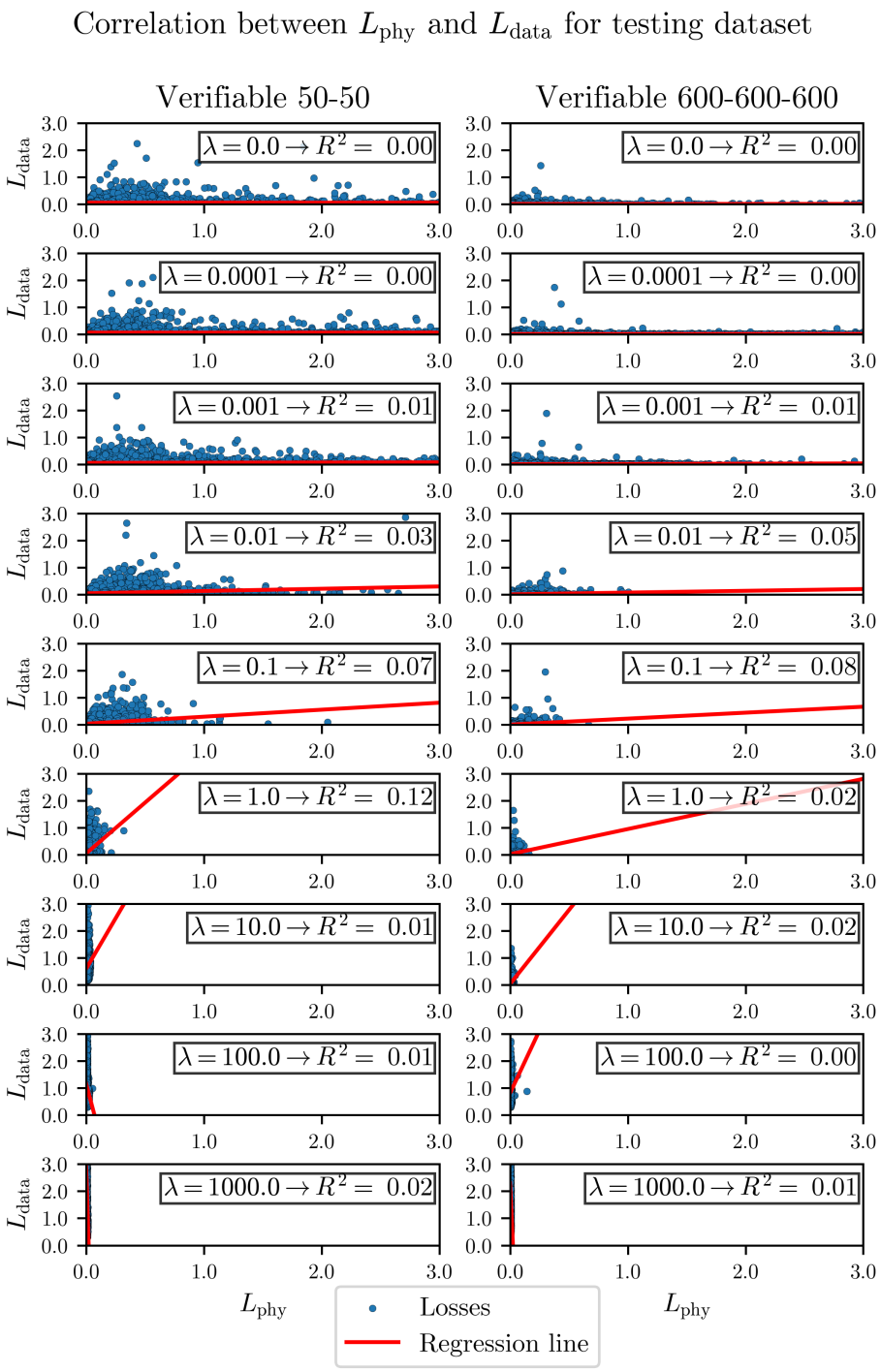


Figure 6.16: Coefficients of determinations between L_{data} and $L_{\text{phy,verifiable}}$ for various verifiable architectures and λ values.

investigate if a proper acknowledgement of the critical load arrangement, especially for potential global stiffness relations, lead to increased performance in the system. Furthermore, if a successful PINN formulation, specifically one in which physics losses correlate with data losses can be identified, then the investigation should be expanded to different structural systems to evaluate the versatility of the methodology employed here.

It is possible that the dataset is already sufficiently balanced and well-spaced to negate the need for a PINN formulation. Previous research suggest that PINNs excel in data-scarce environments, specifically when needing to interpolate across large data gaps [93]. The extent to which this similar behaviour arises within PINNs developed for ill-posed problems remains to be investigated.

Lastly, it is worth identifying that PINNs also carry an associated increased computational cost. The total computational cost of this study was just over 13 days as shown by Table 6.9. When compared to data-only structural design models, a PINN formulation will only be worth the additional computational cost if it yields to significantly improved results.

Section	Stage	Computation time		Proportion [%]
		In hours [h]	In days [D]	
Section 6.3.1	All standard L_{phy} variations	58.54	2.44	
Section 6.3.1	with 10k training dataset	5.64	0.23	18.7
Section 6.3.1	with 100k training dataset	9.93	0.41	
Section 6.3.1	with 700k training dataset	42.97	1.79	
Section 6.3.2	All standard architecture training	88.14	3.67	
Section 6.3.2	Single 50-50 training run	1.25	0.05	28.1
Section 6.3.2	Single 600-600-600 training run	8.55	0.36	
Section 6.3.3	All verifiable architecture training	166.79	6.95	
Section 6.3.3	Single 50-50 training run	1.30	0.05	53.2
Section 6.3.3	Single 600-600-600 training run	17.54	0.73	
Total computation time (Sections 6.3.1, 6.3.2, 6.3.3):		313.47	13.06	100

Table 6.9: Computation time for each physics informed neural network development stage.

6.5 Conclusions

This chapter introduced various physics informed neural network formulations based on local stiffness relations with two explicit aims. The first was to investigate if PINNs are suitable to create structural design models for continuous beam systems with lower prediction errors and error dispersions than of those achieved from data-only models. Whilst the PINNs developed in this investigation did reduce such errors slightly, they were highly sensitive to the λ hyperparameter which controlled the extent to which the physical constraints were enforced during network training. The second aim was to test a structural design model in which design predictions could be verified during inference, which would allow prediction verifications including for unknown inputs. Whilst such a neural network was successfully trained, the physics losses only had a weak correlation

with data losses. Several avenues of future works were identified, the most notable being to develop an equivalent PINN with global stiffness relations.

Part III

Outcomes

Chapter 7

Conclusions

7.1 Review of aims, objectives and novel contributions

The aims of this thesis, as presented in Chapter 1, were to demonstrate the value of the inverse problem perspective for structural design and to highlight the viability of machine-learned design models within this context. These aims were reflected by five objectives listed in Section 1.1 which formed the basis of the five central chapters and four publications presented in this thesis:

- Chapter 2 considered the current structural trends and philosophical underpinnings of structural engineering design to address the first objective, and reviewed the literature on using machine learning to address design challenges faced in industry.
- Chapter 3 proposed the *inverse problem perspective for structural design* as set out by the second objective, and reviewed other research areas in which inverse problems are prevalent and for which machine learning models were developed as inverse operators. This perspective laid the foundations for Chapters 5 and 6 from which such learned operators were investigated in full.
- A novel concept known as the *influence zone* was established in Chapter 4 to identify the pertinent structural information required to accurately design individual elements within a continuous beam system. This concept directly informed the required feature space to help create generalisable design models researched in Chapters 5 and 6 as set out by the third objective. The work also introduced other novel techniques, including *polarity zones* and *polarity sequences*, and provided a rigorous methodology to help identify critical load arrangements of such systems.
- The fourth objective was achieved by Chapter 5, in which a machine learned structural design model for continuous beam systems was developed based on the multilayer perceptron network architecture, with a focus on using the influence zone concept to create a *generalisable* model capable of being applied to structures of arbitrary system size. This also entailed a detailed analysis of various loss and activation function combinations, as well as variations of network size on the errors and accuracy profiles of the predictions. A further contribution from this particular work

was the creation of the **C**ontinuous **Beam** Cross-sections (**X**) **P**redictor dataset, or simply known as *CBeamXP*. This dataset contains one million design examples of beam elements within a continuous system, and has been made available at an open-source data repository [6] to allow validation of the results achieved in this study, and to encourage further research efforts within this field.

- Finally, a design model based on a *physics informed neural network* using local stiffness relations was developed in Chapter 6 by building on the research results established in Chapter 5, which addressed the fifth and final objective of the thesis. Two various architectures were developed to check the PINN's ability to reduce errors dispersions and the network's ability to verify predictions during inference, both of which helped to identify likely areas of future works.

7.2 Future works and concluding remarks

There is significant scope for future research given the number of aforementioned contributions generated through the thesis's investigative efforts. Individual scopes for future research have already been provided at the end of each article presented in Chapters 3, 4, 5 and 6. A key aim of any future work will be the broadening agreement within both the structural engineering and mathematical disciplines to identify structural design as an inverse problem. This should hopefully encourage non-structural engineers familiar with the techniques commonly applied to study and assess inverse problems into the research field of structural design models, motivate efforts to create practical machine learned structural design models and further refine the philosophical nature of design.

Whilst only one particular structural system was investigated in this thesis, it should be relatively easy to expand the development of design models to different systems (including trusses, frames, arches, etc.) based on the existing methodology developed in the courses of these chapters. It is likely that the expansion of structural systems in terms of dimensions (2D and 3D) and complexity (composite materials, design connections) will require further new concepts that either directly expand or mirror the influence zone concept's role to help develop generalisable models. Addressing the need for rapid, accurate and verifiable models will likely serve for future grounds of research.

Ultimately, such research will contribute to both the invention and discovery of novel ideas, which may manifest itself in better philosophical frameworks for engineering design, approximative design techniques of various systems, open-source datasets for machine learning research and improved machine learning modalities. Such efforts, in addition to contributing to the vast intellectual body of knowledge, will help address some of the real and practical challenges faced in industry, and thereby continue to push previous achievements of engineering practice forward.

References

[1] S. L. Goldman, Why we need a philosophy of engineering: a work in progress, Interdisciplinary Science Reviews 29 (2) (2004) 163–176. doi:10.1179/030801804225012572.

[2] R. Descartes, É. Gilson, Discours de la méthode, Vrin, 1987.

[3] K. Popper, The logic of scientific discovery, Routledge, 2005.

[4] W. Bulleit, J. Schmidt, I. Alvi, E. Nelson, T. Rodriguez-Nikl, Philosophy of engineering: What it is and why it matters, Journal of Professional Issues in Engineering Education and Practice 141 (3) (2015) 02514003. doi:10.1061/(ASCE)EI.1943-5541.0000205.

[5] M. Raissi, P. Perdikaris, G. 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 (2019) 686–707. doi:10.1016/j.jcp.2018.10.045.

[6] A. Gallet, D. Smyl, CBeamXP: Continuous Beam Cross-section Predictors dataset (2023). doi:10.15131/shef.data.23945562.
URL https://figshare.com/s/c84c620925c1c36e648c

[7] A. Harris, Civil engineering considered as an art, Proceedings of the Institution of Civil Engineers 58 (1) (1975) 15–23.

[8] W. Addis, Structural engineering: The nature of theory and design, Ellis Horwood Limited, 1990.

[9] F. Bacon, Novum organum, Clarendon press, 1878.

[10] T. Kuhn, The structure of scientific revolutions, Princeton University Press, 2021.

[11] B. V. Koen, Discussion of the Method: Conducting the Engineer's Approach to Problem Solving, Oxford University Press, New York, 2003.

[12] R. Morledge, Building procurement [electronic resource], 2nd Edition, Wiley-Blackwell, Chichester, West Sussex, U.K., 2013.

[13] C. Wise, Introduction: Part 1 Structure is everywhere!, 1st Edition, The Institution of Structural Engineers, London, 2016.

[14] J. Mason, M. Manning, A. Cormie, J. Johnston, A. Low, C. Merlbourne, R. Nicholl, M. Ryland, F. Weare, Structural design – the engineer's role, 2nd Edition, IStructE Ltd, 2018.

[15] Gov.uk, Construction 2025: industrial strategy: government and industry in partnership (2013).

[16] A. Fordham, The History of the Theory of Structures, The Structural Engineer 16 (5) (1938) 154–163.

[17] A. Mann, Historical development of structural theories and methods of modern computer, Institution of Structural Engineers, 2016.

[18] W. Addis, Building : 3000 years of design engineering and construction, Phaidon Press, London, 2007.

[19] W. Addis, The evolution of structural engineering design procedures: A history for that skill called design, Transactions of the Newcomen Society 61 (1) (1989) 51–64. doi:10.1179/tns.1989.004.

[20] J. Heyman, Structural analysis: a historical approach, 1st Edition, Cambridge University Press, 1998.

[21] C. A. Felippa, A historical outline of matrix structural analysis: A play in three acts, Computers and Structures 79 (14) (2001) 1313–1324. doi:10.1016/S0045-7949(01)00025-6.

[22] M. Antwi-Afari, H. Li, E. Pärn, D. J. Edwards, Critical success factors for implementing building information modelling (bim): A longitudinal review, Automation in construction 91 (2018) 100–110. doi:10.1016/j.autcon.2018.03.010.

[23] C. M. Eastman, C. Eastman, P. Teicholz, R. Sacks, K. Liston, BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors, John Wiley & Sons, 2011.

[24] N. Gu, K. London, Understanding and facilitating bim adoption in the aec industry, Automation in construction 19 (8) (2010) 988–999. doi:10.1016/j.autcon.2010.09.002.

[25] B. Becerik-Gerber, K. Kensek, Building information modeling in architecture, engineering, and construction: Emerging research directions and trends, Journal of professional issues in engineering education and practice 136 (3) (2010) 139–147. doi:10.1061/(ASCE)EI.1943-5541.0000023.

[26] Ş. T. Pektaş, M. Pultar, Modelling detailed information flows in building design with the parameter-based design structure matrix, Design Studies 27 (1) (2006) 99–122. doi:10.1016/j.destud.2005.07.004.

[27] M.-L. Chiu, J.-H. Lan, Information and in-formation: information mining for supporting collaborative design, Automation in Construction 14 (2) (2005) 197–205. doi:10.1016/j.autcon.2004.07.011.

[28] M.-H. Construction, Smartmarket report: the business value of bim in north america, Bedford: McGraw-Hill Construction Research and Analytics (2012).

[29] P. Bynum, R. R. Issa, S. Olbina, Building information modeling in support of sustainable design and construction, Journal of construction engineering and management 139 (1) (2013) 24–34. doi:0.1061/(ASCE)CO.1943-7862.0000560.

[30] R. Rempling, A. Mathern, D. T. Ramos, S. L. Fernández, Automatic structural design by a set-based parametric design method, Automation in Construction 108 (2019) 102936. doi:10.1016/j.autcon.2019.102936.

[31] M. Turrin, P. Von Buelow, R. Stouffs, Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms, Advanced Engineering Informatics 25 (4) (2011) 656–675. doi:10.1016/j.aei.2011.07.009.

[32] Robert McNeel & Associates, Grasshopper, www.grasshopper3d.com, version 1.0.0007 (2021-06-09).

[33] Autodesk Inc, Dynamorevit, www.dynamobim.org, version 2.3 (2020-12-07).

[34] Karamba 3D, Karamba3d, www.karamba3d.com, version 1.3.3 (2020-10-12).

[35] str.ucture GmbH, Kiwi!3d, www.kiwi3d.com, version Beta 0.5.0 (2020-10-01).

[36] R. Sacks, R. Barak, Impact of three-dimensional parametric modeling of buildings on productivity in structural engineering practice, Automation in Construction 17 (4) (2008) 439–449. doi:10.1016/j.autcon.2007.08.003.

[37] Y.-C. Lee, C. M. Eastman, J.-K. Lee, Validations for ensuring the interoperability of data exchange of a building information model, Automation in Construction 58 (2015) 176–195. doi:10.1016/j.autcon.2015.07.010.

[38] M. Gallaher, A. C. O'Connor, J. Dettabarn, L. Gilday, Cost analysis of inadequate operability in the us capital facilities industry, National Institute of Standards and Technology (2004). doi:10.6028/NIST.GCR.04-867.

[39] C. Eastman, J. Teizer, M. Venugopal, R. Sacks, Improving the robustness of model exchanges using product modeling" concepts" for ifc schema, in: Computing in Civil Engineering (2011), ASCE International Workshop on Computing in Civil Engineering, 2011, pp. 611–618. doi:10.1061/41182(416)75.

[40] N. W. Young, S. A. Jones, H. M. Bernstein, J. Gudgel, et al., The business value of bim-getting building information modeling to the bottom line (2009).

[41] Building Habitats Object Model, Bhom, www.bhom.xyz (2021-07-22).

[42] Aec Systems Ltd, Speckle, www.speckle.systems (2021-07-22).

[43] Geometry Gym, Geometry gym, www.geometrygym.wordpress.com (2021-07-22).

[44] M. P. Bendsoe, O. Sigmund, Topology optimization: theory, methods, and applications, Springer Science & Business Media, 2013. doi:10.1007/978-3-662-05086-6.

[45] A. Rothwell, Optimization methods in structural design, Vol. 242, Springer, 2017. doi:10.1007/978-3-319-55197-5.

[46] J. S. Arora, Introduction to optimum design, Elsevier, 2004.

[47] Y. Feng, C. Wang, B. Briseghella, L. Fenu, T. Zordan, Structural optimization of a steel arch bridge with genetic algorithm, Structural Engineering International (2020) 1–10doi:10.1080/10168664.2020.1773373.

