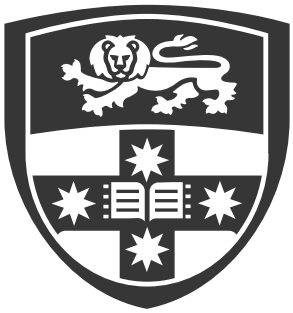


THE UNIVERSITY OF SYDNEY

Machine Learning and Metaheuristics within the Built Environment: From Design to Maintenance

Thomas Lewis Hielscher

Supervisor: Associate Professor Ali Hadigheh

A dissertation submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Civil Engineering

Faculty of Engineering

The University of Sydney

March, 2026

Statement of Originality

This to certify that the content of this thesis is my own work. This thesis has not been submitted for any other degree or purpose.

I certify that the intellectual content of this thesis is the product of my own work, and that all assistance received in preparing this thesis and all sources have been acknowledged. No content produced by generative AI tools has been used in the preparation of this thesis. This research was supported by an Australian Government Research Training Program (RTP) Scholarship.

Tommy Hielscher

Acknowledgements

First and foremost, I would like to thank my parents, Lesley and Rob, for their unwavering love and support, without which this thesis would not have been possible. Their encouragement has carried me through many challenges and has fostered a sense of great resilience and perseverance within me. To you both, I owe everything.

To my partner, Sabrina, you have improved my life immeasurably. Your love and support has remained a constant factor that always brightens my day, regardless of its events. You have been a cornerstone of my success and a light in my life that I treasure deeply.

I would also like to thank my supervisor, Ali, who has guided my research. His support has enriched my experience and his feedback has been useful in the development and progression of this thesis.

Finally, I would like to thank my friends who have never failed to make me laugh, even when times are tough. Weekly card nights with Nick, Cosette, and Matt have been a deep source of joy for me.

There are many people who have supported me throughout this period of my life, which has been filled with challenges, successes, setbacks, and laughter. Without you all, this experience would not have been nearly as rewarding or enjoyable.

Thank you.

Authorship Attribution Statement

Published Papers:

The following paper is one of the works provided in Chapter 4

- **Hielscher, T**. and Hadigheh, S.A., 2025. Optimizing memory-efficient multimodal networks for image classification using differential evolution. *Applied Soft Computing*, *171*, p.112714. DOI: https://doi.org/10.1016/j.asoc.2025.112714

  T. Hielscher: Writing – Original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. S.A. Hadigheh: Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

The following paper is one of the works provided in Chapter 5

- **Hielscher, T**. and Hadigheh, S.A., 2025. Transformer Hypernetworks for Complex Structural Representation. *Journal of Computing in Civil Engineering*, DOI: https://doi.org/10.1061/JCCEE5.CPENG-6388

  T. Hielscher: Writing – Original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. S.A. Hadigheh: Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

The following paper is one of the works provided in Chapter 6

- **Hielscher, T**., Khalil, S., Virgona, N. and Hadigheh, S.A., 2023, November. A neural network based digital twin model for the structural health monitoring of reinforced concrete bridges. *Structures* (Vol. 57, p. 105248). Elsevier. DOI: https://doi.org/10.1016/j.istruc.2023.105248

  T. Hielscher: Writing – Original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. S. Khalil: Writing – Original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. N. Virgona: Writing – Original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. S.A. Hadigheh: Writing – review & editing, Funding acquisition, Supervision, Methodology, Conceptualization, Resources, Project administration.

In addition to the authorship attribution statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Tommy Hielscher, 17/2/2026

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Ali Hadigheh, 17/2/2026

Abstract

Despite significant technological development throughout academia and comparable sectors of the economy, our built environment is increasingly burdened by a lack of automation, data silos, disjointed workflows, slow technological adoption, and uninformed decision making. With lagging productivity trends throughout engineering design, structural maintenance, and long-term asset monitoring practices, these challenges appear to be ubiquitous and pervasive. Each phase of the asset life cycle suffers from unique inefficiencies that have contributed to a growing gap between published research outcomes and the experience of practitioners who still rely on traditional approaches. From a design perspective, metaheuristic algorithms have tremendous potential to empower engineers and architects across applied schematic, structural, and parametric optimisation tasks. However, sensitivity to initialised hyperparameter sets, inherent implementation challenges, and the '*a priori*' expertise required to effectively deploy these algorithms have remained significant barriers impeding their broader adoption across practical design projects. A similar trend may be observed within the research context of predictive design. While a broad variety of compelling machine learning models have been developed within the field of predictive 3D reconstruction, these solutions largely fail to meet the complexity and scale of modern design tasks. Consequently, research within this field often heavily relies on object-based open-source datasets, ignoring the practical requirements of engineers and architects who would substantially benefit from these predictive tools.

Analogously, while the academic field of structural health monitoring has transformed in recent years by adopting cutting-edge sensors and deep learning models, the industry itself has not been mobilised towards a similar level of technological adoption. Widely used traditional forms of nondestructive evaluation are still heavily burdened by subjective condition ratings, inefficient periodic visual inspections, time-consuming reporting, and a reactive approach to preventing structural deterioration. Simultaneously, utilised assets towards the end of their service life are degrading at a rate that is entirely unmanageable by conventional means, with aging infrastructure outpacing inspection capacity across developed nations. The accumulative impacts of aging infrastructure and inadequate maintenance procedures have the potential to drastically reduce the serviceability and safety of our infrastructure at a global scale.

Across these domains, there exists a pervasive disconnect between published academic results and practical outcomes within industry. This thesis proposes that some of the critical limitations that have inhibited the spread of innovation outside of academia may be resolved by studying complementary relationships between established approaches. Namely, there are unrealised research opportunities at the interdisciplinary intersection between metaheuristic and machine learning studies which have far-reaching implications throughout the asset lifecycle. By automating challenging facets of algorithmic design, the barriers that have prevented practitioners

from adopting these tools may be significantly alleviated. Similarly, by vastly improving the resolution and predictive capabilities of sensor-driven deep learning systems, the viability of these promising technologies may be further justified for practical real-world monitoring applications.

The contributions of this thesis are as follows. (1) The development of hybrid algorithms that utilise complementary aspects of metaheuristics and machine learning to reduce the sensitivity of algorithmic design tools to manually tuned hyperparameter sets. (2) The creation of novel methods to scale foundational 3D reconstruction models to meet the complexity of modern design tasks. (3) The demonstration of metaheuristics as viable tools to parametrically build training datasets of optimal structures for predictive design tasks. (4) An unprecedented improvement in prediction capacity, resolution, and inference speed for sensor-based asset monitoring systems, driven by the introduction of hypernetworks within the field of structural health monitoring.

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1. Summary

In the coming years, the fields of architecture, engineering, and construction (AEC) have the capacity to experience a profound transformation, driven by newfound demands for sustainable industry practices, productivity, and innovation. Traditional design and maintenance approaches have often proved themselves to be insufficient and uncompetitive within the context of these paradigm shifts, resulting in lagging rates of productivity throughout each phase of the asset lifecycle. While sectors such as manufacturing, agriculture, and finance have benefitted from decades of modernisation, productivity throughout our built environment has not followed a similar uniform trajectory. This existing period of stagnation provides opportunities for disruptive technologies to emerge that radically redefine conventional practices. This has been reflected in recent years through a growing academic interest in novel computational tools and workflows that aim to improve design and operational procedures. Within this context, methods such as metaheuristics and gradient-based algorithms have provided researchers with a range of comprehensive rules-based tools to navigate vast, complex, and high dimensional solution spaces. These algorithms offer efficient, reliable, agnostic, and adaptive strategies that are highly suited to address the complex engineering problems that industry practitioners face. Similarly, modern machine learning models offer the potential to enable unprecedented capabilities for pattern recognition, structural optimisation, generative design, and predictive analysis tasks. While academics have examined these distinct fields of research across experimental projects and isolated industry tasks, their broader adoption throughout real-world design and maintenance applications remains to be seen at scale. Challenges such as limited access to high-quality training data, expensive hardware for training and inference, requirements for extensive multidisciplinary knowledge, performance sensitivity to hyperparameters, a lack of output interpretability, and a conservative risk-averse institutional culture remain significant challenges that hinder the introduction of these potentially transformative technologies.

This thesis posits that the complementary opportunities that metaheuristics and machine learning offer one another remain largely unexplored within the context of our built environment. While metaheuristic algorithms and deep learning models have inspired extensive bodies of research within the fields of structural design and structural monitoring respectively, they have remained largely independent of one another within the broader academic field of civil engineering research. Likewise, metaheuristic algorithms and machine learning models have found distinct and largely disparate applications throughout each phase of the structural lifecycle. Addressing these research gaps has remarkable potential to transform existing practices and eliminate barriers that have historically impeded technological adoption throughout these unique phases. Consequently, this thesis has taken a holistic approach to integrating these computational methods, developing hybrid solutions that

overcome significant limitations across engineering optimisation, predictive design, and structural health monitoring tasks.

1.2. Metaheuristics and machine learning within the built environment

The terms 'Meta' and 'Heuristic' respectively refer to a high-level optimisation strategy and a practical rules-based approach to search space exploration. As black-box algorithms that were developed to solve complex and high-dimensional optimisation problems, metaheuristics have become widely adopted by academics researching our built environment. Unique research applications include structural optimisation [1], schematic design [2], building energy optimisation [3], thermal performance [4], construction planning [5], urban design [6], and layout optimisation for sensor-based structural health monitoring [7]. While there has been a broad range of demonstrated use-cases for metaheuristics across design, construction, and maintenance related tasks, research papers and real-world industry applications of metaheuristics typically focus on earlier design phases of the asset lifecycle. This is primarily due to the utility of metaheuristic algorithms as high-performing and flexible solvers that can be easily adapted and conceptually aligned with traditional engineering design tasks. By modifying the decision variables and encoded constraints, algorithms such as differential evolution or particle swarm optimisation may be translated from one structural parameter set to another — emphasising the adaptability of a metaheuristic approach across a broad range of design tasks. Likewise, by selectively changing the objective function, the optimisation strategy may be transformed from a focus on cost optimisation to a reduction in material expenditure. For design tasks in which multiple competing outcomes must be balanced, well-established multi-objective variations have also been developed for many leading metaheuristic algorithms, such as the non-dominated sorting genetic algorithm [8], multi-objective particle swarm optimisation [9], and multi-objective differential evolution [10].

While metaheuristics have become deeply rooted throughout the academic field of design optimisation, machine learning models are particularly well-suited for processing the vast quantities of data that are captured over the monitored service life of built assets. Structural health monitoring (SHM) has evolved significantly over the past decades, transitioning from an overreliance on inefficient forms of non-destructive evaluation to applied sensor technologies. These sensors are typically installed during or post-construction, capturing enormous quantities of structural response data such as displacement, strain, temperature, vibrations, and environmental changes. Through this transition, the field of structural health monitoring has inherited a 'Big-Data' problem that presents itself as a valuable research opportunity for machine learning researchers. Academics have struggled to parse these vast structural metric datasets effectively and provide practical insights for industry practitioners. Consequently, there has been an increased demand for researchers to develop predictive models that process these sensor datasets to assess and localise damage, forecast performance, and interpolate structural response. Fundamentally, this field of research aims to provide methods to detect structural deterioration as soon

as possible, optimise maintenance schedules, reduce the downtime of critical infrastructure, improve the serviceability of assets, and ensure the safety and reliability of structures throughout the built environment.

This thesis explores a unique synthesis of metaheuristic algorithms and modern machine learning modelling within the built environment. It proposes that there are symbiotic opportunities at the intersection of these computational methodologies that have been largely unexplored, which have the capacity to disrupt existing boundaries for fields ranging from engineering design to structural maintenance and monitoring. These opportunities are summarised in Figure 1.1. While the field of metaheuristics has established itself throughout the academic field of structural optimisation, the potential for machine learning models to enrich the design process and overcome inherent algorithmic limitations is yet to be fully realised. Likewise, pervasive challenges that significantly impact the training and inference of machine learning models such as the generation of labelled datasets and the initialisation of performance sensitive hyperparameters may be overcome by selectively introducing metaheuristics. From conceptual design and optimisation to real-time structural health monitoring and predictive analysis, this thesis aims to demonstrate novel data-driven computational solutions that utilise these complementary features. Each research scope investigated throughout this thesis addresses a range of distinct challenges by developing hybrid solutions at the intersection of metaheuristic algorithms and machine learning.

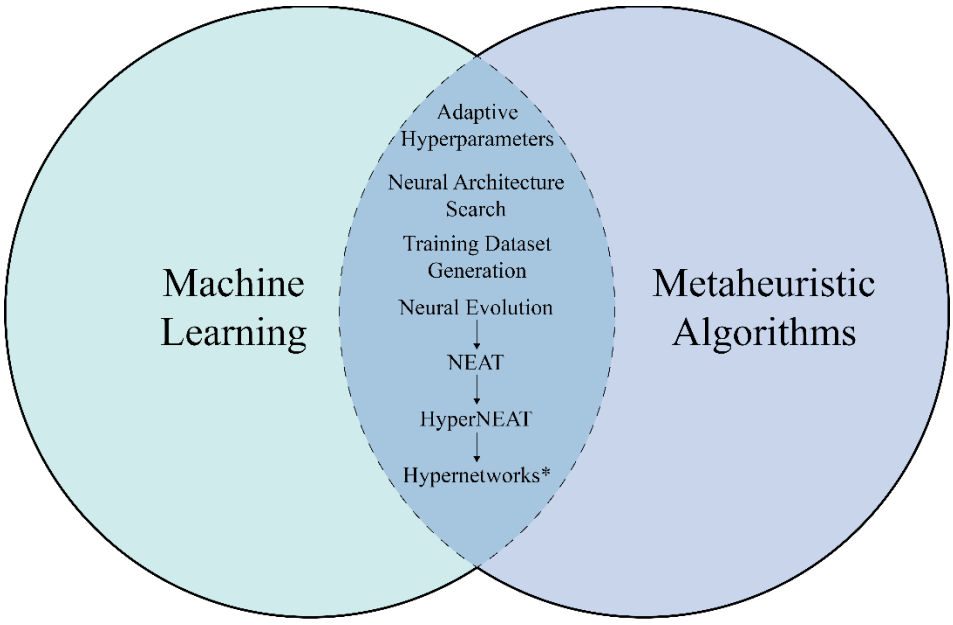


Figure 1.1 – Research scopes at the intersection of metaheuristics and machine learning

1.3. Motivation and Research gaps

1.3.1. Lagging productivity

Despite rapid technological advances throughout academia and applied engineering industries, the built environment lifecycle is increasingly burdened by a lack of automation, data silos, disjointed workflows, and uninformed decision making. From preliminary design to long-term asset maintenance, these limitations have contributed to a lagging productivity growth rate within the construction industry relative to other sectors. Aggregate productivity data from the U.S Bureau of Economic analysis paints a confronting picture: while the broader U.S economy has enjoyed stable growth in productivity in terms of output per unit of labor or capital over the past 50 years, the construction sector has demonstrated a trend of negative productivity growth. Between 1970 and 2020, value added per worker has decreased by approximately 40%, constituting a decades long negative trend in total factor productivity (TFP) [11]. During a period in which U.S sectors such as manufacturing and agriculture experienced average annual compounding growth rates of 2.6% [12] and 1.49% [13] respectively, the construction sector has been subject to a well-documented declining trend in labor productivity [14]. With the construction sector building structures worth approximately $2.1 trillion every year and employing over 8 million people in the United States [15], the industries lagging productivity has also had a very real impact on the health and growth of the countries broader economy. Researchers have argued that prior to the 2008 global financial crisis, financial markets failed to assess this trending deterioration in labour productivity and the significance of the construction sector as an economic barometer [16].

A similar trend can be observed from historic data provided by the Australian Government's Productivity Commission, which revealed a 12% decrease in labour productivity over the past 30 years, during which the broader Australian economy increased by 49%. When measured in terms of housing construction productivity, the number of dwellings that have been constructed per hour worked has also decreased by approximately 53% between 1994 and 2023 [17]. In an age of radical technological transformation and innovation, the lagging productivity of the construction sector is an anomaly amongst industries that have otherwise flourished. The commission continues to argue that had the productivity of the broader economy followed the same trend as the construction sector, the average income of Australians would be 41% lower. Further studies into the economic implications of construction productivity have also argued that even a 1% productivity improvement within the sector would yield a national benefit of $1.252 billion AUD [18].

One of the key issues that the Australian Government's Productivity Commission identifies as a potential root cause for this performance lag is a pervasive lack of innovation and technological adoption throughout the built environment. According to the commission, only 35% of construction firms actively integrate innovative technologies into their existing workflows. In comparison, a survey by the Australian Bureau of Agricultural and Resource Economics and Sciences (ABARES) found that within the agricultural sector, approximately 96%

of Australian farms currently own and implement Information and Communication Technologies (ICT). These include the widespread adoption of new technologies such as GPS guided harvesting equipment and other digital tools which have radically altered farming operations [19]. Similarly, growth within the Australian manufacturing sector has also been driven by an underlying culture of innovation through digital transformation. A report published by the Australian Industry Group identifies that as of October 2024, 84% of surveyed businesses are actively adopting technological upgrades and 52% of businesses are already reporting the integration of AI tools as a means of improving workflow efficiencies and driving productivity growth [20].

The remarkable absence of an innovation culture within the construction industry may be argued to be a failure of academia to adequately capture the needs of industry practitioners and mobilise technological adoption. In recent years, the academic field of sensor-bases structural health monitoring has promised radical transformation through the integration of sensor technologies that are capable of recording the structural response of assets in real-time. These sensors capture enormous quantities of complex structural data, which — within the context of applied structural health monitoring — are extremely difficult for asset managers to extract value from. Despite meaningful advances within the field itself, the current era of academic structural health monitoring research has insufficiently captured the needs of modern infrastructure owners and operators to effectively utilise these new sensors. Instead of tackling this 'Big Data' challenge that is currently inhibiting broader industry adoption of these new innovative technologies, academia has instead restricted itself to low-dimensional forms of analysis that fail to adequately represent the complexity of the underlying problem.

SHM has emerged within this context as an academic field that is preoccupied with technological capabilities rather than addressing the fundamental challenges that these technologies present. Existing publications within this field are almost entirely limited to two-dimensional plots and time series reporting, providing no geometric context for the structural analysis with respect to the monitored asset itself. Within a field that has historically relied entirely on thorough visual inspection of individual structural elements, it is perhaps not surprising that low-dimensional visualisations have not motivated widespread investment from asset owners. At an industry scale, infrastructure owners and operators remain largely unconvinced that the upfront cost of these sensor systems can be reconciled by slow, low-resolution solutions that produce vast quantities of unused data. Despite meaningful advances within the field itself, the current era of SHM has insufficiently captured the needs of modern infrastructure owners and operators.

The lagging productivity of industries surrounding our built environment has, in some ways, become one of their defining characteristics when observing the evolution of modern economies. This thesis is deeply motivated by the need to mobilise the integration of innovative technologies throughout the built environment in order to reverse the declining rate of productivity and improve decision-making across the asset lifecycle. It

asserts that entirely novel solutions are required to overcome the limitations that restrict existing academic research. By examining the broader asset lifecycle rather than isolating to an individual phase, this thesis aims to cultivate a culture of optimism towards technological innovation. From structural design to smart asset monitoring, academia must provide methods to scale predictive methods to meet the complexity of modern engineering tasks, not shy away from them.

1.3.2. The crisis of aging infrastructure

Our built environment currently faces a potentially catastrophic crisis. Assets are deteriorating at a global scale, with vast quantities of critical infrastructure reaching the end of their service life. Across the United States, 30% of the publicly listed 625,000 bridges have been classified by the Department of Transportation as structurally deficient or functionally obsolete [21]. In France, Germany, and the United Kingdom, deficiencies within highway bridges have been recorded at rates of 39%, 30%, and 37% respectively [22]. Closer to home, Dr Colin Caprani from Monash University argues that over 70% of Australian bridges are over 50 years old and beyond their service life, predicting that by the year 2035 an estimated 30% of Australian bridges will be structurally deficient [23]. These damming statistics underscore the importance of proactive investment in innovative monitoring strategies that have the potential to transform the field of SHM and meet the rapidly increasing demands of our deteriorating built environment.

The large-scale aging of civil infrastructure significantly increases the risk of catastrophic structural failure. The 2022 Fern Hollow bridge collapse in Pittsburgh was directly attributed to years of neglected maintenance and unaddressed corrosion damage, emphasising devastating failures in the city's maintenance procedures. In 2024, the National Transportation Safety Board [24] released a full report that blamed the city and state inspection procedures for the deterioration, inadequate repairs, and eventual structural collapse, opening up lawsuits for injured parties to sue the city of Pittsburgh. Despite being rated in "poor" condition for over a decade, the bridge remained in service without weight limits or an effective remediation strategy. This case study is unfortunately not an exception — It is a product of our existing approach towards maintaining civil assets. At a global scale, critical infrastructure towards the end of its service life is being fully utilised without a widely accepted strategy to comprehensively monitor and proactively repair these structures.

The increased risks of environmental disaster and extreme weather events due to climate change only exacerbates the ongoing deterioration of existing infrastructure [25]. Severe weather events that were previously anticipated to occur once every few centuries are now forecast to occur with significantly increased frequency. In 2020, extreme rainfall caused the aging Michigan's Edenville Dam to collapse, forcing over 10,000 people to evacuate and resulting in property damages that surpassed $250 million [26]. In 2023, two dams in Libya experienced catastrophic structural failure due to a record-breaking storm, with extreme rainfall causing

overtopping which eroded the downstream dam face. The Derna dam collapses are estimated to have killed up to 24,000 people, recorded as the second deadliest dam failure in world history. In the aftermath, 12 officials were charged due to mismanagement and maintenance negligence, with some facing up to 27 years in prison [27].

Despite the looming risk of global infrastructure disruptions, investment has lagged significantly behind rising SHM demands. The Global Infrastructure Hub forecasts that by the year 2040, the gap between infrastructure spending and the estimated cost of adequately maintaining and developing global infrastructure will reach $15 trillion [28]. The American Society of Civil Engineers (ASCE) published a revealing "Failure to Act" report that detailed the devastating impact of neglected infrastructure on the US economy, with reduced infrastructure serviceability, decreased productivity, and increased costs for households and businesses—resulting in an estimated loss of over $10.3 trillion in GDP by the year 2039 [29]. Research has demonstrated that the upfront cost of modern SHM systems is often regarded by asset owners as economically unviable in comparison to low-cost periodic visual inspections. In practice, inadequate investment in effective structural health monitoring directly increases total lifecycle costs due to a failure to proactively detect structural damage, inefficient maintenance schedules, increased risk of catastrophic failure, reduced operational lifespan, emergency repairs, and the resulting downtime of critical infrastructure [30].

1.3.3. Limitations of existing deep learning-based SHM systems

The prominent 'Big data' problem that has emerged within the field of structural health monitoring has driven increased research interest in deep learning (DL) based solutions for a vast range of assets, including bridges, buildings, and pipelines. However, a predictive model has not yet been developed that leverages these enormous datasets to comprehensively model the entire response of a monitored structure. Previous research has explored the viability of using sensor datasets to train convolutional neural networks to classify damage, however these networks are typically restricted to a small range of condition classes and do little to improve the precision of low-dimensional downstream SHM analysis [31, 32]. Likewise, sensor driven models that have been developed for the prediction of structural metrics such as displacement or strain typically do not provide any geometric context for the predictive output — relying almost entirely on two-dimensional time-series plots which are limited to discrete sensor locations [33, 34]. These overwhelmingly common forms of academic SHM analysis fail to adequately capture the entire spectrum of structural behavior that the monitored asset is subject to, directly limiting the interpretability and value that these solutions provide to asset managers and engineers.

This lack of structural-level inference analysis has resulted in many modern machine learning SHM solutions falling short of structural diagnostic or holistic analysis. High-level indicators such as binary damage detection [35], region-based structural response predictions [36], and two-dimensional visualisation [37] ignore the

opportunity for element-level structural analysis, structural simulation, or large scale high-precision predictive monitoring. The three-dimensional analysis and context that these solutions ignore is particularly challenging when considering the fact that these tools are being developed to reduce the existing industry's overreliance on conventional forms of visual inspection. While significant advances have been made in machine learning-based structural health monitoring, existing approaches do not provide comprehensive, memory-efficient, high-resolution, and spatially rich modelling capabilities for sensor-driven, real-time predictive asset monitoring. By closely aligning the predictive output of the proposed SHM solution with the practical strategies currently utilised by industry practitioners, this thesis aims to further encourage the broader adoption of sensor technologies and improve established asset maintenance procedures.

1.3.4. Challenges facing algorithmic adoption

The design considerations expected from modern structural engineers and architects require a holistic appreciation for buildability, construction, maintenance, material expenditure, and sustainable practices. The focus of design has broadened to encompass an understanding of an asset's entire lifecycle, from discrete elements of the supply chain to the impact of our built environment on broader society. This increased responsibility brings with it a necessary critical analysis of the existing methodologies and tools that designers rely on to meet the changing needs of our communities and infrastructure. Traditional structural design practices throughout the construction industry rely on a point-based optimisation approached that has an inherent reliance on trial and error, as well as the subjective intuition and experience of the engineer tasked with the project. In practice, this often results in inefficient designs and redundant material expenditure, with little consideration for the multifaceted spectrum of design that is required from modern engineers. To meaningfully satisfy these new-found expectations, the construction industry must integrate modern design methods which allow for the autonomous exploration and hierarchical ranking of various potential design solutions and alternatives. The ability to effectively quantify the quality of proposed designs is essential not only for the process of structural optimisation, but also for the broader ideological goal of empowering designers and structural engineers towards productive workflows.

Despite these increased demands for improved design workflows, the widespread adoption of innovative tools from the academic field algorithmic optimisation remains to be seen. From a metaheuristic perspective, it may be argued that this is partially due to the sensitivity of these algorithms to the initialised sets of hyperparameters that guide the exploration-exploitation response. The appropriate selection of metaheuristic subfunctions such as the fitness, mutation, crossover, selection, and population generation is essential for optimal algorithmic performance. Consequently, applying these algorithms effectively typically requires extensive experimentation, experience, and practical knowledge from industry practitioners. Practical implementations of conventional metaheuristic algorithms task industry practitioners with selecting appropriate algorithmic parameters, design variables, and objective functions in order to control the domain, reduce search space complexities, mediate the

exploration-exploitation trade-off, and promote efficient search space traversal. The prerequisite algorithmic experience required to effectively apply these methods remains a pervasive challenge that has inhibited the growth of metaheuristics beyond academic contexts.

1.3.5. Design complexity and resolution capacity

The integration of predictive 3D modelling tools into existing architectural and engineering design workflows has significant potential to transform decision making processes, optimise performance, and enhance the efficiency of industry practitioners. For this to be achieved, innovative solutions must be proposed to overcome the significant limitations that have come to define the academic field of 3D reconstruction. Object-based opensource datasets such as ShapeNet and PartNet have become widely used throughout the available literature, with limited scene complexity across published papers. Training samples within these benchmark datasets are self-contained objects such as bookshelves, mugs, and furniture, rather than components of a larger, interconnected geometric environment. Importantly, these object databases are not representative of applied design tasks, which require the construction of complex structures that may contain hundreds or even thousands of interconnected elements. Consequently, there currently exists a vast research gap that demands new methods of scaling these foundational models for complex real-world applications. Throughout the available literature, there are currently no proposed solutions that enable researchers to scale these foundational 3D reconstruction techniques to meet the practical needs of designers and engineers.

This challenge is compounded by the resolution capacity of prominent voxel-based 3D reconstruction models, which are largely inadequate when evaluated within a practical design context. Voxel models are conventionally limited to fixed sampling grid resolutions due to the memory requirements of high-resolution grid datasets and the popularity of convolutional architectures, typically implementing grid sizes of 32^3 , 64^3, or 128^3. The cubic relationship between grid resolution and memory usage has drastically impeded the practicality of high-resolution voxel models, which inherit computationally expensive training requirements and limited inference applications. As a result, the academic field of predictive 3D modelling has struggled to train foundational machine learning models that are capable of solving practical design tasks. Despite significant potential to revolutionise design practices, discrepancies between predictive resolution capacity and the complexity of designed structures have remained largely unanswered.

1.4. Research questions and objectives

The limitations that have shaped previous research scopes across the fields of algorithmic design and predictive structural monitoring have impeded the introduction of technologies such as metaheuristics and machine learning models throughout the full asset lifecycle. These challenges have in turn had a profound impact on the stagnant growth of the AEC and SHM sectors, which have been burdened by a culture of low technological

adoption and an overreliance on traditional methods that are becoming increasingly antiquated. While rich and deep subfields of academia explore these individual computational opportunities in isolation, there exists a pervasive disconnect between compelling outcomes in academia and the lagging rates of productivity throughout the built environment. This argument is particularly poignant when taken within the context of the radical technological transformations that have been experienced by other sectors of the broader economy, such as manufacturing, agriculture, transportation, and logistics. From early-stage preliminary design to long term structural maintenance, this thesis aims to provide novel solutions that bridge the gap between academia and industry by addressing targeted research questions, establishing clear objectives, and pushing boundaries that have limited the existing scope of academic research.

To guide this research, the following key questions are posed across various stages of the asset lifecycle:

1. How can metaheuristics and machine learning models be implemented in a hybrid computational workflow to improve algorithmic design processes?
2. What additional design opportunities emerge by further developing novel models at the intersection of metaheuristics and deep learning?
3. How can hypernetworks — directly inspired by the evolutionary algorithm HyperNEAT — empower preliminary design practices for industry practitioners?
4. What methods can be proposed to scale 3D reconstruction techniques to bridge the gap between established academic research tasks and the complexity of real-world engineering and architectural design tasks?
5. What are the critical limitations that have inhibited the broader adoption of technological innovation across modern structural health monitoring workflows?
6. How can hypernetworks be used to address the pervasive ‘Big Data’ problem that has been inherited within the field of sensor-based structural health monitoring research?
7. What are the potential improvements — in terms of predicted structural response resolution, inference speed, and model efficiency — that can be made by introducing hypernetwork architectures into the field of structural health monitoring?

1.5. Research scope

The fields of structural design and structural health monitoring have demonstrated compelling yet isolated applications for metaheuristics and deep learning models, respectively. Research scopes rarely intersect across these domains, which inherently limits the tools available to researchers and industry practitioners. Within this context, metaheuristics are typically restricted to the field of algorithmic structural engineering, providing a significant gap within the available literature to explore potential applications of these methods to supplement pervasive ML modelling challenges. Issues relating to dataset generation, hyperparameter optimisation, and neural architecture are well suited scopes for interdisciplinary research across metaheuristic and machine learning algorithms. Likewise, the potential for cutting edge ML techniques to augment traditional metaheuristic

approaches and overcome existing limitations within the field has not been fully realised. These hybrid methodologies offer significant potential to radically improve existing approaches, empower industry practitioners, and mobilise technological innovation throughout applied facets of our built environment.

The differential evolution algorithm has been widely explored throughout this thesis, largely due to its competitive performance throughout the available literature, the results from the first two benchmark research scopes, its intuitive hyperparameters, and the flexibility it offers through clearly defined mutation functions. Algorithmic benchmarking was a key focus within Chapter 4, given the relevance of the research within the field of algorithmic structural design. Consequently, alternative algorithms such as particle swarm optimisation and the genetic algorithm were also examined during this stage of the thesis. As the research scope evolved to encompass additional phases of the asset lifecycle, DE was adopted as the chosen metaheuristic approach, allowing the research scope of later papers to extend beyond the context of algorithmic benchmarking.

The solutions proposed throughout this thesis are relatively radical in their approach, forfeiting incremental improvements in established methodologies for disruptive and unique tools. This thesis argues that the impermeable barriers that have inhibited technological growth throughout the fields of structural optimisation, predictive design, and structural health monitoring are unlikely to be overcome by gradual refinements of well-established methods. The over-saturation of low-dimensional analysis throughout the academic field of SHM has done little to mobilise the broader adoption of smart-sensors throughout the built environment. Likewise, the dominant use of single-object geometry datasets for academic 3D reconstruction tasks has ignored the true complexity of real-world design applications. Innovation within academic fields often follows an asymptotic trajectory in which the rate of truly novel contributions decreases over time, approaching an ‘innovation plateau’ beyond which performance improvements are marginal and insurmountable research barriers become recognised. These periods provide profound opportunities for radical solutions to disrupt this asymptotic pattern, redefine established methodologies, and advance research beyond these widely accepted barriers.

1.6. Summary of contributions

Importantly, the scope of this thesis has been defined by the unique opportunities and gaps at the intersection of metaheuristic algorithms and machine learning research, with a focus on the broad industry implications that these novel solutions present throughout the built environment. By framing the research through this highly specialised lens, this thesis has introduced entirely novel solutions to the field of structural engineering, preliminary iterative design, and structural health monitoring. Across these domains, the new interdisciplinary models proposed in this thesis have achieved remarkable performance improvements and, in some cases, have introduced unprecedented capabilities that push the bounds of these academic fields into new and fertile territory.

The first research contribution revealed through this thesis includes the automated control of critical hyperparameters using neural networks, adapting algorithmic behaviour during the optimisation process. For the proposed algorithm, differential evolution was implemented with a deep fully-connected neural network controlling the mutation function between explorative and exploitative approaches. By modifying the mutation of the differential evolution across each generation of the optimisation process, search space navigation and population diversity was dynamically adapted to maximise convergence performance. The implemented neural network developed this strategy through proximal policy optimisation — a cutting-edge reinforcement learning technique that strikes a balance between network stability and learning efficiency using a clipping policy objective. This novel approach effectively reduces the performance dependency of differential evolution on key algorithmic hyperparameters, which typically limit the viability of metaheuristics within an applied context. By reducing the highly technical knowledge and experience required for industry practitioners to effectively deploy these algorithms, this research aims to mobilise the use of metaheuristics for applied engineering design tasks.

The consistent performance improvements achieved through this hybrid approach inspired the development of the MEMODE algorithm, standing for memory efficient multimodal optimisation with differential evolution. This algorithm reverses this relationship between metaheuristics and machine learning to explore the viability of using differential evolution to optimise the architecture and hyperparameters of complex, memory efficient multimodal models. The manual tuning and architectural design of machine learning models is typically a time-intensive and computationally inefficient process, which is unlikely to produce models that are proximate to optimal performance. This challenge is compounded for complex multimodal models, which require the careful design of multiple input data branches and downstream data handling. This thesis presents the first application of a metaheuristic algorithm to automate the design of memory efficient multimodal neural networks, which have significant potential applications across tasks relating to architectural and engineering design, 2D drawing reviews, infrastructure monitoring, and predictive asset maintenance. As alternative implementations were also benchmarked within this paper, MEMODE also demonstrates the compelling performance of differential evolution when compared against competing approaches such as particle swarm optimisation, the genetic algorithm, and Bayesian optimisation.