[48] J. J. Kingman, K. Tsavdaridis, V. Toropov, Applications of topology optimisation in structural engineering: high-rise buildings & steel components, Jordan Journal of Civil Engineering 9 (3) (2015) 335–357.

[49] I. Heitel, F. Fu, Form finding and structural optimisation of tensile cable dome using parametric modelling tools, Structural Engineering International 31 (2) (2021) 271–280. doi:10.1080/10168664.2020.1750937.

[50] M. Hamdy, A.-T. Nguyen, J. L. Hensen, A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems, Energy and Buildings 121 (2016) 57–71. doi:10.1016/j.enbuild.2016.03.035.

[51] RIBA, Plan of work 2020 overview (2020).

[52] A. S. Oyegoke, M. Dickinson, M. M. Khalfan, P. McDermott, S. Rowlinson, Construction project procurement routes: an in-depth critique, International Journal of Managing Projects in Business (2009). doi:10.1108/17538370910971018.

[53] T. Ibn-Mohammed, R. Greenough, S. Taylor, L. Ozawa-Meida, A. Acquaye, Operational vs. embodied emissions in buildings — a review of current trends, Energy and Buildings 66 (2013) 232–245. doi:10.1016/j.enbuild.2013.07.026.

[54] B. D. Rogers, Efficient design in a modern engineering world–the challenges (2017). doi:10.1680/jencm.2017.170.3.89.

[55] M. Bilal, L. O. Oyedele, J. Qadir, K. Munir, S. O. Ajayi, O. O. Akinade, H. A. Owolabi, H. A. Alaka, M. Pasha, Big data in the construction industry: A review of present status, opportunities, and future trends, Advanced engineering informatics 30 (3) (2016) 500–521. doi:10.1016/j.aei.2016.07.001.

[56] V. Ahmed, A. Tezel, Z. Aziz, M. Sibley, The future of big data in facilities management: opportunities and challenges, Facilities (2017). doi:10.1108/F-06-2016-0064.

[57] V. Ahmed, Z. Aziz, A. Tezel, Z. Riaz, Challenges and drivers for data mining in the aec sector, Engineering, Construction and Architectural Management (2018). doi:10.1108/ECAM-01-2018-0035.

[58] C. McMahon, A. Lowe, S. Culley, Knowledge management in engineering design: personalization and codification, Journal of Engineering Design 15 (4) (2004) 307–325. doi:10.1080/09544820410001697154.

[59] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015) 85–117. doi:10.1016/j.neunet.2014.09.003.

[60] Y. Duan, J. S. Edwards, Y. K. Dwivedi, Artificial intelligence for decision making in the era of Big Data – evolution, challenges and research agenda, International Journal of Information Management 48 (2019) 63–71. doi:10.1016/j.ijinfomgt.2019.01.021.

[61] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv (2018). doi:10.48550/ARXIV.1810.04805.

[62] H. Sun, H. V. Burton, H. Huang, Machine learning applications for building structural design and performance assessment: state-of-the-art review, Journal of Building Engineering 33 (2020) 101816. doi:10.1016/j.jobe.2020.101816.

[63] P. Debney, Computational Engineering, version 1.1 Edition, Institution of Structural Engineers, London, 2021.

[64] E. L. Jacobsen, J. Teizer, Deep Learning in Construction: Review of Applications and Potential Avenues, Journal of Computing in Civil Engineering 36 (2) (2022) 03121001. doi:10.1061/(ASCE)CP.1943-5487.0001010.

[65] L. Sun, Z. Shang, Y. Xia, S. Bhowmick, S. Nagarajaiah, Review of Bridge Structural Health Monitoring Aided by Big Data and Artificial Intelligence: From Condition Assessment to Damage Detection, Journal of Structural Engineering 146 (5) (2020) 04020073. doi:10.1061/(ASCE)ST.1943-541X.0002535.

[66] Y.-A. Hsieh, Y. J. Tsai, Machine Learning for Crack Detection: Review and Model Performance Comparison, Journal of Computing in Civil Engineering 34 (5) (2020) 04020038. doi:10.1061/(ASCE)CP.1943-5487.0000918.

[67] X. Wang, R. K. Mazumder, B. Salarieh, A. M. Salman, A. Shafieezadeh, Y. Li, Machine Learning for Risk and Resilience Assessment in Structural Engineering: Progress and Future Trends, Journal of Structural Engineering 148 (8) (2022) 03122003. doi:10.1061/(ASCE)ST.1943-541X.0003392.

[68] G. C. Marano, M. M. Rosso, A. Aloisio, G. Cirrincione, Generative adversarial networks review in earthquake-related engineering fields, Bulletin of Earthquake Engineering (Feb. 2023). doi:10.1007/s10518-023-01645-7.

[69] M. Mohtasham Moein, A. Saradar, K. Rahmati, S. H. Ghasemzadeh Mousavinejad, J. Bristow, V. Aramali, M. Karakouzian, Predictive models for concrete properties using machine learning and deep learning approaches: A review, Journal of Building Engineering 63 (2023) 105444. doi:10.1016/j.jobe.2022.105444.

[70] H.-T. Thai, Machine learning for structural engineering: A state-of-the-art review, Structures 38 (2022) 448–491. doi:10.1016/j.istruc.2022.02.003.

[71] R. Falcone, C. Lima, E. Martinelli, Soft computing techniques in structural and earthquake engineering: A literature review, Engineering Structures 207 (2020) 110269. doi:10.1016/j.engstruct.2020.110269.

[72] H. Adeli, Neural Networks in Civil Engineering: 1989–2000, Computer-Aided Civil and Infrastructure Engineering 16 (2) (2001) 126–142. doi:10.1111/0885-9507.00219.

[73] C. Málaga-Chuquitaype, Machine Learning in Structural Design: An Opinionated Review, Frontiers in Built Environment 8 (2022) 815717. doi:10.3389/fbuil.2022.815717.

[74] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the national academy of sciences 79 (8) (1982) 2554–2558. doi:10.1073/pnas.79.8.2554.

[75] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, nature 323 (6088) (1986) 533–536. doi:10.1038/323533a0.

[76] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) (1996) 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.

[77] A. E. Hoerl, R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67. doi:10.1080/00401706.1970.10488634.

[78] J. H. Friedman, Multivariate adaptive regression splines, The annals of statistics (1991) 1–67doi:10.1214/aos/1176347963.

[79] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and regression trees, Routledge, 2017. doi:10.1201/9781315139470.

[80] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.

[81] Y. S. Abu-Mostafa, M. Magdon-Ismail, H.-T. Lin, Learning from data, Vol. 4, AMLBook New York, NY, USA:, 2012.

[82] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (3) (1995) 273–297. doi:10.1007/BF00994018.

[83] J. H. Friedman, et al., Flexible metric nearest neighbor classification, Tech. rep., Citeseer (1994).

[84] Y. Reich, Machine learning techniques for civil engineering problems, Computer-Aided Civil and Infrastructure Engineering 12 (4) (1997) 295–310. doi:10.1111/0885-9507.00065.

[85] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[86] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, Neural computation 29 (9) (2017) 2352–2449. doi:10.1162/neco_a_00990.

[87] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE transactions on neural networks 20 (1) (2008) 61–80. doi:10.1109/TNN.2008.2005605.

[88] W. Fan, Y. Chen, J. Li, Y. Sun, J. Feng, H. Hassanin, P. Sareh, Machine learning applied to the design and inspection of reinforced concrete bridges: Resilient methods and emerging applications, in: Structures, Vol. 33, Elsevier, 2021, pp. 3954–3963. doi:10.1016/j.istruc.2021.06.110.

[89] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need (2017). doi:10.48550/arXiv.1706.03762.

[90] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar, Theory-guided data science: A new paradigm for scientific discovery from data, IEEE Transactions on knowledge and data engineering 29 (10) (2017) 2318–2331. doi:10.1109/TKDE.2017.2720168.

[91] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, N. Ramakrishnan, Incorporating prior domain knowledge into deep neural networks, in: 2018 IEEE international conference on big data (big data), IEEE, 2018, pp. 36–45. doi:10.1109/BigData.2018.8621955.

[92] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, Science China Technological Sciences 63 (10) (2020) 1872–1897. doi:10.1007/s11431-020-1647-3.

[93] J. J. Pannell, S. E. Rigby, G. Panoutsos, Physics-informed regularisation procedure in neural networks: An application in blast protection engineering, International Journal of Protective Structures 13 (3) (2022) 555–578. doi:10.1177/20414196211073501.

[94] LH. Song, C. Wang, JS. Fan, HM. Lu, Elastic structural analysis based on graph neural network without labeled data, Computer-aided civil and infrastructure engineering 38 (10) (2023) 1307–1323. doi:10.1111/mice.12944.

[95] AISC, 2024 AISC Milek Fellowship awarded to Mohannad Z. Nasar, https://www.aisc.org/modernsteel/news/2023/december/2024-aisc-milek-fellowship-awarded-to-mohannad-z.-naser/ (Dec. 2023).

[96] M. E. Ororbia, G. P. Warn, Design synthesis through a markov decision process and reinforcement learning framework, Journal of Computing and Information Science in Engineering 22 (2) (2021) 021002. doi:10.1115/1.4051598.

[97] R. Vanluchene, R. Sun, Neural networks in structural engineering, Computer-Aided Civil and Infrastructure Engineering 5 (3) (1990) 207–215. doi:10.1111/j.1467-8667.1990.tb00377.x.

[98] H.-T. Kang, C. J. Yoon, Neural network approaches to aid simple truss design problems, Computer-Aided Civil and Infrastructure Engineering 9 (3) (1994) 211–218. doi:10.1111/j.1467-8667.1994.tb00374.x.

[99] S. Tseranidis, N. C. Brown, C. T. Mueller, Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures, Automation in Construction 72 (2016) 279–293. doi:10.1016/j.autcon.2016.02.002.

[100] M. M. Behzadi, H. T. Ilieş, Real-Time Topology Optimization in 3D via Deep Transfer Learning, Computer-Aided Design 135 (2021) 103014. doi:10.1016/j.cad.2021.103014.

[101] E. Whalen, C. Mueller, Toward Reusable Surrogate Models: Graph-Based Transfer Learning on Trusses, Journal of Mechanical Design 144 (2) (2022) 021704. doi:10.1115/1.4052298.

[102] K. S. Ochoa, P. O. Ohlbrock, P. D'Acunto, V. Moosavi, Beyond typologies, beyond optimization: Exploring novel structural forms at the interface of human and machine intelligence, International Journal of Architectural Computing 0 (0) (2020) 1478077120943062. doi:10.1177/1478077120943062.

[103] H. Zheng, V. Moosavi, M. Akbarzadeh, Machine learning assisted evaluations in structural design and construction, Automation in Construction 119 (2020) 103346. doi:https://doi.org/10.1016/j.autcon.2020.103346.

[104] M. Wang, E. Poulsen, D. Reynolds, R. Otani, Conceptual stage predictive and collaborative design using machine learning, in: Proceedings of IASS Annual Symposia, Vol. 2, International Association for Shell and Spatial Structures (IASS), 2018, pp. 1–7.

[105] J. Seo, L. Dueñas-Osorio, J. I. Craig, B. J. Goodno, Metamodel-based regional vulnerability estimate of irregular steel moment-frame structures subjected to earthquake events, Engineering Structures 45 (2012) 585–597. doi:10.1016/j.engstruct.2012.07.003.

[106] F. K. G. Jough, S. Şensoy, Prediction of seismic collapse risk of steel moment frame mid-rise structures by meta-heuristic algorithms, Earthquake Engineering and Engineering Vibration 15 (4) (2016) 743–757. doi:10.1007/s11803-016-0362-9.

[107] C. C. Mitropoulou, M. Papadrakakis, Developing fragility curves based on neural network ida predictions, Engineering Structures 33 (12) (2011) 3409–3421. doi:10.1016/j.engstruct.2011.07.005.

[108] K. Morfidis, K. Kostinakis, Seismic parameters' combinations for the optimum prediction of the damage state of r/c buildings using neural networks, Advances in Engineering Software 106 (2017) 1–16. doi:10.1016/j.advengsoft.2017.01.001.

[109] Y. Zhang, H. V. Burton, H. Sun, M. Shokrabadi, A machine learning framework for assessing post-earthquake structural safety, Structural safety 72 (2018) 1–16. doi:10.1016/j.strusafe.2017.12.001.

[110] S. Moradi, H. V. Burton, Response surface analysis and optimization of controlled rocking steel braced frames, Bulletin of Earthquake Engineering 16 (10) (2018) 4861–4892. doi:10.1007/s10518-018-0373-1.

[111] S. Moradi, H. V. Burton, I. Kumar, Parameterized fragility functions for controlled rocking steel braced frames, Engineering Structures 176 (2018) 254–264. doi:10.1016/j.engstruct.2018.09.001.

[112] H. Sun, H. Burton, J. Wallace, Reconstructing seismic response demands across multiple tall buildings using kernel-based machine learning methods, Structural Control and Health Monitoring 26 (7) (2019) e2359. doi:10.1002/stc.2359.

[113] J.-S. Jeon, A. Shafieezadeh, R. DesRoches, Statistical models for shear strength of rc beam-column joints using machine-learning techniques, Earthquake engineering & structural dynamics 43 (14) (2014) 2075–2095. doi:10.1002/eqe.2437.

[114] H. Luo, S. G. Paal, A locally weighted machine learning model for generalized prediction of drift capacity in seismic vulnerability assessments, Computer-Aided Civil and Infrastructure Engineering 34 (11) (2019) 935–950. doi:10.1111/mice.12456.

[115] N.-D. Hoang, X.-L. Tran, H. Nguyen, Predicting ultimate bond strength of corroded reinforcement and surrounding concrete using a metaheuristic optimized least squares support vector regression model, Neural Computing and Applications 32 (11) (2020) 7289–7309. doi:10.1007/s00521-019-04258-x.

[116] D.-T. Vu, N.-D. Hoang, Punching shear capacity estimation of frp-reinforced concrete slabs using a hybrid machine learning approach, Structure and Infrastructure Engineering 12 (9) (2016) 1153–1161. doi:10.1080/15732479.2015.1086386.

[117] S. Mangalathu, H. Jang, S.-H. Hwang, J.-S. Jeon, Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls, Engineering Structures 208 (2020) 110331. doi:10.1016/j.engstruct.2020.110331.

[118] H. Huang, H. V. Burton, Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning, Journal of Building Engineering 25 (2019) 100767. doi:10.1016/j.jobe.2019.100767.

[119] N.-D. Hoang, Q.-L. Nguyen, X.-L. Tran, Automatic detection of concrete spalling using piecewise linear stochastic gradient descent logistic regression and image texture analysis, Complexity 2019 (2019). doi:10.1155/2019/5910625.

[120] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, O. Büyüköztürk, Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, Computer-Aided Civil and Infrastructure Engineering 33 (9) (2018) 731–747. doi:10.1111/mice.12334.

[121] Y. Gao, K. M. Mosalam, Deep transfer learning for image-based structural damage recognition, Computer-Aided Civil and Infrastructure Engineering 33 (9) (2018) 748–768. doi:10.1111/mice.12363.

[122] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional lstm and other neural network architectures, Neural networks 18 (5) (2005) 602–610. doi:10.1016/j.neunet.2005.06.042.

[123] S. Mangalathu, H. V. Burton, Deep learning-based classification of earthquake-impacted buildings using textual damage descriptions, International journal of disaster risk reduction 36 (2019) 101111. doi:10.1016/j.ijdrr.2019.101111.

[124] H. Sohn, C. R. Farrar, Damage diagnosis using time series analysis of vibration signals, Smart materials and structures 10 (3) (2001) 446. doi:10.1088/0964-1726/10/3/304.

[125] J. P. Lynch, Decentralization of wireless monitoring and control technologies for smart civil structures, Stanford University, 2002.

[126] H. Young Noh, K. Krishnan Nair, D. G. Lignos, A. S. Kiremidjian, Use of wavelet-based damage-sensitive features for structural damage diagnosis using strong motion data, Journal of Structural Engineering 137 (10) (2011) 1215–1228. doi:10.1061/(ASCE)ST.1943-541X.0000385.

[127] E. Figueiredo, A. Santos, Machine learning algorithms for damage detection, in: Vibration-Based Techniques for Damage Detection and Localization in Engineering Structures, World Scientific, 2018, pp. 1–39. doi:10.1142/q0145.

[128] R. Ghiasi, P. Torkzadeh, M. Noori, A machine-learning approach for structural damage detection using least square support vector machine based on a new combinational kernel function, Structural Health Monitoring 15 (3) (2016) 302–316. doi:10.1177/1475921716639587.

[129] L. Chen, A. Gallet, S.-S. Huang, D. Liu, D. Smyl, Probabilistic cracking prediction via deep learned electrical tomography, Structural Health Monitoring 21 (4) (2022) 1574–1589. doi:10.1177/14759217211037236.

[130] Y. Mack, T. Goel, W. Shyy, R. Haftka, Surrogate model-based optimization framework: a case study in aerospace design, in: Evolutionary computation in dynamic and uncertain environments, Springer, 2007, pp. 323–342. doi:10.1007/978-3-540-49774-5_14.

[131] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324. doi:10.1109/5.726791.

[132] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[133] E. M. Rathje, C. Dawson, J. E. Padgett, J.-P. Pinelli, D. Stanzione, A. Adair, P. Arduino, S. J. Brandenberg, T. Cockerill, C. Dey, et al., Designsafe: New cyberinfrastructure for natural hazards engineering, Natural Hazards Review 18 (3) (2017) 06017001. doi:10.1061/(ASCE)NH.1527-6996.0000246.

[134] DataCenterHub, Deeds: a platform for shared data and computing that supports the entire research process, https://datacenterhub.org/ (2021-08-04).

[135] H. V. Burton, S. Sreekumar, M. Sharma, H. Sun, Estimating aftershock collapse vulnerability using mainshock intensity, structural response and physical damage indicators, Structural safety 68 (2017) 85–96. doi:10.1016/j.strusafe.2017.05.009.

[136] S. Mangalathu, J.-S. Jeon, Classification of failure mode and prediction of shear strength for reinforced concrete beam-column joints using machine learning techniques, Engineering Structures 160 (2018) 85–94. doi:10.1016/j.engstruct.2018.01.008.

[137] D. Hobbs, The compressive strength of concrete: a statistical approach to failure, Magazine of Concrete Research 24 (80) (1972) 127–138. doi:10.1680/macr.1973.24.80.127.

[138] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232.

[139] A. Goldstein, A. Kapelner, J. Bleich, E. Pitkin, Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation, journal of Computational and Graphical Statistics 24 (1) (2015) 44–65. doi:10.1080/10618600.2014.907095.

[140] A. Gallet, S. Rigby, T. N. Tallman, X. Kong, I. Hajirasouliha, A. Liew, D. Liu, L. Chen, A. Hauptmann, D. Smyl, Structural engineering from an inverse problems perspective, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 478 (2257) (2022) 20210526. doi:10.1098/rspa.2021.0526.

[141] J. N. Reddy, Introduction to the finite element method, McGraw-Hill Education, 2019.

[142] K. S. Surana, J. N. Reddy, The finite element method for boundary value problems: mathematics and computations, CRC press, 2016.

[143] T. J. Hughes, The finite element method: linear static and dynamic finite element analysis, Courier Corporation, 2012.

[144] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, J. Zhu, The finite element method, Vol. 3, McGraw-hill London, 1977.

[145] K. S. Virdi, Finite difference method for nonlinear analysis of structures, Journal of Constructional Steel Research 62 (11) (2006) 1210–1218. doi:10.1016/j.jcsr.2006.06.015.

[146] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, Computers & Structures 11 (1-2) (1980) 83–95. doi:10.1016/0045-7949(80)90149-2.

[147] J. D. De Basabe, M. K. Sen, A comparison of finite-difference and spectral-element methods for elastic wave propagation in media with a fluid-solid interface, Geophysical Journal International 200 (1) (2015) 278–298. doi:10.1093/gji/ggu389.

[148] P. Kudela, A. Żak, M. Krawczuk, W. Ostachowicz, Modelling of wave propagation in composite plates using the time domain spectral element method, Journal of Sound and Vibration 302 (4-5) (2007) 728–745. doi:10.1016/j.jsv.2006.12.016.

[149] B. E. Griffith, X. Luo, Hybrid finite difference/finite element immersed boundary method, International journal for numerical methods in biomedical engineering 33 (12) (2017) e2888. doi:10.1002/cnm.2888.

[150] C. R. Farrar, K. Worden, Structural health monitoring: a machine learning perspective, John Wiley & Sons, 2012.

[151] G.-R. Liu, X. Han, Computational inverse techniques in nondestructive evaluation, CRC press, 2003.

[152] E. Turco, Tools for the numerical solution of inverse problems in structural mechanics: review and research perspectives, European Journal of Environmental and Civil Engineering 21 (5) (2017) 509–554. doi:10.1080/19648189.2015.1134673.

[153] J. Kaipio, E. Somersalo, Statistical and computational inverse problems, Vol. 160, Springer Science & Business Media, 2006.

[154] J. Adler, O. Öktem, Solving ill-posed inverse problems using iterative deep neural networks, Inverse Problems 33 (12) (2017) 124007. doi:10.1088/1361-6420/aa9581.

[155] A. Calderon, On an inverse boundary, in: Seminar on Numerical Analysis and its Applications to Continuum Physics, Rio de Janeiro, 1980, Brazilian Math. Soc., 1980, pp. 65–73.

[156] A. Tarantola, B. Valette, et al., Inverse problems= quest for information, Journal of geophysics 50 (1) (1982) 159–170.

[157] F. Natterer, The mathematics of computerized tomography, SIAM, 2001.

[158] J. L. Mueller, S. Siltanen, Linear and nonlinear inverse problems with practical applications, SIAM, 2012.

[159] H. W. Engl, M. Hanke, A. Neubauer, Regularization of inverse problems, Vol. 375, Springer Science & Business Media, 1996.

[160] A. I. Nachman, Reconstructions from boundary measurements, Annals of Mathematics 128 (3) (1988) 531–576. doi:10.2307/1971435.

[161] A. I. Nachman, Global uniqueness for a two-dimensional inverse boundary value problem, Annals of Mathematics (1996) 71–96doi:10.2307/2118653.

[162] A. Tikhonov, V. Arsenin, Solutions of Ill-posed Problems, John Wiley & Sons Inc, 1977.

[163] L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D: Nonlinear Phenomena 60 (1) (1992) 259–268. doi:10.1016/0167-2789(92)90242-F.

[164] E. Haber, U. M. Ascher, D. Oldenburg, On optimization techniques for solving nonlinear inverse problems, Inverse problems 16 (5) (2000) 1263. doi:10.1088/0266-5611/16/5/309.

[165] S. Arridge, P. Maass, O. Öktem, C.-B. Schönlieb, Solving inverse problems using data-driven models, Acta Numerica 28 (2019) 1–174. doi:10.1017/S0962492919000059.

[166] K. H. Jin, M. T. McCann, E. Froustey, M. Unser, Deep convolutional neural network for inverse problems in imaging, IEEE Transactions on Image Processing 26 (9) (2017) 4509–4522. doi:10.1109/TIP.2017.2713099.

[167] I. Kang, M. J. Schulz, J. H. Kim, V. Shanov, D. Shi, A carbon nanotube strain sensor for structural health monitoring, Smart Materials and Structures 15 (3) (2006) 737. doi:10.1088/0964-1726/15/3/009.

[168] S. J. Hamilton, A. Hauptmann, Deep D-Bar: Real-Time Electrical Impedance Tomography Imaging With Deep Neural Networks, IEEE Transactions on Medical Imaging 37 (10) (2018) 2367–2377. doi:10.1109/TMI.2018.2828303.

[169] J. Adler, O. Öktem, Learned primal-dual reconstruction, IEEE transactions on medical imaging 37 (6) (2018) 1322–1332. doi:10.1109/TMI.2018.2799231.

[170] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, F. Knoll, Learning a variational network for reconstruction of accelerated mri data, Magnetic resonance in medicine 79 (6) (2018) 3055–3071. doi:10.1002/mrm.26977.

[171] A. Hauptmann, F. Lucka, M. Betcke, N. Huynh, J. Adler, B. Cox, P. Beard, S. Ourselin, S. Arridge, Model-based learning for accelerated, limited-view 3-d photoacoustic tomography, IEEE transactions on medical imaging 37 (6) (2018) 1382–1393. doi:10.1109/TMI.2018.2820382.

[172] D. Smyl, T. N. Tallman, D. Liu, A. Hauptmann, An efficient quasi-newton method for nonlinear inverse problems via learned singular values, IEEE Signal Processing Letters 28 (2021) 748–752. doi:10.1109/LSP.2021.3063622.

[173] J. Zhang, B. Ghanem, Ista-net: Interpretable optimization-inspired deep network for image compressive sensing, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1828–1837.

[174] V. Monga, Y. Li, Y. C. Eldar, Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing, IEEE Signal Processing Magazine 38 (2) (2021) 18–44. doi:10.1109/MSP.2020.3016905.

[175] E. Crawley, J. Malmqvist, S. Ostlund, D. Brodeur, K. Edstrom, Rethinking engineering education, The CDIO Approach 302 (2007) 60–62. doi:10.1007/978-3-319-05561-9.

[176] B. E. Seely, The other re-engineering of engineering education, 1900-1965, Journal of Engineering Education 88 (3) (1999) 285–294. doi:10.1002/j.2168-9830.1999.tb00449.x.

[177] E. Coleman, T. Shealy, J. Grohs, A. Godwin, Design thinking among first-year and senior engineering students: A cross-sectional, national study measuring perceived ability, Journal of Engineering Education 109 (1) (2020) 72–87. doi:10.1002/jee.20298.

[178] H. A. Simon, The structure of ill structured problems, Artificial Intelligence 4 (3-4) (1973) 181–201. doi:10.1016/0004-3702(73)90011-8.

[179] C. L. Dym, A. M. Agogino, O. Eris, D. D. Frey, L. J. Leifer, Engineering design thinking, teaching, and learning, IEEE Engineering Management Review 34 (1) (2006) 65–90. doi:10.1109/emr.2006.1679078.

[180] H. W. Rittel, M. M. Webber, Dilemmas in a general theory of planning, Policy Sciences 4 (2) (1973) 155–169. doi:10.1007/BF01405730.

[181] O. M. Alifanov, Methods of solving ill-posed inverse problems, Journal of Engineering Physics 45 (5) (1983) 1237–1245. doi:10.1007/BF01254725.

[182] J.-N. Yang, Application of optimal control theory to civil engineering structures, Journal of the engineering Mechanics Division 101 (6) (1975) 819–838. doi:10.1061/JMCEA3.0002075.

[183] J. Rombouts, A. Liew, G. Lombaert, L. De Laet, P. Block, M. Schevenels, Designing bending-active gridshells as falsework for concrete shells through numerical optimization, Engineering Structures 240 (2021) 112352. doi:10.1016/j.engstruct.2021.112352.

[184] A. Liew, Y. Stürz, S. Guillaume, T. Van Mele, R. Smith, P. Block, Active control of a rod-net formwork system prototype, Automation in Construction 96 (2018) 128–140. doi:10.1016/j.autcon.2018.09.002.

[185] T. Van Mele, D. Panozzo, O. Sorkine-Hornung, P. Block, Best-fit thrust network analysis - rationalization of freeform meshes, in: S. Adriaenssens, P. Block, D. Veenendaal, C. Williams (Eds.), Shell Structures for Architecture: Form Finding and Optimization, Routledge, London, 2014, pp. 157–170. doi:10.4324/9781315849270.

[186] H. E. Fairclough, M. Gilbert, A. V. Pichugin, A. Tyas, I. Firth, Theoretically optimal forms for very long-span bridges under gravity loading, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474 (2217) (2018) 20170726. doi:10.1098/rspa.2017.0726.

[187] A. Liew, Constrained force density method optimisation for compression-only shell structures, Structures 28 (2020) 1845–1856. doi:https://doi.org/10.1016/j.istruc.2020.09.078.

[188] L. He., M. Gilbert, Rationalization of trusses generated via layout optimization, Structural and Multidisciplinary Optimization 52 (4) (2015) 677–694. doi:10.1007/s00158-015-1260-x.

[189] R. McKinstray, J. B. Lim, T. T. Tanyimboh, D. T. Phan, W. Sha, Optimal design of long-span steel portal frames using fabricated beams, Journal of Constructional Steel Research 104 (2015) 104–114. doi:10.1016/j.jcsr.2014.10.010.

[190] J. Ye, J. Becque, I. Hajirasouliha, S. M. Mojtabaei, J. B. Lim, Development of optimum cold-formed steel sections for maximum energy dissipation in uniaxial bending, Engineering Structures 161 (2018) 55–67. doi:https://doi.org/10.1016/j.engstruct.2018.01.070.

[191] B. V. Koen, Discussion of The Method: Conducting the engineer's approach to problem solving, Oxford University Press, New York, 2003.

[192] D. Cormie, G. Mays, P. Smith, Blast effects on buildings, 2nd ed., Thomas Telford, London, UK, 2009.

[193] S. E. Rigby, T. J. Lodge, S. Alotaibi, A. D. Barr, S. D. Clarke, G. S. Langdon, A. Tyas, Preliminary yield estimation of the 2020 Beirut explosion using video footage from social media, Shock Waves 30 (6) (2020) 671–675. doi:10.1007/s00193-020-00970-z.

[194] D. Ambrosini, B. Luccioni, A. Jacinto, R. Danesi, Location and mass of explosive from structural damage, Engineering Structures 27 (2) (2005) 167–176. doi:10.1016/j.engstruct.2004.09.003.

[195] M. van der Voort, R. van Wees, S. Brouwer, M. van der Jagt-Deutekom, J. Verreault, Forensic analysis of explosions: Inverse calculation of the charge mass, Forensic Science International 252 (2015) 11–21. doi:10.1016/j.forsciint.2015.04.014.

[196] S. D. Clarke, S. D. Fay, J. A. Warren, A. Tyas, S. E. Rigby, I. Elgy, A large scale experimental approach to the measurement of spatially and temporally localised loading from the detonation of shallow-buried explosives, Measurement Science and Technology 26 (2015) 015001. doi:10.1088/0957-0233/26/1/015001.

[197] S. Rigby, A. Tyas, S. Clarke, S. Fay, J. Reay, J. Warren, M. Gant, I. Elgy, Observations from Preliminary Experiments on Spatial and Temporal Pressure Measurements from Near-Field Free Air Explosions, International Journal of Protective Structures 6 (2) (2015) 175–190. doi:10.1260/2041-4196.6.2.175.

[198] R. J. Curry, G. S. Langdon, Transient response of steel plates subjected to close proximity explosive detonations in air, International Journal of Impact Engineering 102 (2017) 102–116. doi:10.1016/j.ijimpeng.2016.12.004.

[199] S. E. Rigby, A. Tyas, R. J. Curry, G. S. Langdon, Experimental measurement of specific impulse distribution and transient deformation of plates subjected to near-field explosive blasts, Experimental Mechanics 59 (2) (2019) 163–178. doi:10.1007/s11340-018-00438-3.

[200] S. E. Rigby, O. I. Akintaro, B. J. Fuller, A. Tyas, R. J. Curry, G. S. Langdon, D. J. Pope, Predicting the response of plates subjected to near-field explosions using an energy equivalent impulse, International Journal of Impact Engineering 128 (2019) 24–36. doi:10.1016/j.ijimpeng.2019.01.014.

[201] J. Pannell, G. Panoutsos, S. Cooke, D. Pope, S. Rigby, Predicting specific impulse distributions for spherical explosives in the extreme near-field using a gaussian function, International Journal of Protective Structures 0 (0) (2021) 2041419621993492, (in press). doi:10.1177/2041419621993492.

[202] S. Xu, X. Deng, V. Tiwari, M. A. Sutton, W. L. Fourney, D. Bretall, An inverse approach for pressure load identification, International Journal of Impact Engineering 37 (7) (2010) 865–877. doi:10.1016/j.ijimpeng.2009.10.007.

[203] S. Xu, V. Tiwari, X. Deng, M. A. Sutton, W. L. Fourney, Identification of interaction pressure between structure and explosive with inverse approach, Experimental Mechanics 51 (6) (2011) 815–830. doi:10.1007/s11340-010-9390-y.

[204] C. Langran-Wheeler, S. Rigby, S. Clarke, A. Tyas, C. Stephens, R. Walker, Near-field spatial and temporal blast pressure distributions from non-spherical charges: Horizontally-aligned cylinders, International Journal of Protective Structures 0 (0) (2021) 20414196211013443, (in press). doi:10.1177/20414196211013443.

[205] A. Tyas, J. J. Reay, S. D. Fay, S. D. Clarke, S. E. Rigby, J. A. Warren, D. J. Pope, Experimental studies of the effect of rapid afterburn on shock development of near-field explosions, International Journal of Protective Structures 7 (3) (2016) 456–465. doi:10.1177/2041419616665931.

[206] S. E. Rigby, R. Knighton, S. D. Clarke, A. Tyas, Reflected near-field blast pressure measurements using high speed video, Experimental Mechanics 60 (7) (2020) 875–888. doi:10.1007/s11340-020-00615-3.

[207] S. Rigby, S. Fay, A. Tyas, S. Clarke, J. Reay, J. Warren, M. Gant, I. Elgy, Influence of particle size distribution on the blast pressure profile from explosives buried in saturated soils, Shock Waves 28 (3) (2018) 613–626. doi:10.1007/s00193-017-0727-7.

[208] M. I. Friswell, Damage identification using inverse methods, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 365 (1851) (2007) 393–410. doi:10.1098/rsta.2006.1930.

[209] G. Kerschen, K. Worden, A. F. Vakakis, J.-C. Golinval, Past, present and future of nonlinear system identification in structural dynamics, Mechanical systems and signal processing 20 (3) (2006) 505–592. doi:10.1016/j.ymssp.2005.04.008.

[210] J. K. Sinha, M. Friswell, S. Edwards, Simplified models for the location of cracks in beam structures using measured vibration data, Journal of Sound and vibration 251 (1) (2002) 13–38. doi:10.1006/jsvi.2001.3978.

[211] J.-J. Sinou, A review of damage detection and health monitoring of mechanical systems from changes in the measurement of linear and non-linear vibrations, Mechanical vibrations: measurement, effects and control (2009) 643–702.

[212] C. Farhat, F. M. Hemez, Updating finite element dynamic models using an element-by-element sensitivity methodology, AIAA journal 31 (9) (1993) 1702–1711. doi:10.2514/3.11833.

[213] J. Ricles, J. Kosmatka, Damage detection in elastic structures using vibratory residual forces and weighted sensitivity, AIAA journal 30 (9) (1992) 2310–2316. doi:10.2514/3.11219.

[214] H.-P. Lin, Direct and inverse methods on free vibration analysis of simply supported beams with a crack, Engineering structures 26 (4) (2004) 427–436. doi:10.1016/j.engstruct.2003.10.014.

[215] D. Hester, A. González, A wavelet-based damage detection algorithm based on bridge acceleration response to a vehicle, Mechanical Systems and Signal Processing 28 (2012) 145–166. doi:10.1016/j.ymssp.2011.06.007.