While the academic fields of metaheuristics and machine learning have established themselves across a vast array of practical applications, they have largely remained as distinct and separate methodologies with limited interdisciplinary research between them. Notable exceptions to this assertion include the remarkable Neural Evolution from Augmented Topologies algorithm (NEAT) and HyperNEAT variations, which came to inspire the creation of the hypernetwork method by David Ha in 2016. As a novel meta-modelling approach that evolved from the limited research published at the intersection of these two fields, hypernetworks have also been examined extensively within this thesis — offering profound research opportunities that range from preliminary

design modelling to predictive structural health monitoring. Hypernetworks were first introduced within this thesis within the context of structural design, reconstructing 2D and 3D designs using structural datasets that were parametrically generated using differential evolution. This chapter illustrated the powerful representation capacity of hypernetworks for interactive 'text-to-structure' design and the viability of using metaheuristics for training database development. These contributions were extended in the following paper through the introduction of transformer-predicted translation vectors, which are used to scale foundational reconstruction models to build structural designs with unprecedented output complexity.

The hypernetwork framework developed through these design-based chapters was then redesigned to radically overcome some of the most severe challenges faced within the field of sensor-based structural health monitoring. Namely, the academic field of SHM has become burdened by a 'Big data' problem, collecting vast quantities of sensor data that are extremely difficult to interpret or utilise. As a result, modern sensor-based SHM solutions have largely stuck to low-dimensional forms of structural analysis, ignoring the geometric context that underpins conventional approaches to element condition rating and visual inspection. This thesis argues that these critical limitations have impeded the broader adoption of these technologies throughout the applied field of structural health monitoring, reflecting a deeper void between academics and practitioners in terms of practical outcomes. In contrast, hypernetworks offer a high-dimensional implicit neural representation of structural performance, which can be sampled with theoretically infinite resolution. These models allow for the structural response at any point within a monitored structure to be evaluated according to the sensor-based metrics recorded within the underlying training dataset. While strain-based monitoring was selected within this research project due to the compelling performance of fibre Bragg grating sensors for monitoring strain, the proposed hypernetwork framework is broadly applicable for the prediction of alternative structural metrics such as deflection and stress. Additionally, the hypernetwork models developed within the context of this thesis are extremely efficient in terms of their memory requirements and inference speed, allowing for their application across remote devices for real-time predictive monitoring tasks. The results from this study reveal the vast applications of hypernetworks within the context of sensor-based predictive asset monitoring, with potential future applications throughout the broader field of structural health monitoring itself.

1.7. Thesis structure

This thesis contains six distinct research scopes that each examine hybrid computational approaches and their applications throughout a diverse range of asset lifecycle phases. These phases range from structural design optimisation and interactive preliminary design iteration to predictive structural health monitoring over the service life of constructed assets. While the isolated asset lifecycle phase varies throughout the chapters of this thesis, the novel algorithms and deep learning models developed in each paper are deeply inspired by an expanded interdisciplinary study of metaheuristics and cutting-edge machine learning models. This thesis has been ordered according to the iterative evolution of these novel approaches over sequential research projects,

with each research scope building off the contributions from previous chapters. Consequently, the first research chapter examines the complementary relationship between metaheuristics and reinforcement learning for dynamic algorithmic behavior during the optimisation of steel frame structures. This relationship is inversed within the scope of the following chapter, which explored the potential for differential evolution to generate highly optimised multimodal model architectures. Both of these sections may be grouped within the initial algorithmic benchmarking scope within the broader structure of this thesis.

Chapter 4 explores the opportunity of using deep neural networks to mediate the exploration-exploitation trade-off for metaheuristic algorithms, applied within the context of conventional structural optimisation benchmark tasks. This includes a benchmark analysis of the differential evolution algorithm, with an adaptive exploration-exploitation response. The findings of this preliminary study proved to be significantly influential in the ongoing application of the differential evolution algorithm throughout this thesis. This relationship between metaheuristics and deep learning is inverted in the following section, which investigates the performance of multimodal models to generate architectures and hyperparameter sets using differential evolution. This is applied within the context of memory-efficient multimodal model design, in which multiple complex input streams were reconciled into a single fully connected branch for classification. The final model evolved through the proposed algorithm outperformed over 1,600 competing models, including competitive ResNet and Inception variations.

Chapter 5 investigates the representation capabilities of hypernetwork models within the context of predictive structural design. An interactive ‘text-to-structure’ approach is proposed, in which a transformer-based named entity recognition model was trained to extract design criteria from a user’s input command. These parameters were normalised before being passed as inputs to trained hypernetworks, generating structures that range from 2D cantilevered trusses to 3D gridshells and bridge systems. These structural datasets were generated through a novel application of metaheuristic algorithms, overcoming data limitations that have restricted the field of predictive 3D reconstruction. This concept is scaled to meet the complexity of real-world design tasks by introducing the concept of a predicted ‘translation vector’. These vectors manipulate predicted meshes throughout the design space, enabling the creation of complex structural systems with hundreds or even thousands of interconnected elements. This novel methodology allowed for the predictive construction of steel frame structures, multistorey high-rises, and long-span tied arch bridges.

Chapter 6 transitions from a design focus to explore the application of deep neural networks for asset maintenance, driven by embedded sensor technologies. A series of 64 Fibre Bragg grating sensors were installed along the longitudinal rebar of a footbridge prior to construction, capturing the strain response of the structure due to mechanical effects. The strain recordings from a loading experiment were then used to train a deep neural

network, the output from which was used to generate a 3D heatmap visualisation of the structural response. This framework is built upon by introducing the developed hypernetwork models into the field of structural health monitoring. This chapter proposes a new frontier for the field of structural health monitoring, which has historically been hindered by low-dimensional structural analysis and limited real-time capabilities. The proposed hypernetwork approach enables unprecedented structural analysis capabilities, with theoretically unlimited visualisation resolution, rapid inference speeds, and remarkable model accuracy for strain-based predictions. Within a practical setting, these novel predictive monitoring capabilities could facilitate the proactive identification of structural deterioration and failure, enabling remedial works to be undertaken prior to serviceability disruptions or structural failure — marking a transition from reactive to proactive asset maintenance.

Chapter 7 concludes with an extended discussion on the broader implications of this research for academics and practitioners throughout the built environment. This includes the improvements demonstrated for algorithmic design, the scaling capacity provided to foundational 3D reconstruction models, and the new-found opportunities hypernetworks present to the field of structural health monitoring. Potential areas for future research are also examined within this chapter, providing recommendations to modify the proposed approaches across alternative promising applications. The progression of each research chapter from preliminary algorithmic design to long-term asset maintenance and structural health monitoring is outlined in Figure 1.2.

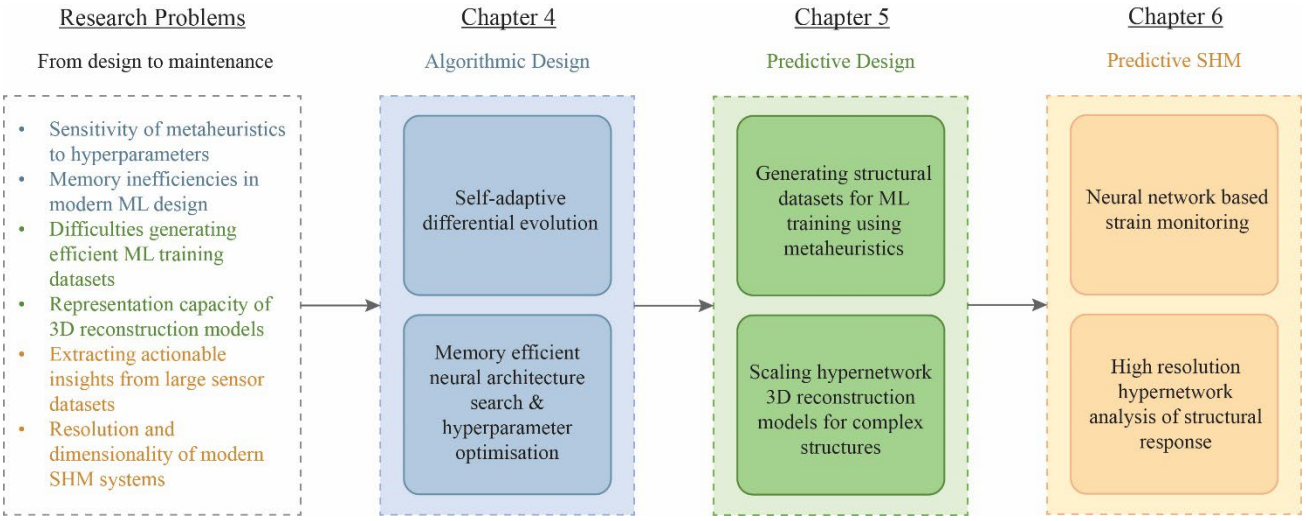


Figure 1.2 – Development of research scopes throughout thesis chapters

CHAPTER 2: LITERATURE REVIEW

2.1. Summary

This thesis argues that innovation – and the removal of established limitations that have inhibited growth throughout academia and industry – is often mobilised by new ideas that lie at the intersection of multiple fields of study. This literature review begins with a high-level introduction to the relevant disciplines throughout the built environment that have significant potential to benefit from an interdisciplinary application of metaheuristics and machine learning models. The limitations and opportunities that are present for each of these fields are then discussed through this interdisciplinary lens. From preliminary algorithm design and predictive structural engineering to machine learning asset monitoring systems, this thesis has taken a holistic approach to addressing critical challenges facing academics and industry practitioners when introducing these innovative technologies for real-world applications.

2.2. Structural health monitoring

The field of structural health monitoring (SHM) is primarily concerned with assessing the integrity and condition of existing structures throughout the built environment, such as bridges, buildings, tunnels, dams, and railway infrastructure. Historically, qualitative visual inspections have been one of the most dominant methods of evaluating the health of civil infrastructure, largely due to its low direct costs, simplicity, and independence from technologies that require additional technical skillsets. Unfortunately, these visual inspections are severely antiquated and are insufficient to address the rising challenges facing our built environment. The U.S Federal Highway administration reports that 95% of structural element condition ratings provided by state department inspectors varied between two rating points of the average, revealing the subjective nature of these visual condition ratings [38]. This issue of asset rating consistency is largely attributed to the individual experience of inspectors, variance in training procedures, and the challenges associated with accurately interpreting damages that are often not directly visible. Consequently, the field of SHM has been severely inhibited by subjective condition ratings, inefficient maintenance schedules, uninformed decision making, and an inability to evaluate hidden or internal structural damage. The combined effects of inadequate monitoring strategies, unaddressed structural deterioration, and the increased risk of extreme weather events due to climate change has the potential to cause devastating damage at a global scale.

Additional forms of non-destructive testing have also been introduced to evaluate the health of civil infrastructure, such as periodic ultrasonic testing, acoustic emission, thermography, and radiography. These techniques have been used with varying degrees of success to inspect assets such as pressure vessels, oil and gas pipelines, nuclear plants, and offshore platforms. However, these techniques all suffer from limitations that have far-reaching consequences for efficient asset monitoring. The intermittent nature of data collected using

non-destructive evaluation methods removes the capacity for real-time monitoring, resulting in a responsive approach to asset management rather than a proactive and preventive strategy. These tests require highly trained personal to undertake expensive, labour-intensive, and time-consuming scans. Consequently, the frequency of these checks is often inevitably reduced within practical applied contexts. Beyond these initial scans, many non-destructive monitoring techniques also require further interpretation from experienced professionals, introducing additional costs and variability due to subjective analysis. While developments in image-based monitoring and unmanned aerial vehicles have made considerable steps towards autonomous SHM, these methods are also highly sensitive to local environmental conditions, weather events, payload limitations, airspace regulatory restrictions, data processing bandwidth, and resolution capacity at scale.

As a result of the inefficiencies of periodic structural analysis and the subjectivity inherent to many non-destructive techniques, sensor-based monitoring strategies have become increasingly popular throughout academia. In contrast to traditional methods, sensors offer the opportunity for real-time monitoring, autonomous analysis, continuous long-term data, low ongoing labor costs, and immediate response capabilities in the event of catastrophic structural failure. The academic field of SHM has become dominated by innovative sensor technologies in recent years, tracking a diverse range of distinct structural behaviors. Accelerometers have been introduced to measure dynamic effects such as vibrations or seismic events throughout bridges, turbines, and buildings [39, 40]. For strain-based load monitoring, structural fatigue analysis, and the tracking of crack propagation, strain gauges and fibre optic sensors have become widely utilised [41-43]. In place of indirect methods of assessing structural changes, sensors such as linear variable differential transformers (LVDTs) and laser displacement sensors also have been deployed as deflection-based indicators of structural performance [44, 45].

One of the most compelling smart sensor technologies that has emerged in recent years within the field of SHM is the Fibre Bragg grating (FBG) sensor, which has demonstrated highly competitive monitoring capabilities in terms of scalability, precision, and environmental durability. These cables typically consist of a fibre core through which light is transmitted, the laser inscribed fibre Bragg grating which reflects targeted wavelengths of light, and a buffer coating which provides protection against mechanical effects. In contrast to conventional strain gauges, FBG sensors do not suffer from electromagnetic interference and are relatively durable for civil applications. Additional protective layers are also often used to shield against abrasions, moisture, and chemical effects. Consequently, FBG sensors may be directly embedded within composite structures during construction or retrofitted to existing structures as a robust and environmentally resilient sensor technology. Accumulatively, these characteristics have driven significant research interest in recent years into the use of FBG sensors as a cornerstone technology for modern structural health monitoring systems.

Fibre Bragg grating sensors are designed to measure the strain at discrete locations along a 5-7 mm diameter optical fibre, using a periodic modulation of the core's refractive index. This modulation is achieved by a manufactured grating that is inscribed within the core using a UV laser, which reflects targeted wavelengths of light referred to as the Bragg wavelength. The Bragg wavelength equation is defined as follows:

$$\lambda_B = 2n_{eff}\Lambda \tag{2.1}$$

where n_{eff} is taken as the effective refractive index of the optical cable and Λ represents the grating period manufactured within the cable itself. One of the major challenges in measuring strain using FBGs is their sensitivity to thermal effects, which can alter the manufactured grating period within the sensor itself and the corresponding wavelength of light that is reflected. Additionally, changes in temperature also influence the effective refractive index of the fibre core itself, directly impacting the final strain values calculated by the interrogator unit. As a result, applied SHM systems that utilise FBG sensors also require the installation of additional temperature sensors, which can capture these thermal variations across the monitored asset. The calculation of the final strain values at each discrete FBG location is highly sensitive to any mechanical and thermal effects. The impact of changing strain and temperature on the Bragg wavelength shift may be described using the following equation:

$$\frac{\Delta\lambda_B}{\lambda_B} = k * \left(\varepsilon_m + \alpha_{sp} * \Delta T\right) + \alpha_\delta * \Delta T \tag{2.2}$$

where $\Delta\lambda_B$ is the relative shift in the Bragg wavelength, λ_B is the original Bragg wavelength, k is taken as the photo-elastic constant of the fibre, ε_m is the mechanically induced strain, α_{sp} is the thermal expansion coefficient, α_δ is the change in refraction index, and ΔT is the change in temperature. By measuring the relative shift in the Bragg wavelength recorded by an interrogator unit and recording any changes in temperature with proximate temperature sensors, thermal effects may be compensated for, and the direct impact of mechanical effects may be isolated.

2.3. Algorithmic structural engineering

The field of computational structural optimisation first gained significant academic interest through a paper published by Professor Lucien Schmit in 1960, in which Schmit explored autonomous design processes by combining traditional finite element analysis with a numerical optimisation solver. In this research, Schmit identified various programming criteria that must be integrated within the proposed autonomous workflow, including the definition of structural design limitations, a chosen optimisation algorithm, and a criterion to evaluate and rank the success of generated solutions. Schmit demonstrated the design potential of this proposed framework through the optimisation of a three-bar truss system to minimise structural weight; a benchmark which has become widespread within this field of research. The value of this framework, as proposed by Schmit, lies not only within the structural analysis phase, but also through the automation of the preliminary design and subsequent redesigning cycles. Given this potential however, Schmit appropriately concluded that

computational optimisation should not be regarded as a substitute for creativity or fundamental analysis on the part of the structural engineer. Rather, this technology must serve as a supplementary tool to enrich the judgement, experience, and intuition of the engineer [46].

The value of structural optimisation research has historically been framed through financial incentives or potential improvements in structural resilience, with a range of successful applications bridging the gap between academia and industry. Structures such as wind turbines [47], prestressed concrete bridges [48, 49], composite beams and floor systems [50, 51], high-rise buildings [52, 53], reinforced concrete footings [54], and steel frames [55, 56] have all been used to demonstrate the possible cost efficiencies that metaheuristic algorithms can discover for real-world design tasks. By validating the application of various optimisation algorithms to improve the economic viability of designed structures, collaborative research projects have created significant value for practicing engineers and academics alike. This, in turn, has mobilised engineers towards the use of advanced structural optimisation tools.

As the design focus and priorities of engineers, designers and clients have shifted over time, project deliverables have come to have an increased focus on sustainable outcomes. This has been reflected within the field of structural optimisation through the introduction of new carbon-efficient optimisation techniques over the past few years, with contemporary projects ranging from multiobjective genetic algorithms capable of optimising project cost and reducing embodied carbon footprint [57], to novel generative tools that leverage thermal and lighting simulations to accommodate for passive design objectives [58]. The field of structural optimisation has grown to encompass a diverse range of applied solutions, which has developed alongside increased interest in environmentally considerate design throughout academia and industry. The construction industry's slow shift towards sustainability has highlighted the multifaceted relevance of this field of research, as a tool to empower engineers towards passive design, resourceful material expenditure, cost efficiency, ecological performance and an overall reduction in the carbon footprint of new projects.

The contemporary practice of algorithmic structural optimisation may be broadly divided into four categories, which each adopt a unique approach to fitness function design [59]. While there is often significant overlap in the shared outcomes of these optimisation objectives, these typical research scopes may be summarised as follows:

1. Reduction in project costs: conventionally, this is achieved through a reduction in the overall material expenditure for a given project, however research has also incorporated additional parameters within the optimisation function, such as constructability, scheduling factors, specific material prices and labour costs [60].