[216] L. Hadjileontiadis, E. Douka, A. Trochidis, Fractal dimension analysis for crack identification in beam structures, Mechanical Systems and Signal Processing 19 (3) (2005) 659–674. doi:10.1016/j.ymssp.2004.03.005.

[217] P. M. Pawar, R. Ganguli, Genetic fuzzy system for damage detection in beams and helicopter rotor blades, Computer methods in applied mechanics and engineering 192 (16-18) (2003) 2031–2057. doi:10.1016/S0045-7825(03)00237-8.

[218] E. Lourens, E. Reynders, G. De Roeck, G. Degrande, G. Lombaert, An augmented kalman filter for force identification in structural dynamics, Mechanical Systems and Signal Processing 27 (2012) 446–460. doi:10.1016/j.ymssp.2011.09.025.

[219] J. Nichols, S. Trickey, M. Todd, L. Virgin, Structural health monitoring through chaotic interrogation, Meccanica 38 (2) (2003) 239–250. doi:10.1023/A:1022898403359.

[220] J. Liu, X. Sun, X. Han, C. Jiang, D. Yu, A novel computational inverse technique for load identification using the shape function method of moving least square fitting, Computers & Structures 144 (2014) 127–137. doi:10.1016/j.compstruc.2014.08.002.

[221] H. Nasrellah, C. Manohar, A particle filtering approach for structural system identification in vehicle–structure interaction problems, Journal of Sound and Vibration 329 (9) (2010) 1289–1309. doi:10.1016/j.jsv.2009.10.041.

[222] H. Ebrahimian, R. Astroza, J. P. Conte, C. Papadimitriou, Bayesian optimal estimation for output-only nonlinear system and damage identification of civil structures, Structural Control and Health Monitoring 25 (4) (2018) e2128. doi:10.1002/stc.2128.

[223] C. Farrar, K. Worden, An introduction to structural health monitoring, Phil. Trans. R. Soc. A 365 (2007) 303–315. doi:10.1098/rsta.2006.1928.

[224] K. Worden, C. R. Farrar, J. Haywood, M. Todd, A review of nonlinear dynamics applications to structural health monitoring, 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 15 (4) (2008) 540–567. doi:10.1002/stc.215.

[225] L. Huang, L. Zeng, J. Lin, Baseline-free damage detection in composite plates based on the reciprocity principle, Smart materials and Structures 27 (1) (2017) 015026. doi:10.1088/1361-665X/aa9cc1.

[226] H. Sohn, H. W. Park, K. H. Law, C. R. Farrar, Combination of a time reversal process and a consecutiv outlier analysis for baseline-free damage diagnosis, Journal of Intelligent Material Systems and Structures 18 (4) (2007) 335–346. doi:10.1177/1045389X0606629.

[227] J. P. Santos, A. D. Orcesi, C. Crémona, P. Silveira, Baseline-free real-time assessment of structural changes, Structure and Infrastructure Engineering 11 (2) (2015) 145–161. doi:10.1080/15732479.2013.858169.

[228] S. R. Anton, D. J. Inman, G. Park, Reference-free damage detection using instantaneous baseline measurements, AIAA journal 47 (8) (2009) 1952–1964. doi:10.2514/1.43252.

[229] D. Inaudi, Long-term static structural health monitoring, in: Structures Congress 2010, 2010, pp. 566–577. doi:10.1061/41130(369)52.

[230] A. Downey, A. D'Alessandro, M. Baquera, E. García-Macías, D. Rolfes, F. Ubertini, S. Laflamme, R. Castro-Triguero, Damage detection, localization and quantification in conductive smart concrete structures using a resistor mesh model, Engineering Structures 148 (2017) 924–935. doi:10.1016/j.engstruct.2017.07.022.

[231] Z. Zhou, W. Liu, Y. Huang, H. Wang, H. Jianping, M. Huang, O. Jinping, Optical fiber bragg grating sensor assembly for 3d strain monitoring and its case study in highway pavement, Mechanical Systems and Signal Processing 28 (2012) 36–49. doi:10.1016/j.ymssp.2011.10.003.

[232] D. Lecompte, A. Smits, S. Bossuyt, H. Sol, J. Vantomme, D. Van Hemelrijck, A. Habraken, Quality assessment of speckle patterns for digital image correlation, Optics and lasers in Engineering 44 (11) (2006) 1132–1145. doi:10.1016/j.optlaseng.2005.10.004.

[233] L. Borcea, Electrical impedance tomography, Inverse problems 18 (6) (2002) R99. doi:10.1088/0266-5611/18/6/201.

[234] A. Zare Hosseinzadeh, G. Ghodrati Amiri, K.-Y. Koo, Optimization-based method for structural damage localization and quantification by means of static displacements computed by flexibility matrix, Engineering Optimization 48 (4) (2016) 543–561. doi:10.1080/0305215X.2015.1017476.

[235] A. Kefal, E. Oterkus, Displacement and stress monitoring of a panamax containership using inverse finite element method, Ocean Engineering 119 (2016) 16–29. doi:10.1016/j.oceaneng.2016.04.025.

[236] M. Sanayei, A. Khaloo, M. Gul, F. N. Catbas, Automated finite element model updating of a scale bridge model using measured static and modal test data, Engineering Structures 102 (2015) 66–79. doi:10.1016/j.engstruct.2015.07.029.

[237] B. Ni, H. Gao, A deep learning approach to the inverse problem of modulus identification in elasticity, MRS Bulletin 46 (1) (2021) 19–25. doi:10.1557/s43577-020-00006-y.

[238] J. Waeytens, B. Rosić, P.-E. Charbonnel, E. Merliot, D. Siegert, X. Chapeleau, R. Vidal, V. Le Corvec, L.-M. Cottineau, Model updating techniques for damage detection in concrete beam using optical fiber strain measurement device, Engineering Structures 129 (2016) 2–10. doi:10.1016/j.engstruct.2016.08.004.

[239] A. P. Ruybalid, J. P. Hoefnagels, O. van der Sluis, M. G. Geers, Comparison of the identification performance of conventional fem updating and integrated dic, International Journal for Numerical Methods in Engineering 106 (4) (2016) 298–320. doi:10.1002/nme.5127.

[240] A. Karageorghis, D. Lesnic, L. Marin, The method of fundamental solutions for three-dimensional inverse geometric elasticity problems, Computers & Structures 166 (2016) 51–59. doi:10.1016/j.compstruc.2016.01.010.

[241] F. Mathieu, H. Leclerc, F. Hild, S. Roux, Estimation of elastoplastic parameters via weighted femu and integrated-dic, Experimental Mechanics 55 (1) (2015) 105–119. doi:10.1007/s11340-014-9888-9.

[242] A. Kefal, E. Oterkus, A. Tessler, J. L. Spangler, A quadrilateral inverse-shell element with drilling degrees of freedom for shape sensing and structural health monitoring, Engineering science and technology, an international journal 19 (3) (2016) 1299–1313. doi:10.1016/j.jestch.2016.03.006.

[243] D. Smyl, K.-N. Antin, D. Liu, S. Bossuyt, Coupled digital image correlation and quasi-static elasticity imaging of inhomogeneous orthotropic composite structures, Inverse Problems 34 (12) (2018) 124005. doi:10.1088/1361-6420/aae793.

[244] B. Shen, I. Stanciulescu, G. H. Paulino, Inverse computation of cohesive fracture properties from displacement fields, Inverse Problems in Science and Engineering 18 (8) (2010) 1103–1128. doi:10.1080/17415977.2010.512661.

[245] Z.-H. Xu, Y. Yang, P. Huang, X. Li, Determination of interfacial properties of thermal barrier coatings by shear test and inverse finite element method, Acta Materialia 58 (18) (2010) 5972–5979. doi:10.1016/j.actamat.2010.07.013.

[246] S. Avril, M. Bonnet, A.-S. Bretelle, M. Grédiac, F. Hild, P. Ienny, F. Latourte, D. Lemosse, S. Pagano, E. Pagnacco, et al., Overview of identification methods of mechanical parameters based on full-field measurements, Experimental Mechanics 48 (4) (2008) 381–402. doi:10.1007/s11340-008-9148-y.

[247] B. Lecampion, J. Gunning, Model selection in fracture mapping from elastostatic data, International journal of solids and structures 44 (5) (2007) 1391–1408. doi:10.1016/j.ijsolstr.2006.06.022.

[248] F. Amiot, F. Hild, J. P. Roger, Identification of elastic property and loading fields from full-field displacement measurements, International Journal of Solids and Structures 44 (9) (2007) 2863–2887. doi:10.1016/j.ijsolstr.2006.08.031.

[249] F. Hild, S. Roux, Digital image correlation: from displacement measurement to identification of elastic properties–a review, Strain 42 (2) (2006) 69–80. doi:10.1111/j.1475-1305.2006.00258.x.

[250] M. Bonnet, A. Constantinescu, Inverse problems in elasticity, Inverse Problems 21 (2) (2005) R1. doi:10.1088/0266-5611/21/2/R01.

[251] A. Morassi, E. Rosset, Stable determination of cavities in elastic bodies, Inverse Problems 20 (2) (2004) 453. doi:10.1088/0266-5611/20/2/010.

[252] A. Maniatty, N. Zabaras, K. Stelson, Finite element analysis of some inverse elasticity problems, Journal of engineering mechanics 115 (6) (1989) 1303–1317. doi:10.1061/(ASCE)0733-9399(1989)115:6(1303).

[253] G. F. Gomes, J. V. P. Pereira, Sensor placement optimization and damage identification in a fuselage structure using inverse modal problem and firefly algorithm, Evolutionary Intelligence 13 (4) (2020) 571–591. doi:10.1007/s12065-020-00372-1.

[254] Y. An, E. Chatzi, S.-H. Sim, S. Laflamme, B. Blachowski, J. Ou, Recent progress and future trends on damage identification methods for bridge structures, Structural Control and Health Monitoring 26 (10) (2019) e2416. doi:10.1002/stc.2416.

[255] L. Colombo, C. Sbarufatti, M. Giglio, Definition of a load adaptive baseline by inverse finite element method for structural damage identification, Mechanical Systems and Signal Processing 120 (2019) 584–607. doi:10.1016/j.ymssp.2018.10.041.

[256] M. Amanzadeh, S. M. Aminossadati, M. S. Kizil, A. D. Rakić, Recent developments in fibre optic shape sensing, Measurement 128 (2018) 119–137. doi:10.1016/j.measurement.2018.06.034.

[257] R. Di Sante, Fibre optic sensors for structural health monitoring of aircraft composite structures: Recent advances and applications, Sensors 15 (8) (2015) 18666–18713. doi:10.3390/s150818666.

[258] P. Cerracchio, M. Gherlone, M. Di Sciuva, A. Tessler, A novel approach for displacement and stress monitoring of sandwich structures based on the inverse finite element method, Composite Structures 127 (2015) 69–76. doi:10.1016/j.compstruct.2015.02.081.

[259] M. Gherlone, P. Cerracchio, M. Mattone, M. Di Sciuva, A. Tessler, An inverse finite element method for beam shape sensing: theoretical framework and experimental validation, Smart Materials and Structures 23 (4) (2014) 045027. doi:10.1088/0964-1726/23/4/045027.

[260] A. Derkevorkian, S. F. Masri, J. Alvarenga, H. Boussalis, J. Bakalyar, W. L. Richards, Strain-based deformation shape-estimation algorithm for control and monitoring applications, AIAA journal 51 (9) (2013) 2231–2240. doi:10.2514/1.J052215.

[261] D. Weisz-Patrault, A. Ehrlacher, N. Legrand, Evaluation of contact stress during rolling process, by three dimensional analytical inverse method, International Journal of Solids and Structures 50 (20-21) (2013) 3319–3331. doi:10.1016/j.ijsolstr.2013.06.005.

[262] M. Gherlone, P. Cerracchio, M. Mattone, M. Di Sciuva, A. Tessler, Shape sensing of 3d frame structures using an inverse finite element method, International Journal of Solids and Structures 49 (22) (2012) 3100–3112. doi:10.1016/j.ijsolstr.2012.06.009.

[263] A. Tessler, J. L. Spangler, M. Gherlone, M. Mattone, M. Di Sciuva, Real-time characterization of aerospace structures using onboard strain measurement technologies and inverse finite element method, Tech. rep., National Aeronautics and Space Administration VA Langley Research Center (2011).

[264] S. Kiesel, P. Van Vickle, K. J. Peters, T. Hassan, M. Kowalsky, Intrinsic polymer optical fiber sensors for high-strain applications, in: Smart Structures and Materials 2006: Smart Sensor Monitoring Systems and Applications, Vol. 6167, International Society for Optics and Photonics, 2006, p. 616713. doi:10.1117/12.657436.

[265] J.-H. Song, E.-T. Lee, H.-C. Eun, Optimal sensor placement through expansion of static strain measurements to static displacements, International Journal of Distributed Sensor Networks 17 (1) (2021) 1550147721991712. doi:10.1177/1550147721991712.

[266] M. Esposito, M. Gherlone, Composite wing box deformed-shape reconstruction based on measured strains: Optimization and comparison of existing approaches, Aerospace Science and Technology 99 (2020) 105758. doi:10.1016/j.ast.2020.105758.

[267] W. Ostachowicz, R. Soman, P. Malinowski, Optimization of sensor placement for structural health monitoring: A review, Structural Health Monitoring 18 (3) (2019) 963–988. doi:10.1177/1475921719825601.

[268] A. F. G. Tenreiro, A. M. Lopes, L. F. da Silva, A review of structural health monitoring of bonded structures using electromechanical impedance spectroscopy, Structural Health Monitoring (2021) 1475921721993419doi:10.1177/147592172199341.

[269] G. Alessandrini, A. Morassi, E. Rosset, Detecting cavities by electrostatic boundary measurements, Inverse Problems 18 (5) (2002) 1333. doi:10.1088/0266-5611/18/5/308.

[270] A. Downey, A. D'Alessandro, S. Laflamme, F. Ubertini, Smart bricks for strain sensing and crack detection in masonry structures, Smart Materials and Structures 27 (1) (2017) 015009. doi:10.1088/1361-665X/aa98c2.

[271] M. Sadoughi, A. Downey, J. Yan, C. Hu, S. Laflamme, Reconstruction of unidirectional strain maps via iterative signal fusion for mesoscale structures monitored by a sensing skin, Mechanical Systems and Signal Processing 112 (2018) 401–416. doi:10.1016/j.ymssp.2018.04.023.

[272] X. Kong, J. Li, W. Collins, C. Bennett, S. Laflamme, H. Jo, Sensing distortion-induced fatigue cracks in steel bridges with capacitive skin sensor arrays, Smart Materials and Structures 27 (11) (2018) 115008. doi:10.1088/1361-665X/aadbfb.

[273] X. Kong, J. Li, W. Collins, C. Bennett, S. Laflamme, H. Jo, A large-area strain sensing technology for monitoring fatigue cracks in steel bridges, Smart Materials and Structures 26 (8) (2017) 085024. doi:10.1088/1361-665X/aa75ef.

[274] A. Downey, S. Laflamme, F. Ubertini, Reconstruction of in-plane strain maps using hybrid dense sensor network composed of sensing skin, Measurement Science and Technology 27 (12) (2016) 124016. doi:10.1088/0957-0233/27/12/124016.

[275] A. Voss, P. Hosseini, M. Pour-Ghaz, M. Vauhkonen, A. Seppänen, Three-dimensional electrical capacitance tomography – a tool for characterizing moisture transport properties of cement-based materials, Materials & Design 181 (2019) 107967. doi:https://doi.org/10.1016/j.matdes.2019.107967.

[276] K. Grudzien, Z. Chaniecki, A. Romanowski, D. Sankowski, J. Nowakowski, M. Niedostatkiewicz, Application of twin-plane ect sensor for identification of the internal imperfections inside concrete beams, in: 2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings, 2016, pp. 1–6. doi:10.1109/I2MTC.2016.7520512.

[277] D. Smyl, M. Pour-Ghaz, A. Seppänen, Detection and reconstruction of complex structural cracking patterns with electrical imaging, NDT & E International 99 (2018) 123–133. doi:10.1016/j.ndteint.2018.06.004.

[278] A. Seppänen, M. Hallaji, M. Pour-Ghaz, A functionally layered sensing skin for the detection of corrosive elements and cracking, Structural Health Monitoring 16 (2) (2017) 215–224. doi:10.1177/1475921716670574.

[279] K. Karhunen, A. Seppänen, A. Lehikoinen, P. Monteiro, J. Kaipio, Electrical resistance tomography imaging of concrete, Cement Concrete Research 40 (2010) 137–145. doi:10.1016/j.cemconres.2009.08.023.

[280] R. Rashetnia, M. Hallaji, D. Smyl, A. Seppänen, M. Pour-Ghaz, Detection and localization of changes in two-dimensional temperature distributions by electrical resistance tomography, Smart Materials and Structures 26 (11) (2017) 115021. doi:10.1088/1361-665x/aa8f75.

[281] A. Hauptmann, D. Smyl, Fusing electrical and elasticity imaging, Philosophical Transactions of the Royal Society A 379 (2200) (2021) 20200194. doi:10.1098/rsta.2020.0194.

[282] Y. Bao, Z. Chen, S. Wei, Y. Xu, Z. Tang, H. Li, The state of the art of data science and engineering in structural health monitoring, Engineering 5 (2) (2019) 234–242. doi:10.1016/j.eng.2018.11.027.

[283] X. Fan, J. Li, H. Hao, S. Ma, Identification of minor structural damage based on electromechanical impedance sensitivity and sparse regularization, Journal of Aerospace Engineering 31 (5) (2018) 04018061. doi:10.1061/(ASCE)AS.1943-5525.0000892.

[284] J. Kim, K.-W. Wang, Electromechanical impedance-based damage identification enhancement using bistable and adaptive piezoelectric circuitry, Structural Health Monitoring 18 (4) (2019) 1268–1281. doi:10.1177/1475921718794202.

[285] J. M. Brownjohn, Structural health monitoring of civil infrastructure, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 365 (1851) (2007) 589–622. doi:10.1098/rsta.2006.1925.

[286] A. Entezami, H. Shariatmadar, Structural health monitoring by a new hybrid feature extraction and dynamic time warping methods under ambient vibration and non-stationary signals, Measurement 134 (2019) 548–568. doi:10.1016/j.measurement.2018.10.095.

[287] L. F. Ramos, R. Aguilar, P. B. Lourenço, S. Moreira, Dynamic structural health monitoring of saint torcato church, Mechanical Systems and Signal Processing 35 (1-2) (2013) 1–15. doi:10.1016/j.ymssp.2012.09.007.

[288] F. Magalhães, Á. Cunha, E. Caetano, Vibration based structural health monitoring of an arch bridge: from automated oma to damage detection, Mechanical Systems and Signal Processing 28 (2012) 212–228. doi:10.1016/j.ymssp.2011.06.011.

[289] S. Nagarajaiah, Y. Yang, Modeling and harnessing sparse and low-rank data structure: a new paradigm for structural dynamics, identification, damage detection, and health monitoring, Structural Control and Health Monitoring 24 (1) (2017) e1851. doi:10.1002/stc.1851.

[290] K.-Y. Koo, J. Brownjohn, D. List, R. Cole, Structural health monitoring of the tamar suspension bridge, Structural Control and Health Monitoring 20 (4) (2013) 609–625. doi:10.1002/stc.1481.

[291] A. Marcuzzi, A. Morassi, Dynamic identification of a concrete bridge with orthotropic plate-type deck, Journal of structural engineering 136 (5) (2010) 586–602. doi:10.1061/(ASCE)ST.1943-541X.0000146.

[292] D. Pines, L. Salvino, Structural health monitoring using empirical mode decomposition and the hilbert phase, Journal of sound and vibration 294 (1-2) (2006) 97–124. doi:10.1016/j.jsv.2005.10.024.

[293] Y. H. Hong, H.-K. Kim, H. S. Lee, Reconstruction of dynamic displacement and velocity from measured accelerations using the variational statement of an inverse problem, Journal of Sound and Vibration 329 (23) (2010) 4980–5003. doi:10.1016/j.jsv.2010.05.016.

[294] Z. Wan, S. Li, Q. Huang, T. Wang, Structural response reconstruction based on the modal superposition method in the presence of closely spaced modes, Mechanical Systems and Signal Processing 42 (1-2) (2014) 14–30. doi:10.1016/j.ymssp.2013.07.007.

[295] B. Paul, R. C. George, S. K. Mishra, Phase space interrogation of the empirical response modes for seismically excited structures, Mechanical Systems and Signal Processing 91 (2017) 250–265. doi:10.1016/j.ymssp.2016.12.008.

[296] J. Nichols, M. Todd, J. Wait, Using state space predictive modeling with chaotic interrogation in detecting joint preload loss in a frame structure experiment, Smart Materials and Structures 12 (4) (2003) 580. doi:10.1088/0964-1726/12/4/310.

[297] J. Fernández-Sáez, A. Morassi, L. Rubio, The λ-curves method for crack identification in beams, Procedia engineering 199 (2017) 1964–1969. doi:10.1016/j.proeng.2017.09.304.

[298] G. Liu, Z. Mao, M. Todd, Z. Huang, Damage assessment with state–space embedding strategy and singular value decomposition under stochastic excitation, Structural Health Monitoring 13 (2) (2014) 131–142. doi:10.1177/1475921713513973.

[299] R. Yao, S. N. Pakzad, Autoregressive statistical pattern recognition algorithms for damage detection in civil structures, Mechanical Systems and Signal Processing 31 (2012) 355–368. doi:10.1016/j.ymssp.2012.02.014.

[300] K. K. Nair, A. S. Kiremidjian, K. H. Law, Time series-based damage detection and localization algorithm with application to the asce benchmark structure, Journal of Sound and Vibration 291 (1-2) (2006) 349–368. doi:10.1016/j.jsv.2005.06.016.

[301] N. Dervilis, H. Shi, K. Worden, E. Cross, Exploring environmental and operational variations in shm data using heteroscedastic gaussian processes, in: Dynamics of Civil Structures, Volume 2, Springer, 2016, pp. 145–153. doi:10.1007/978-3-319-29751-4_15.

[302] M. K. Ramancha, R. Astroza, J. P. Conte, J. I. Restrepo, M. D. Todd, Bayesian nonlinear finite element model updating of a full-scale bridge-column using sequential monte carlo, in: Model Validation and Uncertainty Quantification, Volume 3, Springer, 2020, pp. 389–397. doi:10.1007/978-3-030-47638-0_43.

[303] H.-P. Wan, Y.-Q. Ni, Bayesian multi-task learning methodology for reconstruction of structural health monitoring data, Structural Health Monitoring 18 (4) (2019) 1282–1309. doi:10.1177/1475921718794953.

[304] H.-P. Wan, W.-X. Ren, Stochastic model updating utilizing bayesian approach and gaussian process model, Mechanical Systems and Signal Processing 70 (2016) 245–268. doi:10.1016/j.ymssp.2015.08.011.

[305] Z. Mao, M. Todd, Statistical modeling of frequency response function estimation for uncertainty quantification, Mechanical Systems and Signal Processing 38 (2) (2013) 333–345. doi:10.1016/j.ymssp.2013.01.021.

[306] S. Cantero-Chinchilla, J. L. Beck, M. Chiachío, J. Chiachío, D. Chronopoulos, A. Jones, Optimal sensor and actuator placement for structural health monitoring via an efficient convex cost-benefit optimization, Mechanical Systems and Signal Processing 144 (2020) 106901. doi:10.1016/j.ymssp.2020.106901.

[307] G. Capellari, E. Chatzi, S. Mariani, Cost–benefit optimization of structural health monitoring sensor networks, Sensors 18 (7) (2018) 2174. doi:10.3390/s18072174.

[308] T.-H. Yi, H.-N. Li, C.-W. Wang, Multiaxial sensor placement optimization in structural health monitoring using distributed wolf algorithm, Structural Control and Health Monitoring 23 (4) (2016) 719–734. doi:10.1002/stc.1806.

[309] H. Sun, O. Büyüköztürk, Optimal sensor placement in structural health monitoring using discrete optimization, Smart Materials and Structures 24 (12) (2015) 125034. doi:10.1088/0964-1726/24/12/125034.

[310] M. Z. A. Bhuiyan, G. Wang, J. Cao, J. Wu, Sensor placement with multiple objectives for structural health monitoring, ACM Transactions on Sensor Networks (TOSN) 10 (4) (2014) 1–45. doi:10.1145/2533669.

[311] T.-H. Yi, H.-N. Li, X.-D. Zhang, A modified monkey algorithm for optimal sensor placement in structural health monitoring, Smart Materials and Structures 21 (10) (2012) 105033. doi:10.1088/0964-1726/21/10/105033.

[312] T.-H. Yi, H.-N. Li, M. Gu, Optimal sensor placement for structural health monitoring based on multiple optimization strategies, The Structural Design of Tall and Special Buildings 20 (7) (2011) 881–900. doi:10.1002/tal.712.

[313] H. Guo, L. Zhang, L. Zhang, J. Zhou, Optimal placement of sensors for structural health monitoring using improved genetic algorithms, Smart materials and structures 13 (3) (2004) 528. doi:10.1088/0964-1726/13/3/011.

[314] S. Cantero-Chinchilla, G. Aranguren, J. M. Royo, M. Chiachío, J. Etxaniz, A. Calvo-Echenique, Structural health monitoring using ultrasonic guided-waves and the degree of health index, Sensors 21 (3) (2021) 993. doi:10.3390/s21030993.

[315] M. Li, A. Kefal, B. Cerik, E. Oterkus, Structural health monitoring of submarine pressure hull using inverse finite element method, in: Trends in the Analysis and Design of Marine Structures: Proceedings of the 7th International Conference on Marine Structures (MARSTRUCT 2019, Dubrovnik, Croatia, 6-8 May 2019), CRC Press, 2019, p. 293.

[316] Y. Shen, Structural health monitoring using linear and nonlinear ultrasonic guided waves, Ph.D. thesis, University of South Carolina (2014).

[317] A. Srivastava, F. Lanza di Scalea, Quantitative structural health monitoring by ultrasonic guided waves, Journal of Engineering mechanics 136 (8) (2010) 937–944. doi:10.1061/(ASCE)EM.1943-7889.0000136.

[318] M. Mitra, S. Gopalakrishnan, Guided wave based structural health monitoring: A review, Smart Materials and Structures 25 (5) (2016) 053001. doi:10.1088/0964-1726/25/5/053001.

[319] J. Eiras, T. Kundu, J. S. Popovics, J. Monzó, L. Soriano, J. Payá, Evaluation of frost damage in cement-based materials by a nonlinear elastic wave technique, in: Health Monitoring of Structural and Biological Systems 2014, Vol. 9064, International Society for Optics and Photonics, 2014, p. 90641G. doi:10.1117/12.2035951.

[320] V. Giurgiutiu, Tuned lamb wave excitation and detection with piezoelectric wafer active sensors for structural health monitoring, Journal of intelligent material systems and structures 16 (4) (2005) 291–305. doi:10.1177/1045389X0505010.

[321] S. W. Shin, C.-B. Yun, J. S. Popovics, J. H. Kim, Improved rayleigh wave velocity measurement for nondestructive early-age concrete monitoring, Research in Nondestructive Evaluation 18 (1) (2007) 45–68. doi:10.1080/09349840601128762.

[322] B. Xu, V. Giurgiutiu, Single mode tuning effects on lamb wave time reversal with piezoelectric wafer active sensors for structural health monitoring, Journal of Nondestructive Evaluation 26 (2) (2007) 123–134. doi:10.1007/s10921-007-0027-8.

[323] C. H. Wang, J. T. Rose, F.-K. Chang, A synthetic time-reversal imaging method for structural health monitoring, Smart materials and structures 13 (2) (2004) 415. doi:10.1088/0964-1726/13/2/020.

[324] J. Moll, C.-P. Fritzen, Time-varying inverse filtering for high resolution imaging with ultrasonic guided waves, in: 10th European Conference on Non-Destructive Testing, 2010, pp. 1–10.

[325] J. He, D. C. Rocha, P. Sava, Guided wave tomography based on least-squares reverse-time migration, Structural Health Monitoring 19 (4) (2020) 1237–1249. doi:10.1177/1475921719880296.

[326] J. He, C. A. Leckey, P. E. Leser, W. P. Leser, Multi-mode reverse time migration damage imaging using ultrasonic guided waves, Ultrasonics 94 (2019) 319–331. doi:10.1016/j.ultras.2018.08.005.

[327] M. Zhao, W. Zhou, Y. Huang, H. Li, Sparse bayesian learning approach for propagation distance recognition and damage localization in plate-like structures using guided waves, Structural Health Monitoring (2020) 1475921720902277doi:10.1177/1475921720902277.

[328] X. Lu, M. Lu, L.-M. Zhou, Z. Su, L. Cheng, L. Ye, G. Meng, Evaluation of welding damage in welded tubular steel structures using guided waves and a probability-based imaging approach, Smart materials and structures 20 (1) (2010) 015018. doi:10.1088/0964-1726/20/1/015018.

[329] D. Smyl, T. N. Tallman, J. A. Black, A. Hauptmann, D. Liu, Learning and correcting non-gaussian model errors, Journal of Computational Physics 432 (2021) 110152. doi:10.1016/j.jcp.2021.110152.

[330] F.-G. Yuan, S. A. Zargar, Q. Chen, S. Wang, Machine learning for structural health monitoring: challenges and opportunities, in: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020, Vol. 11379, International Society for Optics and Photonics, 2020, p. 1137903. doi:10.1117/12.2561610.

[331] R. P. Finotti, A. A. Cury, F. d. S. Barbosa, An shm approach using machine learning and statistical indicators extracted from raw dynamic measurements, Latin American Journal of Solids and Structures 16 (2) (2019). doi:10.1590/1679-78254942.

[332] T. Khuc, F. N. Catbas, Completely contactless structural health monitoring of real-life structures using cameras and computer vision, Structural Control and Health Monitoring 24 (1) (2017) e1852. doi:10.1002/stc.1852.

[333] M. Q. Feng, Y. Fukuda, D. Feng, M. Mizuta, Nontarget vision sensor for remote measurement of bridge dynamic response, Journal of Bridge Engineering 20 (12) (2015) 04015023. doi:10.1061/(ASCE)BE.1943-5592.0000747.

[334] L. Luo, M. Q. Feng, Z. Y. Wu, Robust vision sensor for multi-point displacement monitoring of bridges in the field, Engineering Structures 163 (2018) 255–266. doi:10.1016/j.engstruct.2018.02.014.

[335] H. Yoon, H. Elanwar, H. Choi, M. Golparvar-Fard, B. F. Spencer Jr, Target-free approach for vision-based structural system identification using consumer-grade cameras, Structural Control and Health Monitoring 23 (12) (2016) 1405–1416. doi:10.1002/stc.1850.

[336] Y. Yang, C. Dorn, T. Mancini, Z. Talken, G. Kenyon, C. Farrar, D. Mascareñas, Blind identification of full-field vibration modes from video measurements with phase-based video motion magnification, Mechanical Systems and Signal Processing 85 (2017) 567–590. doi:10.1016/j.ymssp.2016.08.041.

[337] C.-Z. Dong, S. Bas, F. N. Catbas, A completely non-contact recognition system for bridge unit influence line using portable cameras and computer vision, Smart Struct. Syst (2019). doi:10.12989/sss.2019.24.5.617.

[338] D. Jana, S. Nagarajaiah, Computer vision-based real-time cable tension estimation in dubrovnik cable-stayed bridge using moving handheld video camera, Structural Control and Health Monitoring 28 (5) (2021) e2713. doi:10.1002/stc.2713.

[339] Z. Liu, Y. Cao, Y. Wang, W. Wang, Computer vision-based concrete crack detection using u-net fully convolutional networks, Automation in Construction 104 (2019) 129–139. doi:10.1016/j.autcon.2019.04.005.

[340] H.-K. Shen, P.-H. Chen, L.-M. Chang, Human-visual-perception-like intensity recognition for color rust images based on artificial neural network, Automation in Construction 90 (2018) 178–187. doi:10.1016/j.autcon.2018.02.023.

[341] L. Pauly, D. Hogg, R. Fuentes, H. Peel, Deeper networks for pavement crack detection, in: Proceedings of the 34th ISARC, IAARC, 2017, pp. 479–485.

[342] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Computer-Aided Civil and Infrastructure Engineering 32 (5) (2017) 361–378. doi:10.1111/mice.12263.

[343] C. V. Dung, et al., Autonomous concrete crack detection using deep fully convolutional neural network, Automation in Construction 99 (2019) 52–58. doi:10.1016/j.autcon.2018.11.028.

[344] F.-C. Chen, M. R. Jahanshahi, Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion, IEEE Transactions on Industrial Electronics 65 (5) (2017) 4392–4400. doi:10.1109/TIE.2017.2764844.

[345] Y.-J. Cha, K. You, W. Choi, Vision-based detection of loosened bolts using the hough transform and support vector machines, Automation in Construction 71 (2016) 181–188. doi:10.1016/j.autcon.2016.06.008.

[346] T.-C. Huynh, J.-H. Park, H.-J. Jung, J.-T. Kim, Quasi-autonomous bolt-loosening detection method using vision-based deep learning and image processing, Automation in Construction 105 (2019) 102844. doi:10.1016/j.autcon.2019.102844.

[347] X. Kong, J. Li, Non-contact fatigue crack detection in civil infrastructure through image overlapping and crack breathing sensing, Automation in Construction 99 (2019) 125–139. doi:10.1016/j.autcon.2018.12.011.

[348] X. Kong, J. Li, Image registration-based bolt loosening detection of steel joints, Sensors 18 (4) (2018) 1000. doi:10.3390/s18041000.

[349] D. Dominici, M. Alicandro, V. Massimi, Uav photogrammetry in the post-earthquake scenario: case studies in l'aquila, Geomatics, Natural Hazards and Risk 8 (1) (2017) 87–103. doi:10.1080/19475705.2016.1176605.

[350] J. Choi, C. M. Yeum, S. J. Dyke, M. R. Jahanshahi, Computer-aided approach for rapid post-event visual evaluation of a building facade, Sensors 18 (9) (2018) 3017. doi:10.3390/s18093017.

[351] G. Buffi, P. Manciola, S. Grassi, M. Barberini, A. Gambi, Survey of the ridracoli dam: Uav–based photogrammetry and traditional topographic techniques in the inspection of vertical structures, Geomatics, natural hazards and risk 8 (2) (2017) 1562–1579. doi:10.1201/9780429465086.

[352] A. Khaloo, D. Lattanzi, A. Jachimowicz, C. Devaney, Utilizing uav and 3d computer vision for visual inspection of a large gravity dam, Frontiers in Built Environment 4 (2018) 31. doi:10.3389/fbuil.2018.00031.

[353] T. Özaslan, S. Shen, Y. Mulgaonkar, N. Michael, V. Kumar, Inspection of penstocks and featureless tunnel-like environments using micro uavs, in: Field and Service Robotics, Springer, 2015, pp. 123–136. doi:10.1007/978-3-319-07488-7_9.