2. Improved environmental impact: the implementation of material factors such as embodied carbon, constructability, structural weight, energy expenditure, thermal performance, and overall material allocation within the objective function has been demonstrated as a quantifiable method to evaluate the sustainability of solutions generated through structural optimisation.
3. Increase in structural performance: through the use of structural analysis software or fundamental structural equations, the structural response of the generated designs can be evaluated for the desired loading conditions. By introducing this analysis into the autonomous design loop, structures may be optimised to satisfy the necessary performance criteria.
4. Multi-objective optimisation: the integration of multiple objective equations within the fitness function allows for designers to optimise and tune multiple factors at once according to their design needs, such as structural, economic, and environmental performance.

2.3.1. Algorithmic benchmarking

Truss design problems are perhaps one of the most common and widely used benchmark tests used throughout the academic field of algorithmic structural optimisation. This is primarily due to the ease of their implementation with various algorithms and the reduced computational cost for each optimisation iteration when compared against complex finite element techniques, allowing thousands of fitness function evaluations for benchmarking purposes. Truss optimisation problems provide an excellent framework to experiment and iterate through various algorithms and hyperparameter values within a lightweight and widely deployable benchmark environment. Conventionally, these comparative design tasks require the overall weight of a truss structure to be minimised while satisfying one or multiple structural design constraints. These often include an evaluation of the induced stress, deflection, and buckling relative to predetermined structural requirements. Within the context of truss optimisation, these constraints typically define the maximum allowable tensile stress (σ_{max}), compressive stress (σ_{min}), and deflection (δ_{max}) induced within each member of the structure. Nodal stresses and deflections calculated through either finite element simulations or the direct stiffness method [5] during the optimisation process must therefore satisfy the following design constraints:

$$\sigma_{i,min} \leq \sigma_i \leq \sigma_{i,max} \tag{2.3}$$

$$\delta_j \leq \delta_{j,max} \tag{2.4}$$

For these design constraints, the tensile and compressive stress of element i is defined as σ_i , while the nodal displacement at node j is represented as δ_j. In order to relate this work to preexisting literature within the structural optimisation community and allow for further standardisation in future research, all truss benchmark problems and specifications that are explored within this thesis have been borrowed from widely used published structural benchmarks [61].

2.3.2. Categorisation of optimisation algorithms

Algorithms throughout the field of structural optimisation are broadly divided into gradient based and gradient-free approaches. Notably, while gradient-based solvers perform well within the domain of topology optimisation, size optimisation, and shape optimisation [62], difficulties arise when gradient data is not directly derivable, inconsistent, or inefficient to compute. Structural design problems often require the consideration of critical design constraints that inhibit the application of conventional derivative methods such as the hill climbing algorithm. Within an applied industry context, design spaces are often high-dimensional, noisy, complex, and discrete — limiting the application of gradient-based methods for many practical engineering tasks. Due to the challenges associated with problem-specific derivative calculations and the unique requirements of structural optimisation tasks, non-gradient algorithms have emerged as a powerful and robust alternative to conventional gradient-based methods. Unlike gradient-based algorithms, metaheuristics do not require gradient information to solve a given optimisation task. Instead, they rely on a stochastic and probabilistic rules-based approach to navigate the search space and avoid premature convergence at suboptimal local minimums or maximums.

Metaheuristic optimisation algorithms may be broadly divided into population focused or trajectory focused operators, based on the defined search space exploration-exploitation strategy. Genetic algorithms and particle swarm optimisation algorithms are both examples of population-based processes, which generate a series of potential candidates that iteratively converge towards optimal solutions along different paths within the phase space. Population algorithms may be further subdivided into evolutionary based algorithms such as differential evolution and evolutionary programming, and nature inspired algorithms such as particle swarm optimisation and firefly optimisation. Evolutionary algorithms aim to mimic biological evolution through mechanisms such as mutation, selection, crossover, and reproduction to optimise populations of solutions towards high-performing genetic pools. Nature inspired algorithms are broadly driven by observations within nature and the social behaviour of animals, such as bioluminescent communication between fireflies or interactions between birds within a flock. In contrast, trajectory focused algorithms such as simulated annealing (SA) deploy a single solution within the phase space, which iteratively traverses along a specific trajectory with the goal of finding the most optimal solution for the given domain. These disparate techniques provide a broad variety of methodologies to navigate complex design landscapes and reliably converge at competitive solutions.

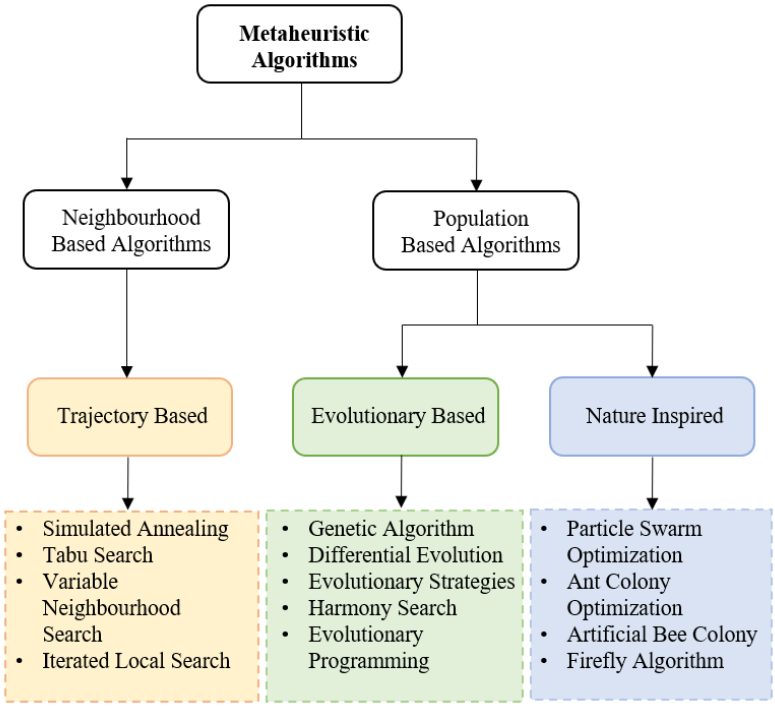


Figure 2.1 - Categorisation of Metaheuristic Algorithms

2.3.3. Fitness functions

At a fundamental level, metaheuristic optimisation aims to iteratively generate potential candidates that satisfy a provided fitness function. Metaheuristic algorithms provide a viable and competitive solution for optimisation problems in which the phase space is uniquely challenging to navigate, when gradient information is not directly available, or when the underlying fitness function inhibits the computational efficiency of state-of-the-art exact algorithms [63]. In their most generalised form, conventional optimisation problems may typically be described through the following equation:

$$\text{Minimise} \quad f(x_1, x_2, x_3, \ldots, x_n) \tag{2.5}$$

Subject to the following constraints:

$$h_j(x_1, \ldots, x_n) = c_j \, , j = (1, 2 \ldots, J) \tag{2.6}$$

$$g_k(x_1, \ldots, x_n) \leq b_K \, , k = (1, 2 \ldots, K) \tag{2.7}$$

where $f(x)$ represents the objective function which evaluates the fitness of generated solutions, x provides the design parameters, $h(x)$ is the equality constraint and $g(x)$ is the inequality constraint [64]. Structural design problems will often be subject to fixed design performance criteria such as deflection and stress limitations, in

addition to geometric constraints that control factors such as member sizing and element spacing. Fitness functions are used to evaluate the quality of generated solutions, with design variables being iteratively adjusted and tuned to maximise or minimise each solution's underlying fitness. For population-based metaheuristics, this is performed across generations by evaluating the performance of each solution within a population according to the chosen fitness function. Solutions that score well according to the chosen fitness function are retained and their unique collection of parameters is transferred to the next generation of potential solutions. Conversely, solutions with a lower fitness as evaluated by the objective function have a reduced probability of passing their parameters to the next generation of solutions. In this sense, the fitness function must be created in alignment with the objectives of the designer, as poorly designed fitness functions will often result in premature convergence or the generation of solutions that inappropriately address the design needs.

2.3.4. Search space exploration

When considering the process by which metaheuristic solvers iteratively improve their outputs, it is useful to consider the collection of generated solutions for the provided system within the context of a phase space. According to dynamical system theory, all possible configurations of a system may be represented as a unique point within the phase space. The dimensionality of the phase space will be directly tied to the complexity of the problem that the algorithm is attempting to solve. For example, if a metaheuristic algorithm is tasked with optimising a beam geometry in which the only changing variable is the length of the beam, the phase space will in turn be one-dimensional. In this example, solutions within the one-dimensional phase space only vary according to the generated beam length. However, if the coordinates of the beam supports to vary along the XY plane, the algorithm will now have five variables that it can leverage to optimise the model. In this example, the phase space that describes all of the potential solutions that exist for this problem will be five dimensional.

The hierarchical ranking and selection of states within the phase space is defined by a fitness function, which quantifies the desirability of a generated solution as a single number. The process by which an algorithm is programmed to navigate the search space for a given fitness function will define the quality of the final optimised solution. This evaluation requires the unification of the search space and fitness function into a single entity, which may be interpreted for two-dimensional problems by extruding each point within the search space. The magnitude of this extrusion is defined by the fitness of the given solution, creating a three-dimensional topography in which peaks represent high quality solutions and valleys represent inadequate solutions—assuming a maximising optimisation task. This relationship is flipped for optimisation problems in which the objective is to minimise the cost function [65]. It must be noted here that this fitness landscape is a conceptual representation of the process by which metaheuristic algorithms navigate the search space to converge on optimal solutions. In reality, the generation of the phase space topography would be far too computationally expensive to efficiently derive solutions for each generation. The practical implementation of metaheuristic

algorithms requires a small portion of the phase-space to be sampled in order to find optimal peaks within the fitness topography.

Given the limited sampling of the phase space, there is no guarantee that a given algorithm will be capable of producing the best solution for the provided domain. The goal of a stochastic solver should be to consistently generate high quality solutions which satisfy the prescribed design criteria and achieve convergence within a reasonable period of time. The appropriate selection of metaheuristic subfunctions such as the fitness, mutation, crossover, selection, and population generation is therefore essential in order to maximise performance. Notably, the number of generations that a given algorithm must process before converging is directly influenced by the quality of the fitness landscape topography. Typically, topographies which have predominantly flat or low-gradient portions are difficult for typical metaheuristics to efficiently navigate, as fitness improvements are minor across iterative generations. Similarly, terrain that is extremely rough or fractured may significantly inhibit the pathing of the algorithm and reduce convergence momentum [66]. As the quality of the fitness topography is defined by the fitness function, the effectiveness of the algorithm is highly sensitive to the hyperparameters defined prior to optimisation. It is the role of the individual constructing the solver to adequately select algorithmic parameters, design variables, and objective functions in order to control the domain, reduce redundant complexities, and promote efficient search space traversal [66]. The sensitivity of these algorithms to fixed hyperparameter sets is a pervasive challenge that has inhibited the broader adoption of metaheuristics within applied contexts.

2.3.5. Elitism

The stochastic nature of many metaheuristics can have a destructive effect on the performance of a given population by eliminating high performing solutions during the optimisation process. In order to prevent these potentially destructive forms of algorithmic exploration, top-k elitism is often implemented across optimisation steps to preserve the parameter sets of the best performing candidate solutions. These elite high performing solutions are not impacted by the explorative functions of the given metaheuristic such as mutation or crossover, succeeding to the next generation without modification. By preventing the loss of these individual designs and allowing their genetic information to be passed to sequential populations, convergence speed and search stability is typically improved. Notably, population diversity can suffer when elitism is overused, with the potential to cause premature convergence towards these historic local optimum solutions. Consequently, static elitism typically only retains the top one or two historic solutions, with variants such as dynamic elitism adapting the number of chosen elite candidates according to the current phase of the optimisation process. In this sense, elitism is a hyperparameter that can directly impact the exploration-exploitation response of a metaheuristic algorithm — requiring careful consideration with respect to the desired algorithmic behavior.

2.3.6. Penalty functions

To integrate constraints that are typically required in applied design tasks, a penalty function may be introduced to modify the fitness of infeasible or unsatisfactory outputs. Conventionally, this is defined as a coefficient within the fitness function which penalises the fitness of a given solution proportionally to the degree that the design criteria have not been met. The degree and application of penalisation has been the subject of extensive research over the past few decades, with the objective of establishing and evaluating a range of potential penalisation techniques which may improve algorithmic performance. There are two distinct categories of penalisation function which are typically used in the evaluation of a solution's fitness. These include the interior and exterior methods, which both seek to resolve the constraints that must be addressed for a given design task. Exterior penalisation is applied to solutions which are infeasible, with the objective of transitioning sequential generations of solutions towards the feasible region by punishing solutions that violate the design constraints. For minimisation problems, this requires an increase to a solution's fitness that is proportional to the degree of violation. Interior penalty functions are applied to optimisation problems in which the perceived optimal solution exists proximate to the boundary constraint, such that points that are not close to the interface between feasible and infeasible regions are penalised. Notably, the introduction of penalisation variables into the objective function has the undesirable effect of producing severe discontinuities within the objective function's topography. These sudden shifts exist at the boundary between feasible and infeasible solutions, with the potential to cause premature convergence and optimisation stagnation if applied excessively.

2.3.7. Criticism of metaheuristic research

The rapid expansion of research within the field of metaheuristics has been questioned in recent years, with some academics positing that the creation of new and "novel" metaphor-based algorithms has the risk of severely diluting promising research and shifting the academic focus of metaheuristic research away from rigorous scientific standards. This critique is founded on the observation that recent publications within the field have made far-reaching claims regarding the discovery of new optimisation algorithms that have been inspired by some phenomenon observed within nature. Critics of this research regard many of these claims as metaphorical fallacies, arguing that many of the metaheuristic algorithms that have been created in recent years are not truly expanding metaheuristic research to new heights and are collectively detrimental to the reputation of metaheuristics as an academic field. These algorithms are described as a simple repackaging of pre-existing concepts, wrapped in the aesthetic of some new exciting metaphor claiming to shift the optimisation paradigm while distracting from promising research within the field [67]. Consequently, this thesis has not taken a significant focus on developing entirely new foundational metaheuristic algorithms. Instead, the improvement and application of existing approaches such as differential evolution, particle swarm optimisation, and the genetic algorithm has been examined within an interdisciplinary deep learning context.

2.4. The genetic algorithm

Within the context of computational design, genetic algorithms (GA) are a search heuristic that use the fundamental principles of natural selection, crossover, and mutation to generate an optimised solution. As observed from Darwinian analysis, populations of living organisms are in a constant state of adaptation and change. The natural variation and mutation of individuals within a population ensure that there are members of each species that have traits which are better suited to the environment in which that species is immediately exposed to. This improves the survivability of these individuals, which in turn increases the chances that these adaptive traits will be passed on to the genes of the next generation through reproduction. Over generations of iterative evolution, species drastically adapt towards favourable traits that maximise survivability [68]. Analogously, genetic algorithms optimise a population of generated solutions by adjusting the genetic code of generated solutions in order to maximise the fitness of sequential generations. The genetic algorithm search function is executed through five phases, which aim to replicate the process of natural evolution. Conventionally, these phases include the generation of the initial population, the user defined fitness function, selection of the fittest solutions for each generation, the crossover of genetic code from each pair of parent solutions for the next generation, and enforced diversity through low rates of genetic mutation.

Genetic algorithms provide a holistic, black-box tool that has been applied across a diverse range of optimisation problems and constraints. As the algorithm is not reliant on gradient information or other prerequisite access to task-specific data, genetic algorithms have experienced a broader application across problems within the field of structural optimisation when compared to gradient-based methods. Analogously, the flexible nature of genetic algorithms furthermore allows the algorithm to be applied for linear or non-linear problems, with discrete or continuous bounds. This is particularly relevant for structural design problems, as issues of constructability, manufacturing, transportation, and connectivity create significant design constraints which must be respected by the chosen optimisation strategy. Early research applications of the genetic algorithm within the context of structural design implemented standardised benchmark tasks with a range of fixed constraints, however in recent years this focus has evolved to incorporate industry-focused objective functions — such as a reduction in project carbon footprint [69], material waste [70], compliance minimisation [71], and cost [72].

The pseudo code for the conventional genetic algorithm may be described as follows:

Genetic Algorithm

1. Initialise the initial population (P(t = 0))
2. Define the desired objective function
3. Evaluate the fitness of each member within the initial population ($P(t = 0)$)
4. **while** generation limit has not been reached or convergence criteria is not satisfied
5. Select the parents $P_p(t)$ for the next generation
6. Generate children $P_c(t)$ which will comprise the next generation (Crossover)
7. Mutate members of the next generation $P_c(t)$
8. Evaluate the fitness of each member within $P_c(t)$
9. $t = t + 1$
10. **end**

2.4.1. Generation of initial population

The first population of solutions generated by the genetic algorithm typically utilise a random or quasi-random parameter distribution, which are evaluated according to their fitness for a given problem. The primary objective of this preliminary step in the optimisation process is to evenly distribute the population of potential solutions throughout the search space. The design parameters tuned by the genetic algorithm are referred to as a solution's 'genes', which are joined together to form the chromosome. By sampling the first generation of the genetic algorithm randomly throughout the search space, genetic diversity is introduced into the population which facilitates explorative optimisation. One of the fundamental assumptions that characterised early forms of the genetic algorithm was that the input parameters which describe the generated solutions could be represented in a binary genetic format. This encoding representation required design variables to be encoded as bit strings, inhibiting the resolution and precision for the parameters to be adequately tuned. Historically, these forms of binary encoding have been one of the most commonly implemented representation scheme for genetic data. Contemporary forms of the genetic algorithm are now capable of representing these parameters as either binary, floats, integers, or tree graphs, based on the encoding requirements inherent to the design problem.