[354] M. Banić, A. Miltenović, M. Pavlović, I. Ćirić, Intelligent machine vision based railway infrastructure inspection and monitoring using uav, Facta Universitatis, Series: Mechanical Engineering 17 (3) (2019) 357–364. doi:10.22190/FUME190507041B.

[355] J. Shao, L. Tang, M. Liu, G. Shao, L. Sun, Q. Qiu, Bdd-net: A general protocol for mapping buildings damaged by a wide range of disasters based on satellite imagery, Remote Sensing 12 (10) (2020) 1670. doi:10.3390/rs12101670.

[356] R. Gupta, M. Shah, Rescuenet: Joint building segmentation and damage assessment from satellite imagery, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 4405–4411. doi:10.1109/ICPR48806.2021.9412295.

[357] Y. Furukawa, C. Hernández, Multi-view stereo: A tutorial, Foundations and Trends® in Computer Graphics and Vision 9 (1-2) (2015) 1–148. doi:0.1561/0600000052.

[358] A. Braun, A. Borrmann, Combining inverse photogrammetry and bim for automated labeling of construction site images for machine learning, Automation in Construction 106 (2019) 102879. doi:10.1016/j.autcon.2019.102879.

[359] M. Stavroulaki, B. Riveiro, G. A. Drosopoulos, M. Solla, P. Koutsianitis, G. E. Stavroulakis, Modelling and strength evaluation of masonry bridges using terrestrial photogrammetry and finite elements, Advances in Engineering Software 101 (2016) 136–148. doi:10.1016/j.advengsoft.2015.12.007.

[360] S. Zollini, M. Alicandro, D. Dominici, R. Quaresima, M. Giallonardo, Uav photogrammetry for concrete bridge inspection using object-based image analysis (obia), Remote Sensing 12 (19) (2020) 3180. doi:10.3390/rs12193180.

[361] R. A. Leishear, Bridge safety dangers-fatigue cracks, brittle failures and grit blasting, Journal of Civil, Construction and Environmental Engineering 6 (2) (2021) 28–45. doi:10.11648/j.jccee.20210602.12.

[362] D. Wagg, K. Worden, R. Barthorpe, P. Gardner, Digital twins: State-of-the-art and future directions for modeling and simulation in engineering dynamics applications, ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg 6 (3) (2020). doi:10.1115/1.4046739.

[363] B. R. Seshadri, T. Krishnamurthy, Structural health management of damaged aircraft structures using digital twin concept, in: 25th aiaa/ahs adaptive structures conference, 2017, p. 1675. doi:10.2514/6.2017-1675.

[364] G. Evensen, Analysis of iterative ensemble smoothers for solving inverse problems, Computational Geosciences 22 (3) (2018) 885–908. doi:10.1007/s10596-018-9731-y.

[365] M. A. Iglesias, K. J. Law, A. M. Stuart, Ensemble kalman methods for inverse problems, Inverse Problems 29 (4) (2013) 045001. doi:10.1088/0266-5611/29/4/045001.

[366] A. Seppänen, M. Vauhkonen, P. Vauhkonen, E. Somersalo, J. Kaipio, State estimation with fluid dynamical evolution models in process tomography-an application to impedance tomography, Inverse Problems 17 (3) (2001) 467. doi:10.1088/0266-5611/17/3/307.

[367] S. Lunz, A. Hauptmann, T. Tarvainen, C.-B. Schönlieb, S. Arridge, On learned operator correction in inverse problems, SIAM Journal on Imaging Sciences 14 (1) (2021) 92–127. doi:10.1137/20M1338460.

[368] J. M. Huttunen, J. P. Kaipio, Approximation errors in nonstationary inverse problems, Inverse Problems & Imaging 1 (1) (2007) 77. doi:10.3934/ipi.2007.1.77.

[369] M. Benning, M. Burger, Modern regularization methods for inverse problems, Acta Numerica 27 (2018) 1–111. doi:10.1017/S0962492918000016.

[370] C. Lieberman, K. Willcox, O. Ghattas, Parameter and state model reduction for large-scale statistical inverse problems, SIAM Journal on Scientific Computing 32 (5) (2010) 2523–2542. doi:10.1137/090775622.

[371] W. E. Nagel, M. M. Resch, D. B. Kroner, High Performance Computing in Science and Engineering'10, Springer, 2019. doi:10.1007/978-3-642-23869-7.

[372] K. Gaska, A. Generowicz, I. Zimoch, J. Ciuła, Z. Iwanicka, A high-performance computing (hpc) based integrated multithreaded model predictive control (mpc) for water supply networks, Architecture, Civil Engineering, Environment 10 (4) (2018).

[373] A. K. Saibaba, T. Bakhos, P. K. Kitanidis, A flexible krylov solver for shifted systems with application to oscillatory hydraulic tomography, SIAM Journal on Scientific Computing 35 (6) (2013) A3001–A3023. doi:10.1137/120902690.

[374] Y. Yang, P. Grover, S. Kar, Coded distributed computing for inverse problems, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 709–719.

[375] A. Lucas, M. Iliadis, R. Molina, A. K. Katsaggelos, Using deep neural networks for inverse problems in imaging: beyond analytical methods, IEEE Signal Processing Magazine 35 (1) (2018) 20–36. doi:10.1109/MSP.2017.2760358.

[376] F. Avilés, A. I. Oliva-Avilés, M. Cen-Puc, Piezoresistivity, strain, and damage self-sensing of polymer composites filled with carbon nanostructures, Advanced Engineering Materials 20 (7) (2018) 1701159. doi:10.1002/adem.201701159.

[377] B. Han, S. Ding, X. Yu, Intrinsic self-sensing concrete and structures: A review, Measurement 59 (2015) 110–128. doi:10.1016/j.measurement.2014.09.048.

[378] J. Han, J. Pan, J. Cai, X. Li, A review on carbon-based self-sensing cementitious composites, Construction and Building Materials 265 (2020) 120764. doi:10.1016/j.conbuildmat.2020.120764.

[379] N. Forintos, T. Czigany, Multifunctional application of carbon fiber reinforced polymer composites: electrical properties of the reinforcing carbon fibers–a short review, Composites Part B: Engineering 162 (2019) 331–343. doi:10.1016/j.compositesb.2018.10.098.

[380] T. N. Tallman, D. J. Smyl, Structural health and condition monitoring via electrical impedance tomography in self-sensing materials: A review, Smart Materials and Structures 29 (12) (2020) 123001. doi:10.1088/1361-665X/abb352.

[381] D. Wang, D. Chung, Through-thickness piezoresistivity in a carbon fiber polymer-matrix structural composite for electrical-resistance-based through-thickness strain sensing, Carbon 60 (2013) 129–138. doi:10.1016/j.carbon.2013.04.005.

[382] M. C. Koecher, J. H. Pande, S. Merkley, S. Henderson, D. T. Fullwood, A. E. Bowden, Piezoresistive in-situ strain sensing of composite laminate structures, Composites Part B: Engineering 69 (2015) 534–541. doi:10.1016/j.compositesb.2014.09.029.

[383] J. d. J. Ku-Herrera, V. La Saponara, F. Avilés, Selective damage sensing in multiscale hierarchical composites by tailoring the location of carbon nanotubes, Journal of Intelligent Material Systems and Structures 29 (4) (2018) 553–562. doi:10.1177/1045389X17711790.

[384] T.-C. Hou, K. J. Loh, J. P. Lynch, Spatial conductivity mapping of carbon nanotube composite thin films by electrical impedance tomography for sensing applications, Nanotechnology 18 (31) (2007) 315501. doi:10.1088/0957-4484/18/31/315501.

[385] L. Groo, J. Nasser, D. Inman, H. Sodano, Fatigue damage tracking and life prediction of fiberglass composites using a laser induced graphene interlayer, Composites Part B: Engineering 218 (2021) 108935. doi:10.1016/j.compositesb.2021.108935.

[386] L. Groo, J. Nasser, D. Inman, H. Sodano, Damage localization in fiberglass-reinforced composites using laser induced graphene, Smart Materials and Structures 30 (3) (2021) 035006. doi:10.1088/1361-665X/abdc0c.

[387] K. K. Talamadupula, S. Povolny, N. Prakash, G. D. Seidel, Piezoresistive detection of simulated hotspots and the effects of low velocity impact at the mesoscale in nanocomposite bonded energetic materials via multiphysics peridynamics modeling, Computational Materials Science 188 (2021) 110211. doi:10.1016/j.commatsci.2020.110211.

[388] K. K. Talamadupula, S. J. Povolny, N. Prakash, G. D. Seidel, Mesoscale strain and damage sensing in nanocomposite bonded energetic materials under low velocity impact with frictional heating via peridynamics, Modelling Simul. Mater. Sci. Eng. 28 (8) (2020) 085011. doi:10.1088/1361-651X/abbfb9.

[389] D.-Y. Yoo, I. You, S.-J. Lee, Electrical and piezoresistive sensing capacities of cement paste with multi-walled carbon nanotubes, Archives of Civil and Mechanical Engineering 18 (2018) 371–384. doi:10.1016/j.acme.2017.09.007.

[390] E. Ricohermoso III, F. Rosenburg, F. Klug, N. Nicoloso, H. F. Schlaak, R. Riedel, E. Ionescu, Piezoresistive carbon-containing ceramic nanocomposites–a review, Open Ceramics (2021) 100057doi:10.1016/j.oceram.2021.100057.

[391] M. C. McCrary-Dennis, O. I. Okoli, A review of multiscale composite manufacturing and challenges, Journal of reinforced plastics and composites 31 (24) (2012) 1687–1711. doi:10.1177/0731684412456612.

[392] P.-C. Ma, N. A. Siddiqui, G. Marom, J.-K. Kim, Dispersion and functionalization of carbon nanotubes for polymer-based nanocomposites: a review, Composites Part A: Applied Science and Manufacturing 41 (10) (2010) 1345–1367. doi:10.1016/j.compositesa.2010.07.003.

[393] S. Gong, Z. H. Zhu, On the mechanism of piezoresistivity of carbon nanotube polymer composites, Polymer 55 (16) (2014) 4136–4149. doi:10.1016/j.polymer.2014.06.024.

[394] N. Hu, Y. Karube, C. Yan, Z. Masuda, H. Fukunaga, Tunneling effect in a polymer/carbon nanotube nanocomposite strain sensor, Acta materialia 56 (13) (2008) 2929–2936. doi:10.1016/j.actamat.2008.02.030.

[395] B. M. Lee, K. J. Loh, A 2d percolation-based model for characterizing the piezoresistivity of carbon nanotube-based films, Journal of materials science 50 (7) (2015) 2973–2983. doi:10.1007/s10853-015-8862-y.

[396] A. Alian, S. Meguid, Multiscale modeling of the coupled electromechanical behavior of multifunctional nanocomposites, Composite Structures 208 (2019) 826–835. doi:10.1016/j.compstruct.2018.10.066.

[397] X. Ren, A. K. Chaurasia, A. I. Oliva-Avilés, J. J. Ku-Herrera, G. D. Seidel, F. Avilés, Modeling of mesoscale dispersion effect on the piezoresistivity of carbon nanotube-polymer nanocomposites via 3d computational multiscale micromechanics methods, Smart Materials and Structures 24 (6) (2015) 065031. doi:10.1088/0964-1726/24/6/065031.

[398] K. K. Talamadupula, G. D. Seidel, Statistical analysis of effective electro-mechanical properties and percolation behavior of aligned carbon nanotube/polymer nanocomposites via computational micromechanics, Computational Materials Science 197 (2021) 110616. doi:10.1016/j.commatsci.2021.110616.

[399] C. Cattin, P. Hubert, Piezoresistance in polymer nanocomposites with high aspect ratio particles, ACS Applied Materials & Interfaces 6 (3) (2014) 1804–1811. doi:10.1021/am404808u.

[400] T. Tallman, K. Wang, An arbitrary strains carbon nanotube composite piezoresistivity model for finite element integration, Applied Physics Letters 102 (1) (2013) 011909. doi:10.1063/1.4774294.

[401] G. Koo, T. Tallman, Higher-order resistivity-strain relations for self-sensing nanocomposites subject to general deformations, Composites Part B: Engineering 190 (2020) 107907. doi:10.1016/j.compositesb.2020.107907.

[402] W. Bao, S. Meguid, Z. Zhu, G. Weng, Tunneling resistance and its effect on the electrical conductivity of carbon nanotube nanocomposites, Journal of Applied Physics 111 (9) (2012) 093726. doi:10.1063/1.4716010.

[403] M. Taya, W. Kim, K. Ono, Piezoresistivity of a short fiber/elastomer matrix composite, Mechanics of materials 28 (1-4) (1998) 53–59. doi:10.1016/S0167-6636(97)00064-1.

[404] Y. Zhao, S. Gschossmann, M. Schagerl, P. Gruener, C. Kralovec, Characterization of the spatial elastoresistivity of inkjet-printed carbon nanotube thin films, Smart Materials and Structures 27 (10) (2018) 105009. doi:10.1088/1361-665X/aad8f1.

[405] P. Gruener, Y. Zhao, M. Schagerl, Characterization of the spatial elastoresistivity of inkjet-printed carbon nanotube thin films for strain-state sensing, in: Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure 2017, Vol. 10169, International Society for Optics and Photonics, 2017, p. 101690F. doi:10.1117/12.2260890.

[406] B. K. Bay, Methods and applications of digital volume correlation, The Journal of Strain Analysis for Engineering Design 43 (8) (2008) 745–760. doi:10.1243/03093247JSA436.

[407] T. Dowrick, D. Holder, Phase division multiplexed eit for enhanced temporal resolution, Physiological measurement 39 (3) (2018) 034005. doi:10.1088/1361-6579/aaad59.

[408] K. J. Loh, T.-C. Hou, J. P. Lynch, N. A. Kotov, Carbon nanotube sensing skins for spatial strain and impact damage identification, Journal of Nondestructive Evaluation 28 (1) (2009) 9–25. doi:10.1007/s10921-009-0043-y.

[409] H. Dai, E. T. Thostenson, Large-area carbon nanotube-based flexible composites for ultra-wide range pressure sensing and spatial pressure mapping, ACS applied materials & interfaces 11 (51) (2019) 48370–48380. doi:10.1021/acsami.9b17100.

[410] S. Nayak, S. Das, Spatial damage sensing ability of metallic particulate-reinforced cementitious composites: Insights from electrical resistance tomography, Materials & Design 175 (2019) 107817. doi:10.1016/j.matdes.2019.107817.

[411] S. Gupta, J. G. Gonzalez, K. J. Loh, Self-sensing concrete enabled by nano-engineered cement-aggregate interfaces, Structural Health Monitoring 16 (3) (2017) 309–323. doi:10.1177/1475921716643867.

[412] T. Tallman, S. Gungor, G. Koo, C. Bakis, On the inverse determination of displacements, strains, and stresses in a carbon nanofiber/polyurethane nanocomposite from conductivity data obtained via electrical impedance tomography, Journal of Intelligent Material Systems and Structures (2017) 1–13doi:10.1177/1045389X17692053.

[413] H. Hassan, T. N. Tallman, Failure prediction in self-sensing nanocomposites via genetic algorithm-enabled piezoresistive inversion, Structural Health Monitoring 19 (3) (2020) 765–780. doi:10.1177/147592171986306.

[414] T. Tallman, K. Wang, An inverse methodology for calculating strains from conductivity changes in piezoresistive nanocomposites, Smart Materials and Structures 25 (11) (2016) 115046. doi:10.1088/0964-1726/25/11/115046.

[415] H. Hassan, T. N. Tallman, A comparison of metaheuristic algorithms for solving the piezoresistive inverse problem in self-sensing materials, IEEE Sensors Journal 21 (1) (2020) 659–666. doi:10.1109/JSEN.2020.3014554.

[416] H. Ghaednia, C. E. Owens, L. E. Keiderling, K. M. Varadarajan, A. J. Hart, J. H. Schwab, T. N. Tallman, Is machine learning able to detect and classify failure in piezoresistive bone cement based on electrical signals?, arXiv preprint arXiv:2010.12147 (2020). doi:10.48550/arXiv.2010.12147.

[417] L. Zhao, T. Tallman, G. Lin, Spatial damage characterization in self-sensing materials via neural network-aided electrical impedance tomography: A computational study, arXiv preprint arXiv:2010.01674 (2020). doi:10.48550/arXiv.2010.01674.

[418] H. Hassan, T. Tallman, Damage sizing in self-sensing materials using a genetic algorithm-supplemented electrical impedance tomography formulation, in: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020, Vol. 11379, International Society for Optics and Photonics, 2020, p. 113790B. doi:10.1117/12.2557025.

[419] D. S. Holder, Electrical impedance tomography: methods, history and applications, CRC Press, 2004.

[420] E. Weinan, The dawning of a new era in applied mathematics, Notices of the American Mathematical Society 68 (4) (2021) 565–571. doi:10.1090/noti2259.

[421] J. Han, A. Jentzen, E. Weinan, Solving high-dimensional partial differential equations using deep learning, Proceedings of the National Academy of Sciences 115 (34) (2018) 8505–8510. doi:10.1073/pnas.1718942115.

[422] S. Lee, J. Ha, M. Zokhirova, H. Moon, J. Lee, Background information of deep learning for structural engineering, Archives of Computational Methods in Engineering 25 (1) (2018) 121–129. doi:10.1007/s11831-017-9237-0.

[423] H. A. Simon, The sciences of the artificial. 3rd (1996).

[424] D. A. Schon, Reflective practitioner, Vol. 5126, Basic books, 1983.

[425] R. Buchanan, Wicked problems in design thinking, Design issues 8 (2) (1992) 5–21.

[426] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, et al., Self-driving cars: A survey, Expert Systems with Applications (2020) 113816doi:10.1016/j.eswa.2020.113816.

[427] T. Mitchell, Machine Learning, McGraw Hill, 1997.