2.4.2. Selection and Crossover

The process of natural selection as observed in nature serves as an analogy which guides the optimisation logic behind the genetic algorithm. Within nature, members of a population which are the fittest for their given environment have a greater chance of mating and passing on their genes to the next generation. Conventional genetic algorithms employ a range of possible selection modules such as tournament selection, roulette wheel selection, and rank selection, which increase the probability that the fittest solutions within each population are paired together [73]. For roulette selection, parents are chosen from a given population according to a simulated roulette wheel, in which individuals are provided a region of the wheel that is proportional to their fitness scores. Notably, this selection operator still allows poorly performing solutions to participate with a reduced probability, promoting genetic diversity from generation to generation. In contrast, tournament selection randomly selects

several individuals from the population to compete against each other for selection as parents to the next generation. By selecting the solution with the greatest fitness from this candidate pool, tournament selection removes the need to evaluate the entire population and thereby reduces the computational cost of the optimisation process. Finally, the rank selection method orders an entire population of solutions according to their individual fitness rank, with selection probabilities being assigned according to position rather than score. This effectively smooths out the selection distribution and reduces the probability of premature convergence caused by extreme variations in fitness performance across a population.

Crossover refers to the method by which the genetic code from each pair of parent solutions is passed on to the next generation. It is a critical component of the genetic algorithm, as it introduces genetic diversity which allow the search space to be productively navigated according to past performance. Once a pair of parents have been chosen using the selection operator, the genes of each solution are combined according to a defined crossover scheme. Single-point and double-point crossover are two widely used techniques to divide the genetic code of each parent. For the single-point crossover, the chromosomes of each parent are aligned with one another and a point along the genetic string is randomly selected. This divider indicates the point at which the genetic code is split and recombined from each parent to the child solution. For double-point crossover, two dividing points are randomly generated to split the exchange of genetic information. Other crossover techniques that are also implemented within the genetic algorithm include uniform crossover and heuristic crossover [74].

2.4.3. Mutation

The final stage within the genetic algorithm optimisation loop is the mutation function, which randomly modifies genes within an individual chromosome to enforce genetic diversity. The degree of mutation is controlled by the mutation rate hyperparameter, which defines the probability of modification for each individual gene within the solution. Simulations which adopt a large mutation rate will closely approximate random search, as the gene pool of each population will begin to resemble random collections of genes rather than collections of subspecies. Due to this undesirable convergence, mutation rates for genetic algorithms are often fixed to low values, allowing for explorative genetic diversity to be induced while preserving broader genetic similarity within each population [75].

2.5. Differential evolution

Designed as a black-box solver for continuous and discrete variables, differential evolution (DE) algorithms belong to the broader family of evolutionary algorithms (EAs). Much like the genetic algorithm, DE algorithms utilise a population of candidate solutions that are manipulated through mutation, recombination, and selection to explore the search space and converge the genetic pool of the population towards an optimum design. Notably, Differential Evolution varies from the conventional GA through its diverse range of mutation operators, which

dictate the exploration-exploitation behaviour of the algorithm in unique ways. While genetic diversity is mediated within the genetic algorithm through a mutation rate, the mutation operators within the differential evolution algorithm offer a more consolidated and targeted strategy towards convergence [76].

The initial population of the Differential Evolution algorithm is randomly generated within the predefined variable bounds that are chosen for the given design problem. For each individual $x_i^{t=0}$ in an initial population of size NP, the genetic vector of length D for each design variable is given by the following:

$$x_i^{t=0} = \{x_{i,1}^0, x_{i,2}^0 \ldots, x_{i,D}^0\} \tag{2.8}$$

$$\text{with} \quad x_{i,j}^{t=0} = x_{min,j} + rand_{i,j}[0,1](x_{max,j} - x_{min,j}) \tag{2.9}$$

where $rand_{i,j}[0,1]$ is a random float chosen between [0,1] and $x_{max,j}$, $x_{min,j}$ represent the maximum and minimum design bounds for each respective variable. For each of the consecutive generational steps that follow, this randomised genetic pool is manipulated through the mutation vectors v_i^t. The behaviour of the differential evolution process is highly predicated on the selection of the mutation scheme that generates these vectors, as they dictate the exploration-exploitation response throughout the optimisation process. The most frequently used mutation operators that are applied to structural design problems include:

$$\underline{\text{DE/rand/1:}} \quad v_i^t = x_{R_1}^t + F(x_{R_2}^t - x_{R_3}^t) \tag{2.10}$$

$$\underline{\text{DE/rand/2:}} \quad v_i^t = x_{R_1}^t + F(x_{R_2}^t - x_{R_3}^t) + F(x_{R_4}^t - x_{R_5}^t) \tag{211}$$

$$\underline{\text{DE/best/1:}} \quad v_i^t = x_{best}^t + F(x_{R_1}^t - x_{R_2}^t) \tag{2.12}$$

$$\underline{\text{DE/best/2:}} \quad v_i^t = x_{best}^t + F(x_{R_1}^t - x_{R_2}^t) + F(x_{R_3}^t - x_{R_4}^t) \tag{2.13}$$

$$\underline{\text{DE/current-to-best/1:}} \quad v_i^t = x_i^t + F(x_{best}^t - x_i^t) + F(x_{R_1}^t - x_{R_2}^t) \tag{2.14}$$

For each of the given mutation schemes, F is a scaling factor which mediates the impact of the differential vector, $x_{R_{1-5}}$ are five randomly selected unique individuals from the population, x_{best} is the solution with the highest fitness within the population, and x_i is the current individual within the iterative population loop [77]. Notably, conventional implementations of the differential evolution algorithm rely on the static selection of one of these mutation operators. This research posits that these mutation schemes offer a diverse variety of mutation responses which may be parametrically selected from throughout the optimisation process to improve convergence performance.

The pseudo code of the differential evolution algorithm is as follows:

Differential Evolution Algorithm

1. Initialise the initial population (P(t = 0))
2. Define the desired objective function
3. Evaluate the fitness of each member within the initial population $(P(t = 0))$
4. **while** generation limit has not been reached or convergence criteria is not satisfied
5. **for** i = 1 to NP
6. Select random unique individuals $x_{R_1}, x_{R_2}, x_{R_3} \ldots$
7. Calculate v_i^t using chosen mutation operator
8. $x_{off} = v_i^t$
9. **for** j = 1 to D
10. Generate *rand*(0,1)
11. **if** *rand*(0,1) ≥ CR
12. $x_{off,j} = x_{i,j}$
13. **end if**
14. **end for**
15. **if** $f(x_{off}) \leq f(x_i)$
16. $x_i = x_{off}$
17. **end if**
18. Update fitness record for x_i
19. **end for**
20. **end**

Across all variations of the Differential Evolution algorithm, the mutation and recombination operators are critical to the overall performance of the optimisation process. For instances in which the degree of genetic exploration is excessively large, the fittest individuals within each generation are often destroyed before their genetic code is passed to the next population. Conversely, when exploration is not prioritised within the mutation operation, the genetic diversity of each population is significantly limited, reducing the capacity of the algorithm to explore potentially competitive solutions and often resulting in premature convergence at a local optima [78]. This challenge is illustrative of the perpetual trade-off between exploration and exploitation that characterises almost all optimisation methodologies. In most cases, whether it be within the field of metaheuristics, machine learning, or Bayesian optimisation, this trade-off is reconciled by the initialisation of some set of hyperparameter values or acquisition functions. This initialisation is typically based on extensive experimentation or the experience of the experiment designer, which is significantly time consuming and often creates misleading results which exaggerate an algorithm's generalisation capabilities or its performance relative to existing benchmarks.

2.6. Particle swarm optimisation

Particle swarm optimisation (PSO) has demonstrated significant utility as powerful black-box solver when applied to structural optimisation problems, producing competitive candidate solutions through a nature-inspired approach to search space exploration. Within the context of PSO, solutions are referred to as 'particles' which collectively comprise the population, or the 'swarm' of each generation. These particles navigate the design search space while recording their current position and velocity. The position of each particle is calculated based on its previous position within the search space, updated iteratively according to the following equation:

$$x_{k+1}^{i} = x_{k}^{i} + v_{k+1}^{i}\,\Delta t \tag{2.15}$$

where x_{k+1}^{i} and v_{k+1}^{i} correspond to the position and velocity vector of the i^{th}particle within the swarm at generation $k+1$ and Δt is given as the time step for the optimisation environment [79]. For the purposes of this research, the simulation time step has been taken as $\Delta t = 1$, which has been carried for subsequent equations for clarity. The velocity vector is defined by the best-known historic position observed by the individual particle, in addition to the position of the best performing particle across the entirety of the swarm. For each particle i in generation k, the velocity v_{k}^{i} is given by the following:

$$v_{k}^{i} = w v_{k}^{i} + c_1 r_1 (p_{i}^{k} - x_{k}^{i}) + c_2 r_2 (p_{sw}^{k} - x_{k}^{i}) \tag{2.16}$$

where w is the weight of inertia definition the influence of the previous velocity vector, x_{k}^{i} is the position of the current particle, p_{i}^{k} is the historic position of best performance for particle i, p_{sw}^{k} is the best positional performance across all particles within the swarm, c is a positive scaling coefficient and r is a coefficient with random initialisation between zero and one. This unique directional movement is calculated for each particle according to the location within the search space that yielded the best performance. When a member of the population discovers a combination of design parameters that exceeds the historic performance of all particles, this location becomes the updated coordinate point of attraction. As a result, the exchange of information across all members of the swarm contributes significantly towards the convergence rate of the algorithm.

2.7. Variations in metaheuristic behaviour

Metaheuristic algorithms offer a vast range of unique approaches for practitioners to navigate the complexity of high-dimensional design tasks. Consequently, the selection of an appropriate algorithm for a given design task is inherently dependent on the anticipated fitness landscape, the allowable computational budget, the expected rate of convergence, and the desired exploration-exploitation response. Evolutionary based methods such as the genetic algorithm and differential evolution aim to simulate biological evolution through genetic mutation and crossover, which adapt the parameters of potential solutions over successive generations. These algorithms drive a balance between exploration and exploitative recombination of existing solutions, with DE specifically tailored to intuitively control this tradeoff. Consequently, these stochastic algorithms typically exhibit a slower rate of convergence when compared against metaheuristics that prioritise exploitative behavior. In contrast,

particle swarm optimisation has been designed based on the observed behavior of natural organisms within groups. This velocity-based movement is directly informed by the historic performance of each particle as well as the best performing solution within the swarm. As a result, swarms tend to rapidly collapse towards the elite particles location, increasing convergence speed while also leaving the algorithm vulnerable to premature convergence towards local optima.

Notably, the introduction of constraints and discontinuities within the fitness landscape has a propound impact on the viability of these algorithms. The combinatorial and stochastic nature of differential evolution and the genetic algorithm allow for these algorithms to traverse discontinuous regions and escape localised fitness pitfalls. Importantly, these algorithms do not rely on the smoothness of the fitness landscape to navigate it effectively. In contrast, particle swarm optimisation is significantly impacted by any irregularities within the search space, as smooth trajectories between particles and historic locations are assumed. PSO is particularly sensitive to fitness landscape discontinuities, which can trap large portions of the swarm and cause irreparable convergence stagnation [80, 81].

2.8. Adaptive metaheuristics

Within the academic field of metaheuristics, there is a significant precedent of research that illustrates the application of solvers such as the differential evolution algorithm for structural optimisation. In recent years, there has also been a growing interest in the application of various machine learning techniques for analogous structural problems, such as Bayesian optimisation, supervised learning, and reinforcement learning [82-84]. Despite the continuous development of these parallel fields within academia, there is currently very little research being conducted that is investigating the intersection of metaheuristic and machine learning within the context of our built environment. The symbiotic potential of these disparate yet parallel fields of research has yet to be fully realised. This research gap provides a crucial opportunity to investigate hybrid algorithms and novel modelling techniques that extend beyond existing research limitations. The fundamental variation between gradient-based learning methods, surrogate probability models, and metaheuristic algorithms provides a diverse range of optimisation techniques which perform exceptionally well within different contexts. In this sense, research that operates at the intersection of these fields is uniquely positioned to evaluate and compare these leading optimisation strategies and accordingly identify their potential multidisciplinary applications within industry.

All metaheuristic algorithms require sets of hyperparameters to be predefined prior to the execution of the optimisation process, which are critical to algorithmic performance [85]. This has been observed by within the context of evolutionary algorithms, the performance of which is highly dependent on the initialisation of key

hyperparameters which mediate the exploration-exploitation response [86]. These hyperparameters are conventionally encoded as static variables, such that the convergence behaviour of the algorithm is fixed throughout the optimisation process. This precedence within the metaheuristic literature is antithetical to observations made within Bayesian research. Bayesian optimisation is predicated on the division of the optimisation process into two distinct phases: the explorative sampling of the search space and the exploitative convergence on observed optimal solutions [87]. Bayesian researchers have furthermore demonstrated that a dynamic exploration-exploitation response can consistently outperform static behaviour by providing a diverse portfolio of acquisition functions that may be chosen from throughout the optimisation process [88]. Notably, the diverse mutation operators commonly implemented within Differential Evolution algorithms offer a similar opportunity to develop a portfolio of explorative and exploitative methodologies which may be dynamically selected from to mediate this trade-off.

Within the field of structural optimisation, reinforcement learning (RL) agents have been successfully implemented as a hybrid implementation with existing metaheuristic algorithms to achieve a similar effect. The combination of powerful metaheuristic algorithms and reinforcement learning models was benchmarked by Huynh, Do and Lee, 2021, in which the crossover rate and scale factor hyperparameters of a differential evolution (DE) algorithm were directly controlled by a Q-learning model [89]. By creating a hybrid interface between these two algorithms, researchers were able to overcome the significant limitations of both Q-learning and DE. Namely, while Q-learning is a powerful and simple tool, its applications are significantly limited when scaled to high dimensional or continuous environments. As the model only needs to account for two parameters within a discrete set of possible combinations, this paper demonstrates that reinforcement methods such as Q-learning are capable of providing dynamic hyperparameter control during the optimisation process as a parametric exploration-exploitation strategy for a confined portfolio of actions.

Researchers have recently confirmed these observation within the context of particle swarm optimisation by introducing a Q-learning model as a controller for level-based PSO. Within this implementation, a Q-table was developed to track the performance of various level sizes, which divided the swarm into distinct groups according to the desired exploration-exploitation trade-off. Particles were divided into these groupings according to their fitness, before selecting an exemplar particle from a higher level within the hierarchy. These exemplar particles serve as attractor points to the updated velocity of lower-level particles, allowing for an interactive optimisation process. By controlling the size delimitation of these levels, the degree of exploration was mediated by a Q-learning model to improve convergence rate and the fitness of the final solution [90].

Conversely, there exists an emerging interest in research that applies metaheuristic algorithms for the selection of hyperparameters used by highly specific machine learning models. Genetic algorithms have been used

successfully in previous research to determine locally optimal values of six variables used by the Deep Deterministic Policy Gradient (DDPG) method. These included the learning rate of the actor and critic networks, the standard deviation of introduced noise, the discount factor, the epsilon value which determines the quantity of random actions taken to induce explorative behaviour and the polyak averaging coefficient. Researchers have demonstrated that algorithmically selected parameters can improve the success rate of the agents relative to preliminary parameter initialisation [91]. Research emerging from this field indicates that that heuristic search methods are capable of generating competitive collections of hyperparameter sets, which can ultimately improve model learning rates and performance when measured against prior baselines.

2.9. Bayesian optimisation

Notably, metaheuristic algorithms provide no direct solution to the computational cost of each objective function evaluated during optimisation. Efficiencies between algorithms within the metaheuristic family are typically gained by modifying the exploration-exploitation behaviour or by altering the transfer of information between generations of proposed solutions. For these algorithms, the underlying cost of the objective function largely remains unchanged, severely limiting the application of metaheuristics for structural tasks that require intensive and time-consuming finite element calculations or simulations. Forms of distributed computing and parallelisation are necessary in these cases to analyse candidate solutions at a rate that is useful for industry practitioners. In light of these factors which bottleneck conventional metaheuristic solvers, Bayesian optimisation has gained significant traction and interest throughout the research community as a viable global design strategy.

Bayesian optimisation (BO) is a highly effective technique employed in the development of probabilistic proxy models, leveraging previous observations collected from a computationally expensive objective function as training data for a surrogate model. Typically, Bayesian optimisation addresses problems which arise in the form $x_* = argmax_{x \in X} f(x)$, such that the solver seeks to locate the global maximum of the function f. Within this context, X is taken as the set of points x, belonging to a phase space which is typically less than 20 dimensions. This is primarily due to what is commonly referred to as the curse of dimensionality, in which the necessary sampling process and acquisition function optimisation for high-dimensional problems drastically undermines the computational efficiency, robustness, and validity of the proxy model.