[428] A. Lipponen, J. M. Huttunen, S. Romakkaniemi, H. Kokkola, V. Kolehmainen, Correction of model reduction errors in simulations, SIAM Journal on Scientific Computing 40 (1) (2018) B305–B327. doi:10.1137/15M1052421.

[429] J. Kaipio, E. Somersalo, Statistical inverse problems: Discretization, model reduction and inverse crimes, Journal of Computational and Applied Mathematics 198 (2007) 493–504. doi:10.1016/j.cam.2005.09.027.

[430] S. Arridge, J. Kaipio, V. Kolehmainen, M. Schweiger, E. Somersalo, T. Tarvainen, M. Vauhkonen, Approximation errors and model reduction with an application in optical diffusion tomography, Inverse Problems 22 (2006) 175–195. doi:10.1088/0266-5611/22/1/010.

[431] J. Koponen, T. Lähivaara, J. Kaipio, M. Vauhkonen, Model reduction in acoustic inversion by artificial neural network, arXiv preprint arXiv:2105.02225 (2021). doi:10.1121/10.0007049.

[432] S. Banert, A. Ringh, J. Adler, J. Karlsson, O. Oktem, Data-driven nonsmooth optimization, SIAM Journal on Optimization 30 (1) (2020) 102–131. doi:doi.org/10.1137/18M1207685.

[433] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, J. M. Solomon, Dynamic graph cnn for learning on point clouds, Acm Transactions On Graphics (tog) 38 (5) (2019) 1–12. doi:10.1145/3326362.

[434] W. Herzberg, D. B. Rowe, A. Hauptmann, S. J. Hamilton, Graph Convolutional Networks for Model-Based Learning in Nonlinear Inverse Problems, IEEE Transactions on Computational Imaging 7 (2021) 1341–1353. doi:10.1109/TCI.2021.3132190.

[435] J. E. Froyd, P. C. Wankat, K. A. Smith, Five major shifts in 100 years of engineering education, Proceedings of the IEEE 100 (Special Centennial Issue) (2012) 1344–1360. doi:10.1109/JPROC.2012.2190167.

[436] W. M. Bulleit, What Makes an Engineering Education an Engineering Education?, in: Structures Congress 2012, American Society of Civil Engineers, Chicago, Illinois, United States, 2012, pp. 1143–1151. doi:10.1061/9780784412367.102.

[437] B. V. Koen, Debunking contemporary myths concerning engineering, in: Philosophy and engineering: Reflections on practice, principles and process, Springer, 2013, pp. 115–137. doi:10.1007/978-94-007-7762-0_10.

[438] C. L. Dym, A. M. Agogino, O. Eris, D. D. Frey, L. J. Leifer, Engineering design thinking, teaching, and learning, Journal of engineering education 94 (1) (2005) 103–120. doi:10.1002/j.2168-9830.2005.tb00832.x.

[439] F. Doridot, Towards an'engineered epistemology'?, Interdisciplinary science reviews 33 (3) (2008) 254–262. doi:10.1179/174327908X366941.

[440] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, IEEE transactions on neural networks and learning systems (2020). doi:10.1109/TNNLS.2020.2978386.

[441] A. Gallet, A. Liew, I. Hajirasouliha, D. Smyl, Influence zones of continuous beam systems, Structures 68 (2024) 107069. doi:https://doi.org/10.1016/j.istruc.2024.107069.

[442] E. Betti, Teoria della elasticita, Il Nuovo Cimento 7–8 (1) (1872) 5–21. doi:10.1007/BF02824590.

[443] L. S. Blake (Ed.), Civil Engineer's Reference Book, Elsevier, 1989. doi:10.1016/C2013-0-00875-4.

[444] I. A. Karnovsky, O. Lebed, Advanced Methods of Structural Analysis, Springer US, Boston, MA, 2010. doi:10.1007/978-1-4419-1047-9.

[445] R. C. Hibbeler, Structural Analysis, ninth edition Edition, Pearson Prentice Hall, Upper Saddle River, N.J, 2015.

[446] V. Hosur, S. Bhavikatti, Influence lines for bending moments in beams on elastic foundations, Computers & Structures 58 (6) (1996) 1225–1231. doi:10.1016/0045-7949(95)00219-7.

[447] G. Fiorillo, M. Ghosn, Application of influence lines for the ultimate capacity of beams under moving loads, Engineering Structures 103 (2015) 125–133. doi:10.1016/j.engstruct.2015.09.003.

[448] E. Buckley, Basic Influence Line Equations for Continuous Beams and Rigid Frames, Journal of Structural Engineering 123 (10) (1997) 1416–1420. doi:10.1061/(ASCE)0733-9445(1997)123:10(1416).

[449] X. Zheng, D.-H. Yang, T.-H. Yi, H.-N. Li, Development of bridge influence line identification methods based on direct measurement data: A comprehensive review and comparison, Engineering Structures 198 (2019) 109539. doi:10.1016/j.engstruct.2019.109539.

[450] X. Zhu, S. Law, Structural Health Monitoring Based on Vehicle-Bridge Interaction: Accomplishments and Challenges, Advances in Structural Engineering 18 (12) (2015) 1999–2015. doi:10.1260/1369-4332.18.12.1999.

[451] Z.-W. Chen, Q.-L. Cai, S. Zhu, Damage quantification of beam structures using deflection influence lines, Structural Control and Health Monitoring 25 (11) (2018) e2242. doi:10.1002/stc.2242.

[452] P. Kuklik, Zone of influence, in: P. Bobrowsky, B. Marker (Eds.), Encyclopedia of Engineering Geology, Springer International Publishing, Cham, 2018, pp. 1–2. doi:10.1007/978-3-319-12127-7-309-1.

[453] J. Yang, Influence Zone for End Bearing of Piles in Sand, Journal of Geotechnical and Geoenvironmental Engineering 132 (9) (2006) 1229–1237. doi:10.1061/(ASCE)1090-0241(2006)132:9(1229).

[454] S. Ekanayake, D. Liyanapathirana, C. Leo, Influence zone around a closed-ended pile during vibratory driving, Soil Dynamics and Earthquake Engineering 53 (2013) 26–36. doi:10.1016/j.soildyn.2013.06.005.

[455] P. Kuklík, M. Kopáčková, Efficient analytical model for calculation of the influence zone inside the subsoil below foundations slabs, Acta Polytechnica 44 (5-6) (Jan. 2004). doi:10.14311/632.

[456] C. Cao, Z. Xu, J. Chai, Y. Qin, J. Cao, Determination method for influence zone of pumped storage underground cavern and drainage system, Journal of Hydrology 595 (2021) 126018. doi:10.1016/j.jhydrol.2021.126018.

[457] D. Jakovovic, A. D. Werner, P. G. de Louw, V. E. Post, L. K. Morgan, Saltwater upconing zone of influence, Advances in Water Resources 94 (2016) 75–86. doi:10.1016/j.advwatres.2016.05.003.

[458] A. Goel, P. Kumar, Zone of influence for particle number concentrations at signalised traffic intersections, Atmospheric Environment 123 (2015) 25–38. doi:10.1016/j.atmosenv.2015.10.054.

[459] M. Alvin, K. M. Adhinugraha, S. Alamri, U. Mir, Influence zone expansion for reverse k nearest neighbours query, Multimedia Tools and Applications (Aug. 2021). doi:10.1007/s11042-021-11275-3.

[460] M. A. Cheema, W. Zhang, X. Lin, Y. Zhang, Efficiently processing snapshot and continuous reverse k nearest neighbors queries, The VLDB Journal 21 (5) (2012) 703–728. doi:10.1007/s00778-012-0265-y.

[461] K. Yoshizawa, B. L. N. Kennett, Determination of the influence zone for surface wave paths, Geophysical Journal International 149 (2) (2002) 440–453. doi:10.1046/j.1365-246X.2002.01659.x.

[462] B. B. Casper, H. J. Schenk, R. B. Jackson, Defining a plant's belowground zone of influence, Ecology 84 (9) (2003) 2313–2321. doi:10.1890/02-0287.

[463] A. Schlueter, F. Thesseling, Building information model based energy/exergy performance assessment in early design stages, Automation in Construction 18 (2) (2009) 153–163. doi:10.1016/j.autcon.2008.07.003.

[464] D. Sinclair, A. Tait, L. Carmichael, RIBA Plan of Work Overview (2020).

[465] M. P. Saka, Z. W. Geem, Mathematical and Metaheuristic Applications in Design Optimization of Steel Frame Structures: An Extensive Review, Mathematical Problems in Engineering 2013 (2013) 1–33. doi:10.1155/2013/271031.

[466] European Committee for Standardisation, BS EN 1993-1-1:2005+A1:2014 Eurocode 3. Design of Steel Structures. General Rules and Rules for Buildings, BSI British Standards, 2015. doi:10.3403/03270565.

[467] European Committee for Standardisation, BS EN 10365:2017 Hot Rolled Steel Channels, I and H Sections. Dimensions and Masses, BSI British Standards, 2017. doi:10.3403/30318327.

[468] European Committee for Standardisation, BS EN 1990:2002+A1:2005 Eurocode. Basis of Structural Design, BSI British Standards, 2010. doi:10.3403/02612036.

[469] A. N. Shiryayev, Interpolation and Extrapolation of Stationary Random Sequences, in: A. N. Shiryayev (Ed.), Selected Works of A. N. Kolmogorov, Springer Netherlands, 1992, pp. 272–280. doi:10.1007/978-94-011-2260-3_28.

[470] L. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353. doi:10.1016/S0019-9958(65)90241-X.

[471] L. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 100 (1999) 9–34. doi:10.1016/S0165-0114(99)80004-9.

[472] H. U. Köylüoğlu, A. Ş. Çakmak, S. R. K. Nielsen, Interval Algebra to Deal with Pattern Loading and Structural Uncertainties, Journal of Engineering Mechanics 121 (11) (1995) 1149–1157. doi:10.1061/(ASCE)0733-9399(1995)121:11(1149).

[473] R. L. Mullen, R. L. Muhanna, Bounds of Structural Response for All Possible Loading Combinations, Journal of Structural Engineering 125 (1) (1999) 98–106. doi:10.1061/(ASCE)0733-9445(1999)125:1(98).

[474] A. Labuschagne, N. Van Rensburg, A. Van Der Merwe, Comparison of linear beam theories, Mathematical and Computer Modelling 49 (1-2) (2009) 20–30. doi:10.1016/j.mcm.2008.06.006.

[475] A. Gallet, D. Smyl, IZ kmax: Influence zone results and design datasets (2024). doi:10.15131/shef.data.24433918.
URL https://figshare.com/s/2a390776378f72f3bf06

[476] C. Preisinger, Linking Structure and Parametric Geometry, Architectural Design 83 (2) (2013) 110–113. doi:10.1002/ad.1564.

[477] L. Deng, H. Wu, W. He, T. Ling, G. Liu, Genuine Influence Line and Influence Surface Identification from Measured Bridge Response Considering Vehicular Wheel Loads, Journal of Bridge Engineering 28 (2) (2023) 04022145. doi:10.1061/JBENF2.BEENG-5604.

[478] X. Zheng, D. Yang, T. Yi, H. Li, Bridge influence surface identification method considering the spatial effect of vehicle load, Structural Control and Health Monitoring 28 (8) (2021). doi:10.1002/stc.2769.

[479] E. Orakdogen, K. Girgin, Direct determination of influence lines and surfaces by F.E.M., Structural Engineering and Mechanics 20 (3) (2005) 279–292. doi:10.12989/SEM.2005.20.3.279.

[480] A. Memari, H. West, Computation of bridge design forces from influence surfaces, Computers & Structures 38 (5-6) (1991) 547–556. doi:10.1016/0045-7949(91)90006-8.

[481] A. Kaveh, M. Z. Kabir, M. Bohlool, Optimal Design of Multi-Span Pitched Roof Frames with Tapered Members, Periodica Polytechnica Civil Engineering (Oct. 2018). doi:10.3311/PPci.13107.

[482] D. Veenendaal, J. Coenders, J. Vambersky, M. West, Design and optimization of fabric-formed beams and trusses: Evolutionary algorithms and form-finding, Structural Concrete 12 (4) (2011) 241–254. doi:10.1002/suco.201100020.

[483] European Committee for Standardisation, BS EN 1992-1-1 Eurocode 2: Design of Concrete Structures - Part 1-1: General Rules and Rules for Buildings, British Standards Institution, 2015. doi:10.3403/03178016U.

[484] Technical Committee, Code of Practice for Structural Use of Concrete 2013, Buildings Department, Hong Kong, 2020.

[485] D. Y. B. Ho, Pattern load analyses for irregular floor systems, Proceedings of the Institution of Civil Engineers - Structures and Buildings 168 (6) (2015) 433–440. doi:10.1680/stbu.13.00116.

[486] D.-H. Yang, T.-H. Yi, H.-N. Li, Y.-F. Zhang, Correlation-Based Estimation Method for Cable-Stayed Bridge Girder Deflection Variability under Thermal Action, Journal of Performance of Constructed Facilities 32 (5) (2018) 04018070. doi:10.1061/(ASCE)CF.1943-5509.0001212.

[487] D.-H. Yang, T.-H. Yi, H.-N. Li, Y.-F. Zhang, Monitoring and analysis of thermal effect on tower displacement in cable-stayed bridge, Measurement 115 (2018) 249–257. doi:10.1016/j.measurement.2017.10.036.

[488] L. Zhao, T. Tallman, G. Lin, Spatial Damage Characterization in Self-Sensing Materials via Neural Network-Aided Electrical Impedance Tomography: A Computational Study, ES Materials & Manufacturing (2021). doi:10.30919/esmm5f919.

[489] J. L. Mueller, S. Siltanen, Linear and Nonlinear Inverse Problems with Practical Applications, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2012. doi:10.1137/1.9781611972344.

[490] N. Pollini, Gradient-based prestress and size optimization for the design of cable domes, International Journal of Solids and Structures 222–223 (2021) 111028. doi:10.1016/j.ijsolstr.2021.03.015.

[491] O. Altay, O. Cetindemir, I. Aydogdu, Size optimization of planar truss systems using the modified salp swarm algorithm, Engineering Optimization (2023) 1–17doi:10.1080/0305215X.2022.2160449.

[492] S. D. Daxini, J. M. Prajapati, Parametric shape optimization techniques based on Meshless methods: A review, Structural and Multidisciplinary Optimization 56 (5) (2017) 1197–1214. doi:10.1007/s00158-017-1702-8.

[493] B. D. Upadhyay, S. S. Sonigra, S. D. Daxini, Numerical analysis perspective in structural shape optimization: A review post 2000, Advances in Engineering Software 155 (2021) 102992. doi:10.1016/j.advengsoft.2021.102992.

[494] I. Hajirasouliha, K. Pilakoutas, H. Moghaddam, Topology optimization for the seismic design of truss-like structures, Computers & Structures 89 (7-8) (2011) 702–711. doi:10.1016/j.compstruc.2011.02.003.

[495] O. Sigmund, K. Maute, Topology optimization approaches: A comparative review, Structural and Multidisciplinary Optimization 48 (6) (2013) 1031–1055. doi:10.1007/s00158-013-0978-6.

[496] A. G. Weldeyesus, J. Gondzio, L. He, M. Gilbert, P. Shepherd, A. Tyas, Adaptive solution of truss layout optimization problems with global stability constraints, Structural and Multidisciplinary Optimization 60 (5) (2019) 2093–2111. doi:10.1007/s00158-019-02312-9.

[497] H. E. Fairclough, M. Gilbert, Layout optimization of long-span structures subject to self-weight and multiple load-cases, Structural and Multidisciplinary Optimization 65 (7) (2022) 197. doi:10.1007/s00158-022-03242-9.

[498] Q. Zaheer, M. M. Manzoor, M. J. Ahamad, A review on developing optimization techniques in civil engineering, Engineering Computations (Feb. 2023). doi:10.1108/EC-01-2022-0034.

[499] R. F. Coelho, T. Tysmans, E. Verwimp, Form finding & structural optimization: A project-based course for graduate students in civil and architectural engineering, Structural and Multidisciplinary Optimization 49 (6) (2014) 1037–1046. doi:10.1007/s00158-013-1021-7.

[500] H. Adeli, C. Yeh, Perceptron Learning in Engineering Design, Computer-Aided Civil and Infrastructure Engineering 4 (4) (1989) 247–256. doi:10.1111/j.1467-8667.1989.tb00026.x.

[501] L. Berke, S. Patnaik, P. Murthy, Optimum design of aerospace structural components using neural networks, Computers & Structures 48 (6) (1993) 1001–1010. doi:10.1016/0045-7949(93)90435-G.

[502] S. Hanna, Inductive machine learning of optimal modular structures: Estimating solutions using support vector machines, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 21 (4) (2007 FAL) 351–366. doi:10.1017/S0890060407000327.

[503] C. Xiang, D. Wang, Y. Pan, A. Chen, X. Zhou, Y. Zhang, Accelerated topology optimization design of 3D structures based on deep learning, Structural and Multidisciplinary Optimization 65 (3) (2022) 99. doi:10.1007/s00158-022-03194-0.

[504] ZG. Nie, T. Lin, HL. Jiang, LB. Kara, TopologyGAN: Topology Optimization Using Generative Adversarial Networks Based on Physical Fields Over the Initial Domain, Journal of Mechanical Design 143 (3) (Mar. 2021). doi:10.1115/1.4049533.

[505] W. Liao, X. Lu, Y. Huang, Z. Zheng, Y. Lin, Automated structural design of shear wall residential buildings using generative adversarial networks, Automation in Construction 132 (2021) 103931. doi:10.1016/j.autcon.2021.103931.