Bayesian optimisation may be divided into two distinct components, each requiring considerable deliberation and care in their construction. First, the iteratively changing proxy model which seeks to replicate the behaviour of a target function which is typically computationally expensive to sample from. Second, the use of a selected

acquisition function $\alpha(x)$ to scalarise the probabilistic model and select the next query point for evaluation [92]. The generalised BO approach is as follows:

1. Collect initial sample data from the simulation for training
2. Initialise the proxy model using the sample data set
3. Determine the acquisition function $\alpha(x)$ which will be used to select the next query point
4. Optimise the acquisition function globally to select the point $x_{next} = argmax\ \alpha(x)$
5. Sample new data and update the proxy model based on new information
6. Repeat steps 3 → 5 until convergence or the iteration limit is achieved
7. Provide a recommendation based on the best performing solution

As a black-box optimisation strategy, Bayesian Optimisation does not require direct access to the target function. Rather, the Bayesian methodology interprets the target as a random function and places a probability distribution over it using what is known as a Gaussian Process. Within the context of Bayesian statistics, a prior represents the initial belief about the target functions behaviour before observing any data. This prior distribution is updated as new observations are collected, forming the posterior distribution which guides the optimisation process. The incorporation of new test evidence allows for the posterior probability distribution to be derived via Bayes' theorem, $P(A|B) = P(A)\,P(B|A)/\,P(B)$. In this equation, $P(A)$ provides the the prior probability of "A" being true before considering new evidence, $P(B|A)$ is the likelihood function that represents the probability of "B" being true if given that "A" is true and $P(B)$ expresses the probability that "B" is true. The posterior provides the conditional probability distribution given the newly observed function evaluations, which is then used to establish the acquisition function $\alpha(x)$ that may in turn be optimised to select the next query point.

2.9.1. Surrogate modelling

For complex structural problems, computer simulations such as the finite element method are used to solve mathematical calculations which are too intensive to be solved by hand. These simulations are often computationally expensive, drastically hindering the efficiency of optimisation techniques that require accurate and comprehensive structural analysis. For these applications, surrogate models offer a potentially robust solution, by serving as a meta-model for the required simulation. If a sufficient quantity of high-quality data is available, a well-fitted surrogate model $\hat{f}$ may be used to substitute for the simulation f, potentially improving the computational efficiency by orders of magnitude. When appropriately integrated into the optimisation workflow, surrogate models have been demonstrated as a flexible, pragmatic, and highly efficient placeholder for computationally expensive analytical simulations. In some cases in which the surrogate model is adequately designed, the surrogate may even produce more desirable output than the simulation that it is replicating, due to the smoothing of noisy data that is achieved through Gaussian Process regression.

Surrogates must satisfy four distinct criteria in order to justify their implementation [93]. Namely, a surrogate model $\hat{f}$ must:

1. leverage a sufficiently representative sample from the target function to create the predictive distribution $\hat{f}(x')$. The mean as calculated by the acquisition function may then be used as a proxy model for $f(x')$ at newly optimised locations x'. The variance at these points as calculated by the acquisition function provides estimated measurements of the model uncertainty at x', which informs the exploration-exploitation trade-off
2. include the capacity to interpolate intermediary data for cases in which the desired computer simulation f is deterministic
3. provide outputs analogous in accuracy and type to f for optimisation
4. consist of a fitting process ($\hat{f}$) and prediction capabilities $\hat{f}(x')$ which are collectively executed faster than simply computing $f(x')$. This is especially important, as the justification of surrogate modelling is predicated on the assumption that the strategy will improve computational efficiency.

2.9.2. Gaussian Process

A Gaussian Process (GP) is a probabilistic model defined by a mean function $\mu(x)$ and a covariance function $k(x, x')$. The GP utilises a collection of previous observations to construct a smooth posterior distribution over all possible functions that can fit to the historical dataset. This allows the posterior mean $\mu(x_*)$ (expected value) and posterior variance $\sigma^2(x_*)$ (uncertainty) to be calculated, which in turn drives the chosen acquisition sampling scheme. The first step within the Gaussian Process is to develop the covariance function, which defines the similarity between two input points. One of the most widely implemented covariance functions is the radial basis function (RBF), defined as follows:

$$k(x, x') = \exp\left(-\frac{(x-x')^2}{2l^2}\right) \tag{2.17}$$

where x and x' are two input points and l is a hyperparameter called the length scale. Points that are proximate to one another are assumed to yield similar results when processed by the target function f, with RBF values approaching 1 as correlation increases and 0 as these points are sampled further apart from one another. The length scale hyperparameter controls the degree to which correlation similarity decays as the distance between points increases. For a given collection of n input points, a complete $n \times n$ covariance matrix (K) is calculated to represent the similarity between all training samples. The similarity between each point and a new test point x_* is then calculated, creating a full covariance vector as follows:

$$k_* = [\, k(x_*, x_1), k(x_*, x_2), \ldots, k(x_*, x_n)] \tag{2.18}$$

The covariance matrix K and covariance vector k_* can then be used to calculate the posterior mean and variance as follows:

$$\mu(x_*) = k_*^{\top}(K + \sigma_n^2 I)^{-1} y \tag{2.19}$$

$$\sigma^2(x_*) = k(x_*, x_*) - k_*^{\top}(K + \sigma_n^2 I)^{-1} k_* \tag{2.20}$$

where y is a vector of previously observed outputs, I is the identity matrix, and σ_n^2 is the assumed gaussian noise that is added to the covariance matrix to account for imprecisions that may be present in the observation dataset. Under this probabilistic framework, the mean $\mu(x_*)$ represents the predicted value of the target function f at x_*, while $\sigma^2(x_*)$ measures the uncertainty of this prediction [94]. Notably, these equations require the noise-adjusted covariance matrix to be inverted, which can be extremely computationally expensive for large n observation datasets—revealing a key limitation of GPs when applied to practical surrogate modelling. The selection of the next point x_* is entirely informed by a carefully designed acquisition function, which aims to mediate the exploration-exploitation response.

2.9.3. Acquisition functions

Within the context of Bayesian optimisation, acquisition functions serve to dictate the next query point within the search space by interpreting the posterior predicted mean and variance generated through the Gaussian process. Conventionally, this is achieved through the exploration-exploitation trade-off, with the chosen function aiming to leverage some quantifiable strategy to either sample unobserved regions of the search space or draw its samples from the existing regions that produce the highest quality solutions. Across conventional implementation of Bayesian optimisation, there are three dominant acquisition functions which are extremely popular. These include the probability of improvement (PI), the expected improvement (EI) and the upper confidence bound (UCB) functions. In its conventional implementations, the probability of improvement function provides the probability that a point sampled at x will outperform the current best candidate solution, according to the following equation:

$$PI(x) = \psi\left(\frac{\mu(x) - f(x^+) - \xi}{\sigma(x)}\right) \tag{2.21}$$

where ξ is a parameter which defines the exploration threshold, $\sigma(x)$ and $\mu(x)$ provide the standard deviation and mean of the predictive function at x, and $f(x^+)$ represents the best function value that has been calculated so far. The parameter ξ may be initialised with a relatively large magnitude to promote exploration during the early phases of optimisation, before a decay is introduced over subsequent iterations to transition towards exploitative behaviour [95].

In practice, the probability of improvement method can often result in simplistic and greedy behaviour. This is due to fact that the PI function is designed to reward solutions which are improvements relative to the current

optimal solution $f(x^+)$ without considering the degree of the improvement itself. The PI method only considers the probability of a particular solution exceeding the previous best, with exploration directly controlled by the hyperparameter ξ. In this sense, the probability of improvement function tends to be overtly exploitive, rewarding points which are proximate to the existing optimal solution. This is demonstrative of the underlying exploration-exploitation trade-off that is inherent to practically all forms of algorithmic optimisation. The expected improvement (EI) function provides an alternative approach to managing this trade-off by promoting solutions which are expected to outperform the current best by a significant margin, while simultaneously sampling points that have a high uncertainty. The expected improvement acquisition function is governed by the following equations:

$$x_{n+1} = argmax\ EI_n(x) \tag{2.22}$$

$$EI(x) = \begin{cases} (\mu(x) - f(x^+) - \xi) \cdot \Phi(Z) + \sigma(x) \cdot \phi(Z) & \text{if } \sigma(x) > 0 \\ 0 & \text{if } \sigma(x) = 0 \end{cases} \tag{2.23}$$

$$Z = \frac{\mu(x) - f(x^+)}{\sigma(x)} \tag{2.24}$$

where x_{n+1} is the next point to be sampled within the design space, n is the current iteration, Z is the Z-score which standardises the difference between the current mean and the current best value, $\Phi(Z)$ is the cumulative distribution function which captures the probability of improvement, and $\phi(Z)$ is the probability density function which increases the expected improvement at points of larger uncertainty. Like the PI method, the degree of exploration within the expected improvement function may also be moderated by introducing a threshold variable ξ. The expected improvement method is one of the most widely used acquisition functions for Bayesian optimisation, primarily due to its simplicity and performance when subject to uncertainty. Solutions which exhibit a low mean are exploitative points which are proximate to previously captured data, while points with larger degrees of uncertainty require the exploration of uncharted domains within the search space. The acquisition point which has the highest expected improvement is selected as the next query point for evaluation.

Similarly, the upper confidence bound acquisition function rewards the exploration of unexplored regions of the design space while estimating regions of high expected value. The upper confidence bound acquisition function is governed by the following equations:

$$x_{n+1} = argmax\ UCB_n(x) \tag{2.25}$$

$$UCB(x) = \mu(x) + \kappa \cdot \sigma(x) \tag{2.26}$$

where the exploration is governed by the hyperparameter κ which encourages the method to sample areas with higher uncertainty. Extensive research has been conducted to evaluate and compare the behaviour of these

acquisition functions. The EI method has become widely adopted due to its exceptional performance across a diverse range of applications and its magnitude-aware value estimation, however it requires the computation of the probability density and cumulative distribution functions. In contrast, while the probability of improvement function does not account for the magnitude of potential improvements, its values are easily interpretable and easy to compute. Finally, the upper confidence bound method is extremely simple and intuitive, however the function does not account for the probability or magnitude of improvement and its performance is heavily dependent on the manually tuned exploration hyperparameter (κ).

2.9.4. Bayesian optimisation in design

In recent years, there has been growing interest in the application of Bayesian optimisation for the creation of proxy models to aid the design process. The primary method common to these optimisation strategies is to initially capture a representative sample of a design space which typically has a domain of less than 20 dimensions. This initial sampling is then used to construct a surrogate model that aims to capture the structural response of potential designs, reducing the need to execute extensive finite element simulations for structural optimisation. Conventionally, the focus of this optimisation strategy has been on cost minimisation, however the changing demands for sustainable design and the rising popularity of multi-objective solvers has significantly broadened the scope of structural optimisation to include parameters such as performance and buildability. Researchers have developed unique applications of constrained multi-objective Bayesian optimisation for structural design accounting for material expenditure, buildability and structural performance [96]. Within this previous research, structural analysis was conducted for simply supported concrete beams with reference to the European design codes, including the induced shear and bending relative to the ultimate limit state and the deflection induced relative to the serviceability limit state. The proposed optimisation strategy was benchmarked against random search and the elite non-dominated sorted multi-objective genetic algorithm NSGA-II. Measured performance metrics included the quality of the generated designs, the structural response, the rate-of-improvement, and the variance that was induced between 25 repeated executions of each method. Across these metrics, the Bayesian Optimisation strategy outperformed the benchmark solvers. After the initialisation of the Bayesian model, the proxy rapidly exceeds the performance of both random search and the NSGA-II algorithms before converging at a solution after roughly 400 iterations, demonstrating significant efficiency when compared to the budgeted 1000 iterations required from the benchmark solvers. The paper concluded that for all five objective functions, the Bayesian surrogate model provided the highest quality pareto-optimal solutions when compared against an elite multi-objective genetic algorithm.

Further research into the application of Bayesian optimisation for multi-objective structural engineering has been conducted within the context of steel frame structures [97]. This research investigated three conflicting objective functions to be minimised, including the structural mass, the average structural inter-story drift and the variance in the inter-story drift. Serviceability and strength performance of the planar steel frames were

optimised in accordance with the standards provided by ANSI/AISC 360–16, with column and beam elements selected from a predefined collection of 40 manufactured sections. The creation of a catalogue of column and beam members was primarily based on design feasibility and intuition, with the goal of reducing stochastic variance induced between executions of the model. This research provides a precedence for the application of Bayesian optimisation within the context of constrained multi-objective problems with discrete element selection, aligning academic applications of BO with relevant constraints required for practical design applications.

The rising interest in the application of Bayesian optimisation within the context of structural design has in turn introduced the need for a comparative study between the performance of traditional heuristic and BO methods. This analysis is particularly necessary within the field of structural engineering, as the repeated execution of complex finite element simulations significantly impedes the practicality of many optimisation strategies. Research conducted by [98] directly addresses these questions by comparing the performance of conventional heuristic algorithms against Bayesian optimisation. In order to benchmark these optimisation techniques, a parametric model of a pedestrian box-girder bridge was created. The section geometry was selected as the input parameters of the black-box solvers, including the section depth, bottom slab width, inclined web thickness, top slab thickness, cantilevered thickness, bottom slab thickness, and web slab thickness. These seven variables were optimised within a confined domain with the objective of minimising the structures embodied energy, with consideration for the extraction and manufacturing of raw materials, as well as the transportation and construction of structural elements. This single objective design problem allowed researchers to observe the accuracy of the proxy kriging model, which effectively reduced the computational costs of the finite element simulations by 99.06%, while still producing competitive solutions relative to the heuristic solver. This surrogate model was created with an initial sampling size of 50, resulting in an ultimate surface accuracy of 4.04% when compared with the target function. Notably, researchers found that the tuning of the model's sampling size offered the BO method a significant degree of versatility in its application, as the optimisation strategy was able to be designed according to the desired trade-off between accuracy, variance, and time investment.

2.9.5. Limitations of Bayesian optimisation

As the number of design parameters increases for a given optimisation application, the size of the prior dataset necessarily increases in order to capture an accurate generalisation of the multi-dimensional phase space. This raises the significant problem of computational efficiency when applying Bayesian optimisation for high-dimensional tasks. Consider a sampling plan which captures an acceptable prediction of a single parameter problem using n sample locations. In order for a sampling plan to capture a similar density of sampling from a k-dimensional phase space, the model will need to sample n^k points. To quantify this example, if 10 samples are taken from the single variable problem, to achieve a similar quality of sampling from a problem containing

8 variables the sample plan would require 10^8 simulations to be run. If each simulation takes an hour to execute, it would take approximately 11,416 years for this data to be adequately sampled [99].

This example provides several important lessons that must be considered prior to designing an optimisation methodology. First, an exhaustive approach that seeks to compute all possible combinations of design parameters is not a viable solution for high-dimensional computationally expensive problems. Secondly, the efficiency and viability of a surrogate model is highly dependent on the dimensionality of the given problem. Designers should take care to minimise the number of necessary parameters and consider removing redundant parameters from the model altogether. This will largely rely on the intuition of the engineer developing the optimisation strategy, reinforcing the use of these proposed methods as supplementary tools that are generally problem dependent. An experienced structural engineer will be capable of intuitively eliminating or proposing problem specific variables in order to reduce the dimensionality of the problem and capture a collection of key parameters that allow for effective optimisation. It must be noted here that certain forms of optimisation are far more sensitive to these limitations. For example, conventional Bayesian optimisation is a sequential method by nature, making it extremely difficult to parallelise or distribute the computational costs to other systems. Sequential methods bear the full burden of high-dimensional optimisation problems, while alternative optimisation techniques such as metaheuristics are capable of generating potential candidates in batches which may be solved across multiple machines. It must be noted here that the Bayesian method excels for computationally expensive simulations, in which the efficiency of the proxy model is emphasised due the time investment required for extensive simulation executions. Additionally, in many professional contexts, access to parallelisation or simulation distribution tools may be significantly limited by commercial resourcing or the software licensing available to the practicing engineer. Within this context, a reduced number of simulation executions enriches the argument for Bayesian optimisation as a pragmatic and robust strategy.

There are many more limitations to Bayesian optimisation beyond dimensionality which must be holistically considered when choosing a suitable optimisation methodology. Traditional methods of Bayesian optimisation are particularly sensitive to the selection of their fixed hyperparameters, creating variability between model outputs based on the user define parameter sets. This is particularly relevant when the selected acquisition function includes a hyperparameter which controls the exploration-exploitation trade-off. Additionally, the selection of the acquisition function itself has a significant influence on each iteration of the optimisation process. Empirical evidence established through benchmarking analysis has concluded that it is more advantageous to adapt the acquisition strategy sequentially throughout the optimisation process [100]. Researchers have sought to address this by developing a portfolio of potential acquisition functions which can be selected from dynamically. This resulted in the creation of the entropy search portfolio (ESP), as a principled method of selecting acquisition functions from a predefined list based on some meta-criterion [101]. For ESP driven algorithms, the acquisition function which results in the greatest expected reduction in entropy is selected

for the given iteration, which has demonstrated performance improvements when compared against runs that use a single acquisition function.

Importantly, Bayesian optimisation requires a set of data to initialise the surrogate model, which increases in size according to the dimensionality of the problem. The knock-on effect of a larger dataset is that it directly impedes the efficiency of the algorithm when optimising the acquisition function, depending on the selected solver. Additionally, in order to calculate the predictive distribution of a Gaussian Process, it is necessary to invert the $n \times n$ kernel matrix Σ^{-1}, where n represents the size of the observed dataset. The inversion of the kernel matrix requires a computational time investment of order $O(n^3)$, severely limiting the size of datasets which Bayesian optimisation can adequately leverage. The surrogate model is useful when it is computationally more efficient to evaluate when directly compared against the objective function. Ignoring edge case applications involving destructive sampling or economically costly data collection, it is difficult to justify the application of traditional GPs for problems involving datasets larger than 3000 observations [102].

The fact that Bayesian Optimisation itself requires an internal optimisation strategy to manage the acquisition function is also an inescapable limitation which requires significant attention when considering the application of BO for a given structural problem. There are several techniques which are conventionally used in practice to find the global maximum of the acquisition function. For simplicity, discretisation is often used to divide the search space into a grid of candidate solutions. If gradient information is available, more sophisticated methods such as quasi-Newton hill-climbing algorithms are recommended to improve performance. Regardless of the chosen solver, the internal management of the acquisition function raises several concerns which researchers continue to struggle with. If the algorithm is not capable of locating the global maximum of the acquisition function at each sequential iteration, the optimisation process is susceptible to getting caught in a local optima and converging prematurely. Additionally, when query points are proximate to one another under an overly exploitative acquisition scheme, consecutive iterations of the surrogate model may not change drastically. This results in a significant waste of processing efficiency, as the cost of constructing and optimising the acquisition function at these iterations is not proportionally worthwhile relative to the minor improvements to the model.

Finally, one commonly omitted fault of Bayesian Optimisation is the fact that the rate at which the method approaches convergence is initially very slow. During the preliminary sampling process, the objective function is conventionally queried using sampling techniques such as Latin Hypercube, Halton, or Sobol sampling. By design, these predefined techniques do not directly inherit any information from the target function, as their purpose is to extract an adequately representative sample dataset for model initialisation. This represents a purely exploratory phase of the Bayesian model's development, in which the sequential improvements recorded within the sampled data have no influence on the sampling technique itself. In some instances, a metaheuristic

algorithm applied to the same problem may be capable of locating an acceptable local optima in a fraction of the time it would take a Bayesian algorithm to begin exploiting the sampled dataset. This issue is more likely to occur for high-dimensional problems, in which the task of generating an adequately representative sample of the search space requires a significant time investment.

2.10. Deep learning

2.10.1. Multilayer perceptrons

One of the most important contributions to the field of machine learning was the development of the backpropagation algorithm and its radical impact on the learning capabilities of neural networks. Prior to its introduction, neural networks did not have an efficient method to train the weights of hidden layers, limiting the application of neuron-based models to shallow architectures and linear tasks. This pervasive challenge characterised a period throughout the 1970s that has become known as the "first AI winter", during which academic interest in neural networks decreased significantly. Sceptics such as Marvin Minsky and Seymour Papert were famously critical of the limited application of these shallow neural networks, casting doubt on the potential future application of perceptron-based models. The breakthrough development and popularisation of the backpropagation algorithm for weight learning in the 1980s radically transformed the field of machine learning by allowing deeper neural networks to be trained. The 1986 landmark paper, "*Learning representations by back-propagating errors*" [103], is considered to be largely responsible for the resurgence of neural networks and the eventual creation of the 'deep learning' field.

The multilayer perceptron (MLP), or full connected (FC), neural network is comprised of multiple layers of neurons. An input layer first takes the raw model input features, which are then passed sequentially through each hidden layer within the network. The neurons of each layer perform a weighted sum of its inputs with an additional learnt bias variable, before passing the result to a non-linear activation function. The outputs from these operations are then provided as the inputs to the next layer. These models are also commonly referred to as feedforward networks due to the one-directional passage of data from input to output during inference. These neuron-based activation calculations may be summarised as follow:

$$z = \sum_i w_i x_i + b \tag{2.27}$$

$$a = f(z) \tag{2.28}$$

where f is the chosen non-linear activation function, x is the neuron input, w is the connection weight, and b is the neuron-specific bias value. This simple architecture has demonstrated remarkable pattern-recognition capabilities by the introduction of multiple hidden layers, with recognised use cases across tasks relating to classification, regression, sentiment analysis, function approximation, signal processing, natural language processing, and time series forecasting. As a landmark architecture, the multilayer perceptron remains a

foundational component of many cutting-edge deep learning architectures, including convolutional neural networks, recurrent neural networks, and transformers.

2.10.2. NEAT and HyperNEAT

Perhaps one of the earliest foundational algorithms developed at the intersection of metaheuristics and machine learning research is known as the Neuroevolution of Augmenting Topologies algorithm (NEAT), created by Professor Kenneth Stanley in 2002. The NEAT algorithm differentiates from conventional machine learning approaches by using a genetic algorithm to optimise the network architecture, removing the need for an experienced machine learning practitioner to define the depth and complexity of the network '*a priori*'. As a self-adapting deep learning system, NEAT belongs to the broader family of topology and weight evolving neural networks (TWEANN), which are capable of outperforming architectures that are manually constructed by human operators [104]. The performance of NEAT has been partially attributed to its roots within the field of metaheuristics, leveraging tools such as crossover to pass genetic information of diverse network topologies between parent solutions. Elitism was also introduced within the original NEAT implementation to preserve the best performing network architectures of each generation, ensuring that mutation and crossover did not inadvertently destroy leading solutions. Within this hybrid genetic algorithm, the genotype of each solution is divided into two lists: the genes of each neuron, and the genes of each connection between neural layers. This genetic representation allows for the unique index and layer type to be stored for all neurons within the network. Generated network architectures which are within a set genetic distance from one another are grouped into a common species, the performance of which is collectively recorded. The survivability of each species is directly dependent on the fitness of its individual members, with extinction introduced when a species fails to improve its performance over a set number of generations.

Complexity is introduced from one generation to another through random genetic mutation, which directly modifies the neurons, nodal connections, and weights of each network. Nodes which are a product of mutation are created by splitting an existing connection into two separate connectors, which are initialised with a weight of one. This allows mutation to introduce complexity into each generation, with models iteratively growing in size according to the given optimisation task. Additionally, the starting model states of the original NEAT implementation use simple shallow network architectures, with early generations beginning as relatively simple perceptron structures. In this sense, evolution within NEAT is a self-regulating process which only retains network complexity when it yields competitive performance. This effectively reduces the risk of overfitting, as the depth and interconnection of the network is designed to evolve according to the complexity of the given problem.

The field of neuroevolution has adapted significantly since its popularisation by Professor Kenneth Stanley in 2002, with variations seeking to scale the evolving principles of NEAT. In 2009, Stanley published a paper detailing a hypercube-based encoding method that enabled neuroevolution to be applied to large scale networks. Termed as the 'HyperNEAT' algorithm, this technique leverages Compositional Pattern-Producing Networks (CPPN) to predict the weights matrix of a target neural network, compressing information about the target network into an efficient predictive model. Within this framework, the CPPN is used as an encoding method that represents the connection relationships of the primary network as functions. Instead of directly storing large collections of weights, CPPNs act as a generative method that maps geometric relationships between nodes and weighted connections. These CPPN models evolve according to the NEAT method, allowing them to iteratively improve through the generational loop. This in turn increases the dimensionality of tasks that the HyperNEAT algorithm can effectively be applied to, demonstrated in Stanley's paper through the successful evolution of a network with over eight million connections [105]. The HyperNEAT algorithm was a significant milestone in bridging the fields of metaheuristics and deep learning, inspiring many later works involving novel encoding methods, meta learning, and neural architecture search. Metaheuristics such as particle swarm optimisation [106], differential evolution [107], and the genetic algorithm [108] have since been introduced as black box hyperparameter optimisers for neural networks across a diverse range of deep learning tasks.

2.10.3. Hypernetworks

The term 'hypernetwork' was introduced by David Ha in 2016 [109], inspired by Stanley's earlier works within the field of neuroevolution. Instead of simply training a single neural network for a given deep learning task, hypernetwork systems generate candidate neural networks via weight prediction. The modern hypernetwork consists of two models: a primary network which is used to generate the desired output for the given task, and a 'hypernetwork' that generates the weights of the primary network. This relationship may be represented as follows:

$$\theta_x = f(x, \theta_f) \tag{2.29}$$

$$s_x^p = g(p, \theta_x) \tag{2.30}$$

where the network f is the hypernetwork that generates the weights θ_x for network g. The hypernetwork takes some input x to predict these weights, which are used to parametrise the network g which predicts output s_x^p given input p. Notably, this affords a level of abstraction in which input x effectively conditions the processing of input p by network g.

The HyperNEAT algorithm may be interpreted as a preliminary form of hypernetwork, in which the network weights are defined according to an evolved Compositional Pattern-Producing Network. This evolutionary relationship between a weight-defining network and a primary network came to inspire the concept of

hypernetworks itself. The key innovation of the hypernetwork architecture that differentiates it from HyperNEAT is the gradient-based training scheme implemented for the network that generates the weights of the primary network. The loss of the output from the primary model is back propagated through the hypernetwork, such that the primary model is never itself trained. Innovative algorithms such as NEAT, hyperNEAT, and hypernetworks all have deep conceptual roots at the intersection of machine learning and evolutionary computation. While novel models that were developed — or inspired — by research positioned at this interdisciplinary intersection have come to demonstrate remarkable performance across a diverse range of deep learning tasks, they have remained unexplored within the complex landscape of the built environment. As a result, the true potential for these models to revolutionise existing approaches and enable new unprecedented predictive capabilities has not yet been revealed. This presents a compelling research opportunity that has motivated each of the papers presented within this thesis, with broad multifaceted implications throughout many civil engineering subdisciplines.

One of the primary advantages of hypernetworks is that the separation of inputs across networks eliminates the need to condition the input for the primary network. While traditional methods such as conditioning-by-concatenation require specific class information to be appended to the input for the primary network, the hypernetwork method provides a form of unstructured learning without direct class labelling. The ability to abstract the network input away from conditioned class information provides the hypernetwork method a degree of modularity that is particularly suited for continual and self-supervised learning. Consequently, researchers have introduced hypernetwork models across a variety of challenging deep learning tasks, including neural architecture search [110, 111], natural language processing [112, 113], computer vision [114, 115], and predictive 3D reconstruction [116-118]. Implicit neural representations from hypernetworks have become particularly compelling for 3D reconstruction tasks, demonstrated by the image-to-surface model developed by Gidi Littwin and Lior Wolf in 2019. Within this research, the hypernetwork f is a Convolutional Neural Network (CNN) that takes an image as its input, before predicting a tensor of weights. The primary network g is a simple multi-layer perceptron (MLP) network, with its weights parametrised from the hypernetworks output tensor. The primary network takes a series of coordinates as its input, with two output nodes defining whether or not the coordinate exists within the desired 3D shape. Within this implementation, Categorical Cross-entropy loss was used to backpropagate this error throughout the hypernetwork. Once trained, the primary network is extensively sampled to create a point cloud of the predicted geometry, which is then represented as a voxel model with a resolution that is defined according to the sampling grid size. The marching cube algorithm was then implemented to convert the sampled grid of points into a polygon mesh for rendering [119].

Hypernetwork models have since presented compelling opportunities within the academic field of 3D reconstruction, overcoming the significant resolution limitations of conventional approaches. Previously, voxel-based models required the design space to be divided into a three-dimensional grid of cubes. The state of each

cube within the grid is defined as either existing within the final output geometry or outside of it, framing the task of 3D reconstruction as a series of structured binary classifications. Consequently, the output resolution of voxel approaches is directly bound by the chosen grid format, with typical implementations using grid sizes of 32^3 [120-122], 64^3 [123, 124], or 128^3 [125]. The cubic relationship between grid resolution and memory usage has severely hindered the viability of training large voxel models at higher resolutions. Furthermore, these structured grid representations assume a fixed rectilinear output format that is inappropriate for many applied design tasks. In contrast, the implicit neural representations utilised by hypernetworks offer a theoretically infinite output resolution and complete freedom of topological form. Implicit neural representations such as signed distance functions (SDF) capture complex manifold geometries in a coordinate-based manner, rather than directly relying on explicit geometric representations such as polygonal meshes, point clouds, or cubic grids. The signed distance value represents the shortest distance between a given point and the surface of the underlying geometry, with coordinates along its surface providing a SFD of zero. Points existing outside of the target geometry record a positive SDF value (SDF>0), while points existing within the shape are assigned a negative SDF value (SDF<0) [126].

2.10.4. Convolutional neural networks

The convolutional neural network was first trained using backpropagation by Yann LeCun in 1989 to recognise handwritten numbers from 32 × 32 grayscale images, setting the foundation for the field of computer vision [127]. This was a groundbreaking demonstration of convolutional models with real-world potential applications directly observed at the time throughout the banking and postal sectors. However, due to the computational memory requirements of these convolutional operations and the hardware limitations of the time, research involving CNN architectures stagnated. Years ahead of its time, it would not be until the development of AlexNet in 2012 that the convolutional architecture re-emerged as a competitive and computationally viable approach [128]. Within the built environment, convolutional neural networks have been broadly applied to solve a diverse range of complex tasks, including structural damage detection [129], safety monitoring for on-site activities [130], floor plan analysis [131], and urban planning [132].

The innovation of the CNN lies in its filter-based processing of structured information, isolating local regions of an input signal using the same set of shared weights. By sharing the weights of a filter, the number of parameters required to train the model is significantly reduced, improving the computational efficiency of the network. These trainable filters are often referred to as kernels, which each detect specific features and patterns present within the input data. Each convolutional operation executed as the filter window slides generates a feature map which targets key hierarchical features of the input signal. By isolating structured subregions of the input image, convolutional neural networks are capable of learning nuanced details and patterns that collectively form high-level features [133]. The convolutional filter is a 2D matrix, with a size $k \times k$ in which k is referred to as the kernel size. The element wise multiplication between the weights of the filter and the isolated patch of

the input signal is calculated, before the resulting products are summed to output a single value into the next feature map. The step size of the filter as it slides across the structured input signal is referred to as the stride. For CNN models that use multiple filters in a given layer, the resulting feature maps are stacked, highlighting distinct patterns within the signal. Non-linear activation functions are used to process these feature maps, allowing the network to model complex non-linear behavior.

CNN architectures typically consist of three distinct layer types; convolutional layers which apply sliding filters to local regions of the input image, pooling layers which down sample local regions to reduce to spatial dimensionality of the data and distil input signals into their most salient features, and fully connected layers which are typically positioned as the last layers of the CNN for classification tasks. Deeper convolutional layers within the CNN model typically learn to recognise complex patterns such as edges or semantically meaningful objects [134]. The pooling layer typically utilises a max pooling or average pooling operation to distill the input signal. The max pooling function retrieves the maximum value within the current subregion, while the average pooling calculates the average subregion value.

2.10.5. Transformers

The task of understanding and predicting sequential text-based information has been a challenge that has captivated researchers for decades. In 1902, Andrey Markov introduced the concept of the Markov Chain and the Markov Decision Process, in which the next element of a sequence was predicted by only using the past elements provided within the chain. Eleven years later, Markov directly applied this concept to a 20,000 letter text exert from the Russian novel *Eugene Onegin* [135]. Within this lecture, Markov demonstrated that it was possible to predict future letters within the sequential letter-based chain — a feat he achieved without the assistance of a computer. Markov's field-defining theory remains a common foundation for many deep learning systems to this day, including recurrent and self-attention-based models.

The motivation behind the transformer architecture is best understood within the context of its predecessors. The concept of the Recurrent Neural Network (RNN) and memory-based models was introduced by John Hopfield in 1982, in which the network retains previous states as a form of persistent memory. The discrete points within these Hopfield networks are capable of storing information, marking a significant departure from conventional perceptron networks of the time [136]. Additional breakthroughs in recurrent architectures were achieved through the creation of Long Short-Term Memory models (LSTM) in 1997, which introduced gated memory cells that retained or forgot information across input sequences. The LSTM model uses these hidden states by passing them from one time step to the next, allowing previous context to sequentially impact future predictions [137]. These memory cells enabled long-term dependencies to be effectively trained without

significantly suffering from exploding or vanishing gradients over the long input sequences required for natural language processing. It was this memory stability and the introduction of temporal relationships that laid the foundation for the field of NLP to rapidly develop in the coming decades.

In more recent years, recurrence has been overshadowed by the concept of self-attention, applied within the context of large transformer language models. Modern self-attention transformer architectures were introduced in the seminal paper 'Attention is All You Need', written by Google researchers [138]. Immediately, transformer-based models outperformed cutting-edge Natural Language Processing models that had previously dominated the field. Instead of relying on the quality of the final hidden state, these models allow each token to 'attend' to every other token. Self-attention transformers utilise positional encodings to inject order information rather than relying on sequential processing and recurrence to infer positional information. This eliminates the sequential processing of input data, which allows self-attention computations to be fully parallelised. The distribution of these operations rapidly speeds up the training and inference processes, providing a scalable method for researchers to build larger models that would not have been practical using traditional recurrent methods.

Broadly speaking, transformer architectures may be broken down into encoder and decoder modules, which each serve distinct purposes depending on the chosen deep learning task. The encoder region of the model is responsible for taking input data and converting it into a sequence of contextually rich high-dimensional vectors. Transformers that exclusively utilise encoders are often uses as embedding models in practice, such as the Bidirectional Encoder Representations from Transformers (BERT) architecture. The decoder module is responsible for processing these contextually informed representations and the current output sequence, generating the next token of the sequence. By combining these unique phases of the transformer architecture, these models are capable of performing sequence-to-sequence tasks such as language translation [139], text summarisation [140], speech recognition [141], image captioning [142], conversational inference [143], and code generation [144].

Transformer-based architectures have been deployed with remarkable success within deep learning fields beyond Natural Language Processing, outperforming traditional convolutional neural networks on computer-vision related tasks with large datasets. When pretrained, vision transformers (ViTs) can achieve higher accuracy and greater generalisation capabilities when benchmarked against traditional CNN-based architectures for image classification tasks [145, 146]. This is in part due to the capacity of transformer-based networks to analyse long-range relationships throughout each input image, as opposed to the localised frame-based computations provided by convolutional approaches. The small kernels used by typical convolutional neural networks are

designed to aggregate local relationships within the input image, with the receptive field increasing towards the deeper layers of the model. Relevant features that are spaced further apart within the image will therefore require deeper network architectures for their relationships to be learnt efficiently. This raises a significant scaling issue when attempting to train larger networks for more complex deep learning tasks. In contrast, ViTs divide the input image into a series of patches that are processed simultaneously by the self-attention mechanism, providing context for each learnt feature relationship. As a result, transformer architectures can be significantly more computationally efficient when scaling to large architectures [147]. Additionally, while the fixed kernel size utilised by convolutional neural networks may be an effective form of structural bias that benefits smaller training datasets, it is often outperformed by the self-attention mechanism when scaled to large training datasets with complex spatial relationships.