[506] R. Danhaive, C. T. Mueller, Design subspace learning: Structural design space exploration using performance-conditioned generative modeling, Automation in Construction 127 (2021) 103664. doi:10.1016/j.autcon.2021.103664.

[507] L. Bleker, R. Pastrana, P. O. Ohlbrock, P. D'Acunto, Structural Form-Finding Enhanced by Graph Neural Networks, in: C. Gengnagel, O. Baverel, G. Betti, M. Popescu, M. R. Thomsen, J. Wurm (Eds.), Towards Radical Regeneration, Springer International Publishing, Cham, 2023, pp. 24–35. doi:10.1007/978-3-031-13249-0_3.

[508] European Committee for Standardisation, BS EN 1990:2002+A1:2005 Eurocode. Basis of Structural Design, BSI British Standards, 2002. doi:10.3403/03202162.

[509] P. Hajela, L. Berke, Neurobiological computational models in structural analysis and design, Computers & Structures 41 (4) (1991) 657–667. doi:10.1016/0045-7949(91)90178-O.

[510] S. M. Mojtabaei, J. Becque, I. Hajirasouliha, R. Khandan, Predicting the buckling behaviour of thin-walled structural elements using machine learning methods, Thin-Walled Structures 184 (2023) 110518. doi:10.1016/j.tws.2022.110518.

[511] H. Adeli, H. S. Park, Optimization of space structures by neural dynamics, Neural Networks 8 (5) (1995) 769–781. doi:10.1016/0893-6080(95)00026-V.

[512] H. T. Mai, D. D. Mai, J. Kang, J. Lee, J. Lee, Physics-informed neural energy-force network: A unified solver-free numerical simulation for structural optimization, Engineering with Computers 40 (1) (2024) 147–170. doi:10.1007/s00366-022-01760-0.

[513] J. Yan, Q. Zhang, Q. Xu, ZR. Fan, HJ. Li, W. Sun, GY. Wang, Deep learning driven real time topology optimisation based on initial stress learning, Advanced Engineering Informatics 51 (Jan. 2022). doi:10.1016/j.aei.2021.101472.

[514] G. G. Wang, S. Shan, Review of Metamodeling Techniques in Support of Engineering Design Optimization, Journal of Mechanical Design 129 (4) (2007) 370–380. doi:10.1115/1.2429697.

[515] S. Koziel, L. Leifsson, Surrogate-Based Modeling and Optimization: Applications in Engineering, Springer New York, New York, NY, 2013. doi:10.1007/978-1-4614-7551-4.

[516] T. M. Mitchell, Machine Learning, McGraw-Hill Series in Computer Science, McGraw-Hill, New York, 1997.

[517] M. Kubat, An Introduction to Machine Learning, third edition Edition, Springer Nature, Cham, Switzerland, 2021. doi:10.1007/978-3-030-81935-4.

[518] A. J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (3) (2004) 199–222. doi:10.1023/B:STCO.0000035301.49549.88.

[519] K. P. Murphy, Probabilistic Machine Learning: An Introduction, Adaptive Computation and Machine Learning Series, The MIT Press, Cambridge, Massachusetts, 2022.

[520] J.-H. Kim, Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap, Computational Statistics & Data Analysis 53 (11) (2009) 3735–3745. doi:10.1016/j.csda.2009.04.009.

[521] A. A. Dennis, S. E. Rigby, The Direction-encoded Neural Network: A machine learning approach to rapidly predict blast loading in obstructed environments, International Journal of Protective Structures (Jun. 2023). doi:10.1177/20414196231177364.

[522] V. Vakharia, M. Shah, P. Nair, H. Borade, P. Sahlot, V. Wankhede, Estimation of Lithium-ion Battery Discharge Capacity by Integrating Optimized Explainable-AI and Stacked LSTM Model, Batteries 9 (2) (2023) 125. doi:10.3390/batteries9020125.

[523] E. Zhang, M. Dao, G. E. Karniadakis, S. Suresh, Analyses of internal structures and defects in materials using physics-informed neural networks, Science Advances 8 (7) (2022) eabk0644. doi:10.1126/sciadv.abk0644.

[524] H. Jeong, JS. Bai, CP. Batuwatta-Gamage, C. Rathnayaka, Y. Zhou, YT. Gu, A Physics-Informed Neural Network-based Topology Optimization (PINNTO) framework for structural optimization, Engineering Structures 278 (Mar. 2023). doi:10.1016/j.engstruct.2022.115484.

Appendices

Appendix A - Designing continuous beam systems

The purpose of this appendix is to provide some additional context on the challenges faced when designing continuous beam systems. The design of a continuous beam system as shown in Figure 7.1 contains multiple challenges which are directly linked to the nature of inverse problems. First, it is important to properly constrain the solution space, since for any given design scenario there generally exist multiple compatible solutions. This reflects the "non-unique" characteristic of ill-posed inverse problems as discussed in Chapter 3. This issue is generally avoided by restraining the solution space to a small subset of cross-sections, such as universal UK beam sections that are appropriate for a given design objective such as minimum depth or weight.

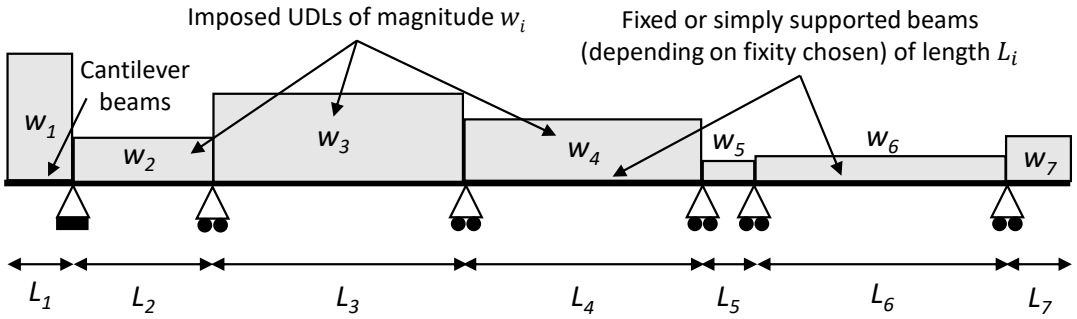
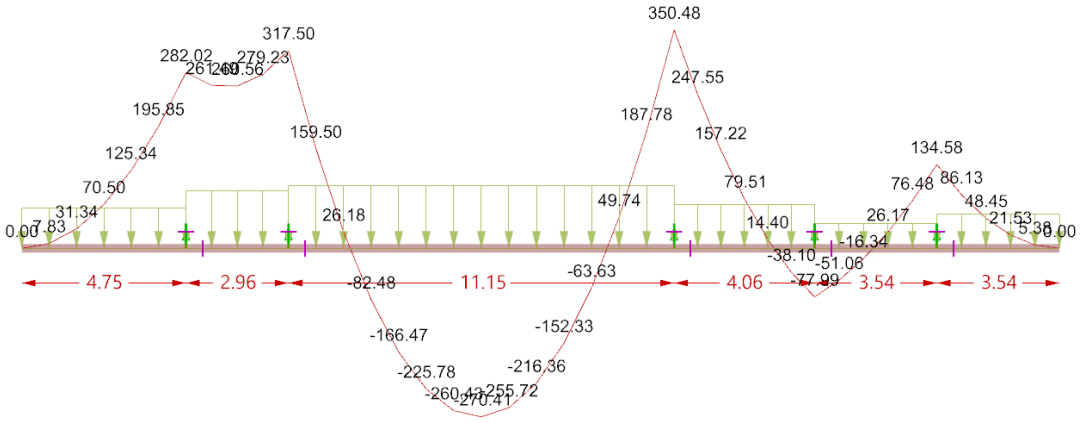


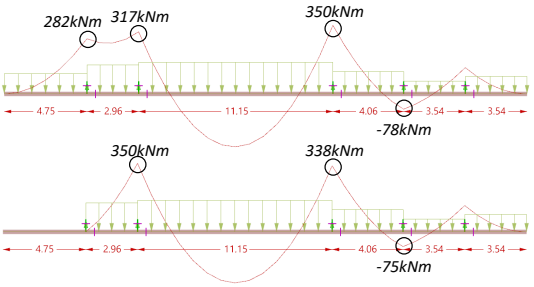
Figure 7.1: Example problem of designing continuous beam systems with w_i UDLs and L_i spans

Furthermore, unlike a system of simply supported beams that are each structurally determinate, the inter-connectivity of continuous beam systems renders such systems indeterminate. The moment and shear forces experienced by any single beam is a function of the pattern-loading as well as the provided cross-sectional capacities of adjacently lying beams, and hence requires iterating through all of the possible pattern loading and testing out various combination of cross-sections until an appropriate selection of beam elements has been established. Figure 7.2 highlights this problem, indicating how changes in loading or cross-section has an impact onto the entire beam system, not just the one where the change occurred.

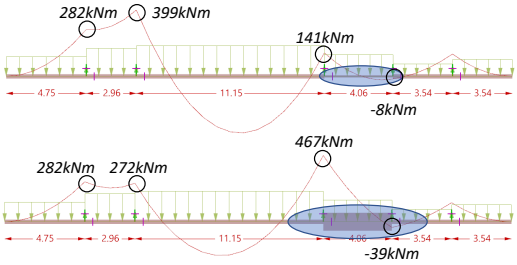
This is reflective of another common feature of inverse problems, namely the relative



(a) Major axis moment for beam system of identical cross-section.



(b) Pattern-loading variation impact.



(c) Cross-section variation impact.

Figure 7.2: Impact of cross-section and loading changes on major-axis moments of an indeterminate beam system.

"unstable" nature of solutions, where relatively small changes (such as the cross-section of one particular beam) can change the maximum experienced moment of a beam drastically (such as from 141kNm to 467kNm as shown in 7.2c for the third beam from the right). In literature, this feature in design has often been defined as ill-structured [178], open-ended [179] or even "wicked" [180].

The conventional methodology is to evaluate appropriate section sizes is through brute-force deterministic models, which iterate through several solutions (either based on gradients or heuristically). Such schemes typically assume some initial beam sizes, analyse the structure based on that system's stiffness, create an envelope of member forces and displacements, conduct design checks, and repeat with different section sizes until a satisfactory combination has been found.

For simple design problem such as the one discussed above, this process can be relatively streamlined without too significant computational expense, although the simplicity of the problem would often mean that in practice engineers are likely to simply test out various cross-sections manually until a sufficiently good solution has been found. An alter-

native solution is to instead take advantage of the known data relationship of completed designs, and create a mapping function between the design inputs (length of beams, loads, fixity conditions and location of support points) and the cross-sectional outputs (required moment of inertia, shear area) that ensures a viable design based on learned data. This type of methodology could be especially beneficial for designs that typically require computationally expensive analysis (non-linear, dynamic, p-delta, modal, etc.). If a working model can be shown for this proof-of-concept problem, then this methodology could be expanded to more complicated design scenarios.

Appendix B - Data generation model

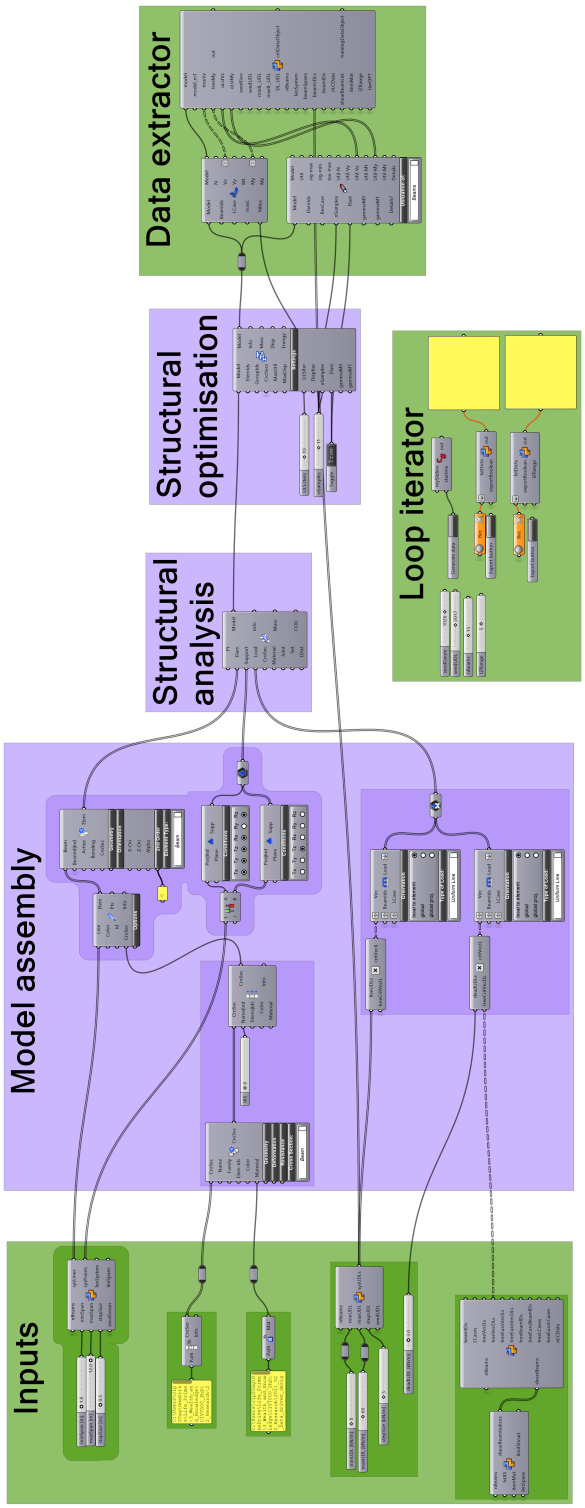


Figure 7.3: Overview of the data generation model used to generate the CBeamXP dataset [6] using Grasshopper [32] and Karamba3D [476].

Appendix C - Physics informed neural network poster

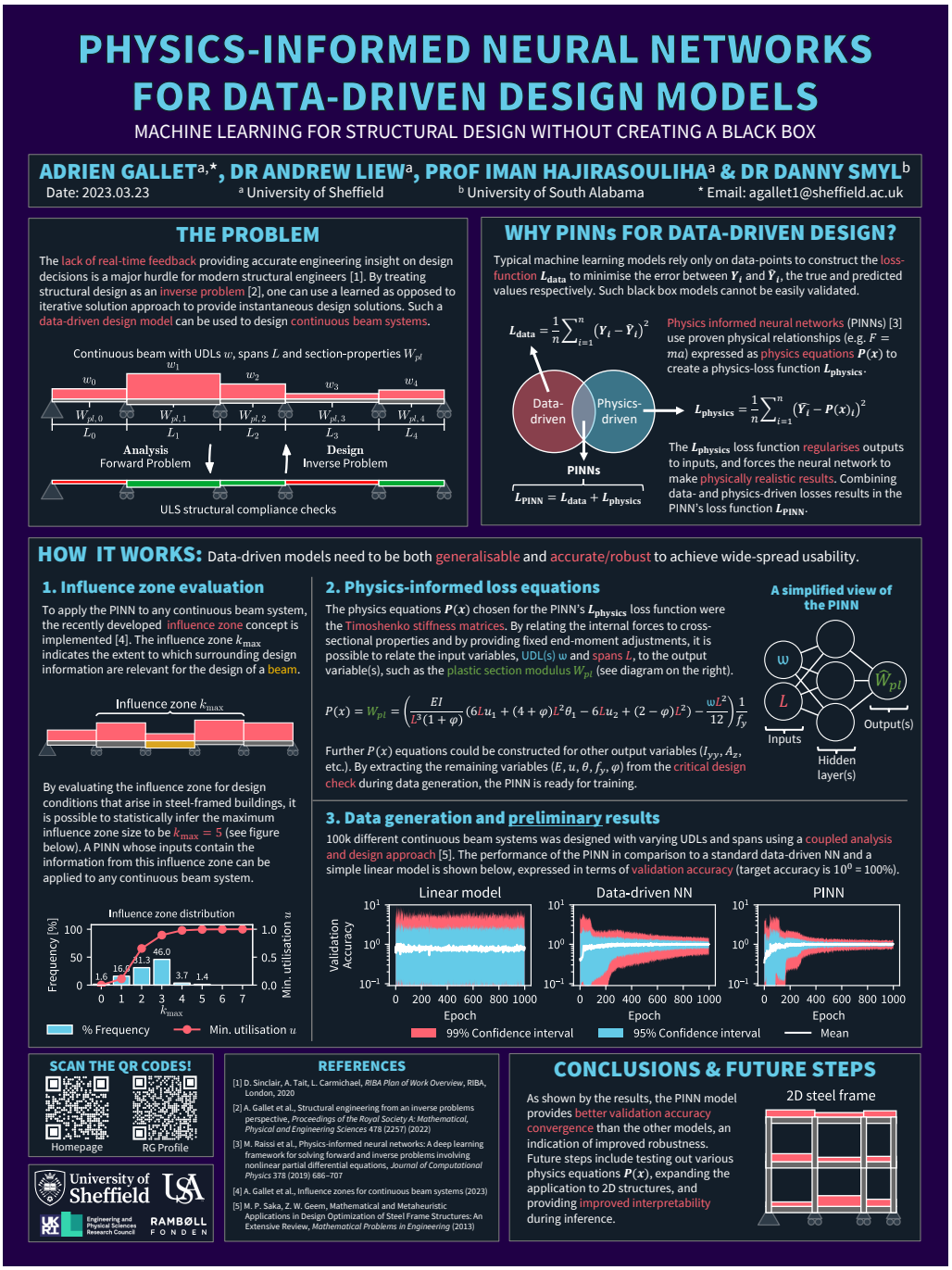


Figure 7.4: Poster submitted to the *Institution of Structural Engineers' Young Researcher's Conference* on March 23rd, 2023.