2.10.5.1 Embeddings

Before processing the input sequence for NLP tasks, the text must be converted into a tokenised form, conventionally as a series of integers. Each input token passed to the transformer model must then be converted into a vectorised representation, which had a length of 512 in Google's original paper. This is significant, as the embedding technique offers some immediate contextual information to the transformer model before training has even started. Positional encoding must also be introduced at this stage as the transformer does not inherit the order of the input, unlike sequential models such as LSTM networks. These positional encodings must be applied across each word embedding vector. Sine and cosine positional encodings are typically used, providing a range of possible frequencies across each token position and each position along the vector. For each embedding vector for a given input token, an equally sized unique positional encoding vector is generated. This information is introduced into the input signal through the following positional encoding equations:

$$\mathrm{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.31}$$

$$\mathrm{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.32}$$

where pos is the integer position of the given token within the input sequence, i is the dimensional index of the embedding vector, and d_{model} is the length of the output embeddings. A large constant scalar hyperparameter is provided to space out the wavelengths of the sine and cosine functions, which can be empirically tested based on varying input sequence lengths. The intuition behind using a sine and cosine function for the positional encodings is to provide a smooth and continuous encoding signal that can be easily learnt by downstream training, allowing the transformer to learn relative positional relationships without explicitly storing them. Within the original implementation, the vectorised embeddings and the positional vector were simply added together. One challenge with this methodology is that there is a resulting loss of information that can arise through the direct combination of these vectors. To overcome this, a scalar multiplier is sometimes used to adjust

the word embedding vector. Notably, the embedding and positional encoding process is only conducted once for a given input sequence and is not repeated beyond the first layer.

2.10.5.2 Self-Attention

The purpose of the self-attention module is to capture the semantic relationships between words within an input sequence. Within the self-attention operation, there are three key representations that are required: the query, key, and value vectors. When operating at the sequence level, multiple vectors are combined to form the respective query (Q), key (K), and value (V) matrices. The query vector is designed to represent the contextual information requirements that must be addressed to make sense of the current token. It may be interpreted as a question that the current token is asking about its context within the input sequence. The key vector represents the answers other tokens can provide to the given question, such that the dot product between the query and key vectors represents the relevance of each key to the current query. This dot product operation provides the attention scores, with higher values representing stronger attention connections between tokens. The value vector contains information that is passed to other tokens if they are weighted via the attention mechanism. In this sense, the value vector contains the token representations that are contributed to the final representation of other tokens. The attention scores are calculated as follows:

$$\text{Attention}(Q, K, V) = softmax\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{2.33}$$

where $K^{\top}$ is the transpose of the key matrix and d_k is the dimensionality of the query and key vectors. The dot product between the query and key matrices is divided by the square root of the vector length to remove the impact of vector length on the output magnitude. The SoftMax function is used to transform these similarity scores into a normalised probability distribution, representing the contextual strength between tokens. By multiplying this probability distribution with the values matrix, the attention scores effectively scale the value vectors as a weighted sum. In this sense, the self-attention mechanism effectively weights the values of each token based on its relevance to the current one, generating a semantically rich output representation for each token. Each new representation captures relevant contextual information from other tokens provided within the sequence. These new information-rich outputs are then passed on to the next layers of the transformer architecture, which often include residual skip connections, normalisation, and feed-forward network processing. In the original implementation, the fully-connected network that follows the attention module contains two layers with 2048 neurons, ReLu activations, inputs with a length of 512, and vector outputs with a length of 512.

2.10.5.3 Multi-head attention

By adding multiple heads to the self-attention operation, a range of different relationships may be learnt across the input sequence. Once an input signal has been tokenised, vectorised, and has had positional information added to it, it is then passed on to the multi-head attention module within the first encoder layer. Each vector is broken up and divided equally across multiple heads, allowing each head to capture a unique understanding of the relationships present within the input sequence. This process furthermore significantly speeds up computational processing, as these operations can be executed in parallel. In the original Google paper, each vector of length 512 is divided across 8 heads, such that each head was responsible for vectors with a length of 64. The output from each of the heads is then concatenated back together after the self-attention operation, restoring the original shape of the input. In this sense, the final output is a concatenation of the outputs from all attention heads into a single unified representation. Extensive research has demonstrated that multi-head self-attention works more effectively than simply increasing the size of a single attention head from a computational efficiency and performance perspective [148, 149].

2.10.5.4 Encoders and decoders

Each encoding layer contains two sub-layers; the self-attention module, and a fully connected network. The encoder module of a transformer model passes the signal through a multi-head attention block, before being processed by a fully-connected feedforward network. A residual connection is used to allow a copy of the input signal to bypass the self-attention operation, retaining any positional encoding information that may be obscured in the process. The attention mechanism itself is a tool to capture the relationship between each word and the other words within the input sequence. These semantic relationships are critical to understanding the meaning of words within the context in which they are used. Attention achieves this by operating dot products between vectorised words to identify strong relationship connections present within the input sequence. These attention mechanisms are also run in parallel through a multi-head arrangement, speeding up training and allowing each head to learn unique semantic connections within the input data. The decoder consists of several stacked layers, each with the same principle structure. Each decoder layer contains three sublayers: a masked multihead self attention region, multihead cross-attention, and a feed-forward fully connected region. The output from the final linear layer is typically passed to a SoftMax function, providing the probability distribution for the next predicted element. Each sublayer is followed by residual skip connections and a layer normalisation, before being passed to the next decoder layer. The probability distribution over all tokens within the vocabulary is then sampled to generate the next token in the sequence, before being fed back into the decoder for the next step in the sequence.

2.10.6. Reinforcement Learning

One challenge that researchers often face within the field of supervised learning is that the collection, labelling, and standardisation of input data is an incredibly time consuming and resource intensive task, particularly for large-scale research projects. As the performance of neural networks is directly proportional to the quantity and quality of input data, this obstacle is increasingly prevalent within academia and industry. In contrast, reinforcement learning (RL) is a field of machine learning which leverages observations collected by a decision-making 'agent' operating within a simulated environment, largely eliminating the need for a curated and labelled training dataset. The behaviour of the agent within this environment is evaluated according to an accumulative reward system, in which actions that produce desired outcomes are reinforced via a defined reward signal, while violations of constraints are punished. Reinforcement learning models are trained by accounting for the rewards an agent receives for each action taken during the simulation, with the goal of learning actions which correspond to optimal performance for given environmental states.

Conventionally, the environments that are constructed for the purposes of reinforcement learning operate under the Markov property, such that *"the future is independent of the past given the present"* [150]. That is, if all necessary observable variables are available for the current timestep, the entire path of historic actions are not required to make an informed action. Each sequential 'state' (S) of the environment is observed by the agent, before an action (A) is taken and the corresponding reward (R) is determined. Hence, the process of determining the optimal action given an observed state is fundamental to the process of reinforcement learning. These elements constitute the Markov Decision Process (MDP), which form what is referred to as a 'trajectory' when recorded sequentially over time.

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots \tag{2.34}$$

2.10.6.1 Proximal Policy Optimisation

Within the context of Proximal Policy Optimisation (PPO), the 'policy' refers to a neural network which receives the observed states from the simulation environment as its input. The output from the policy neural network is a prediction distribution over all possible actions that can be made by the agent within the environment. An additional value network may be used to estimate the discounted sum of rewards, or the 'state-value' function. The advantage $\widehat{A_t}$ that estimates the relative performance of a given action can be determined by subtracting the state value function $V(s)$ from the action value function $Q(s, a)$. The advantage itself estimates the performance of the actions taken during an episode relative to the expected performance. If the advantage is positive, actions taken in the simulated environment yielded a greater reward than the expected baseline, resulting in an increased probability that the same actions will be selected in future analogous states. Conversely, negative advantage values indicate that the actions yielded a result that was worse than the expected baseline.

Proximal Policy Optimisation introduces the concept of the clipped surrogate objective function, in which the hyperparameter ε controls the proxy by limiting the probability ratio $r_t(\theta)$, incentivising the policy to remain within the bounds [1 - ε, 1 + ε]. The Proximal Policy clipping function which regulates modifications to the underlying policy is as follows:

$$L^{CLIP}(\theta) = \widehat{E_t}[min(r_t(\theta)\,\widehat{A_t}\,, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\,\widehat{A_t}\,] \tag{2.35}$$

The expectation operator $\widehat{E_t}$ indicates that this optimisation function is run over multiple batches of trajectories. The clipped and unclipped objectives are first compared and the minimum is taken, allowing for a conservative modification of the policy when the probability ratio extends beyond the clipped threshold. By developing a clipped objective function, PPO allows for controlled limitation of step sizes that can safely navigate complex, high-dimensional optimisation topologies.

Within the family of gradient-based methods for reinforcement learning, Proximal Policy Optimisation (PPO) is innovative through its integration of clipped surrogate modelling. While conventional policy methods typically conduct a single pass of gradient modification per collected data point, PPO allows for the use of a collection of mini-batches over multiple epochs. This is critical, as policies that are updated using a single sample of data will vary significantly across each pass, introducing instability to the training process. Within PPO, the agent-observation and policy optimisation schemes are alternated between frequently during the training process, improving the stability and reliability of the model over iterative policy updates. Initially, the policy $\pi_{\theta old}$ is implemented for the simulated environment over a predetermined T number of timesteps. The advantages estimates $\widehat{A_1}$, ... , $\widehat{A_t}$ are then calculated, before gradient descent is executed on the policy network over K epochs. Observations within these mini batches may be distributed across multiple agents, interacting within separate instances of the same virtual environment. Finally, the training objective may be calculated according to the following function. which is approximately maximised at each iteration:

$$L_t^{CLIP+VF+S}(\theta) = \widehat{E_t}[\,L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)] \tag{2.36}$$

The baseline network is updated by the L_t^{VF} squared error loss term, which is scaled by the c_1 coefficient. Explorative behaviour is induced within the policy through the entropy term S, which measures the uncertainty of the existing policy at the given state. The coefficient c_2 is also a hyperparameter which scales the influence of entropy on the objective function [151].

The pseudo code for the Actor-Critic implementation of Proximal Policy Optimisation is as follows:

Proximal Policy Optimisation

1. **for** iteration = 1, 2, … iteration limit
2. **for** actor = 1, 2, … N
3. Run environment for T timesteps with policy $\pi_{\theta old}$, or until stop condition
4. Calculate advantage estimates $\widehat{A_1}$, ... , $\widehat{A_t}$
5. **end for**
6. Optimise surrogate L with respect to θ
7. $\theta_{old} \leftarrow \theta$
8. **end for**

2.11. Conclusions

This systematic review of modern machine learning systems and metaheuristic algorithms reveals significant research gaps that have largely remained ignored throughout the available literature. By addressing these gaps, new multidisciplinary solutions have significant potential to mobilise technological adoption and innovation throughout unique phases of the asset lifecycle, ranging from preliminary structural design to long-term structural health monitoring. From a design perspective, the performance of leading metaheuristic optimisation algorithms are known to be highly sensitive to the initialised hyperparameter sets, which are static in conventional research applications. The capacity of neural networks to adaptively control these critical parameters throughout the optimisation process has not yet been investigated beyond the limited context of Q-learning. This research gap is addressed in section 4.2, which documents the use of proximal policy optimisation to train a deep neural network tasked with mediating the exploration-exploitation response of the differential evolution algorithm.

Analogously, the design of complex machine learning models remains a deeply challenging task that is often plagued by suboptimal user defined model hyperparameters and computationally inefficient iterative manual tuning. Remarkably, despite the rising demand for these models to be efficient enough to run on consumer-grade hardware, very little research has been published within the field of neural architecture search which priorities model memory footprint and inference resourcefulness. Limited computational budgets within applied industry contexts present a significant barrier that directly inhibits the capacity for practicing engineers and architects to experiment with cutting edge machine learning tools. These practical concerns are investigated in section 4.3 through the development of the MEMODE algorithm, which automates hyperparameter optimisation for complex multimodal machine learning models using differential evolution while penalising memory inefficiencies.

The vast quantities of data that are required to effectively train these models also present a unique challenge that researchers within the field of 3D reconstruction have struggled to reconcile. Currently, very little practical real-world structural data is available for machine learning practitioners who are interested in developing predictive 3D design models. It may be argued that this open source 'data-drought' has driven the prevalence of relatively simple object-based datasets such as ShapeNet and PartNet, which do not provide the degree of scene complexity necessary for practical design tasks. Consequently, the field of predictive design has struggled to meet the needs of industry practitioners. This presents a unique opportunity to investigate the viability of using metaheuristic algorithms to develop high-quality design databases with consistent topological forms that are highly suited for training modern machine learning models. The demonstrated versatility of metaheuristics within the field of structural optimisation suggests their potential as a reliable and consistent data generation strategy. Section 5.2 examines the application of metaheuristic algorithms to support the training of novel 'text-to-structure' hypernetwork models, providing a framework for further research to overcome the pervasive challenge of data acquisition. Section 5.3 provides a transformer-based solution to scale the proposed 'text-to-structure' hypernetworks to bridge the vast gap in output complexity that currently exists between academia and industry.

Finally, despite meaningful technological advances within the field of structural health monitoring, existing sensor-based research approaches such as FBGs are typically severely limited to discrete and low-dimensional forms of analysis. Consequently, these systems fail to adequately capture the entire spectrum of structural behavior that a monitored asset is subject to, directly limiting the interpretability and value that these solutions provide to asset managers and engineers. Simultaneously, these sensors inevitably collect enormous structural datasets that are inherently challenging for asset managers to parse through and extract meaning from. These challenges cast doubt on the sensor-based SHM cost-benefit proposition that many researchers currently advocate for. Chapter 6 seeks to amend this by developing comprehensive, memory-efficient, spatially rich, high-resolution machine learning models for real-time asset monitoring. The proposed hypernetwork model documented in section 6.3 radically improves the representation capabilities of modern SHM systems, aiming to further encourage the broader adoption of FBG sensor technologies and improve asset maintenance procedures within the context of a practical real-world project.

CHAPTER 3: RESEARCH FRAMEWORK

3.1. Summary

When considering a suitable framework to explore the use of metaheuristics and machine learning models for civil engineering tasks, it must be noted that there has historically been a significant gap between published research and practical design applications within industry. As a consequence of the highly specialised domain that these fields of research occupy, the scope of tools developed within academia are typically highly specific. Research projects typically do not require broader adoption from existing teams that may significantly benefit from the insights or processes that the research proposes. As such, tools developed through academic research typically do not have an emphasis placed upon shared usability or practical implementation within industry. Often this is by design, as practical tools require a significant amount of additional out-of-scope work to make them suitable for industry practitioners. While there is significant value in providing a research environment that is detached from these requirements, this thesis argues that the stagnant productivity of the construction sector requires research that mobilises technological adoption. Without research specifically aiming to bridge the ever-growing rift between academia and industry, the true value provided by these tools is unlikely to be fully realised.

3.2. Overview of research scopes

This thesis is comprised of six research scopes that are centered around the application of novel computational methods at the intersection of metaheuristics and machine learning, with the objective of advancing industry practices throughout the built environment. While the scope of these sections are individually focused on developing novel computational solutions at targeted phases of the asset lifecycle — ranging from improved algorithmic performance for preliminary structural design to predictive SHM systems with unprecedented resolution capacity — they collectively contribute to a deeper understanding of how metaheuristics and modern ML models can inform and optimise decision making in architectural design, civil engineering, and structural health monitoring. Underlying each of these chapters is the assertion that there exists a significant gap within the available literature regarding the development of novel computational models at the intersection of these fields, particularly when investigated within the context of the built environment.

The first research scope, section 4.2, investigates the capacity for neural networks to mediate the algorithmic behaviour of metaheuristic algorithms during the optimisation process of steel frame structures. Proximal policy optimisation (PPO) has been implemented as a reinforcement learning strategy to train a deep neural network to control the exploration-exploitation response of the Differential Evolution (DE) algorithm. Remarkably, the hybrid algorithm developed through this research outperformed conventional genetic algorithms, particle swarm optimisation, and static differential evolution approaches across each of the benchmark engineering tasks. This

improved engagement with the exploration-exploitation trade-off extends beyond the novel hybrid algorithm proposed within this research, suggesting the potential application of leading reinforcement learning methods such as PPO to adapt the operators and hyperparameters for algorithms across the metaheuristic family. As a hybrid implementation of foundational metaheuristic approaches and cutting-edge machine learning techniques, this section serves as a compelling precedence that inspired the chapters that followed.

The second research scope, section 4.3, '*Optimising memory-efficient multimodal networks for image classification using differential evolution*', examines the opportunities that emerge when this symbiotic relationship between metaheuristics and deep learning is reversed. Instead of training deep neural networks to mediate the optimisation behaviour of the differential evolution algorithm, this paper applies DE to generate optimal network architectures for memory efficient model design. Importantly, the training of modern deep learning models requires the initialisation of variable sets and architectural parameters that can have a profound impact on prediction performance. Conventional methods of manually tuning these hyperparameters often require '*a priori*' knowledge and extensive experimentation that is time-consuming and unlikely to yield designs that are proximate to globally optimal performance. This problem has become exacerbated as modern cutting-edge architectures have become larger and increasingly complex, particularly for multimodal models that are tasked with reconciling multiple streams of input while maintaining acceptable compute efficiency for training and inference. The model that was evolved through the novel differential evolution algorithm developed within this paper outperformed over 1,600 competing model architectures, including ResNets, inception models, and models that were optimised using particle swarm optimisation, the genetic algorithm, and Bayesian optimisation. These research results demonstrate the competitive performance of the Differential Evolution algorithm for neural architecture search and hyperparameter optimisation, reaffirming the central premise of this thesis: that metaheuristic algorithms and machine learning models possess complementary qualities that have the capacity to enable new research opportunities and deliver practical value for design and decision making. This section has been published within the *Journal of Applied Soft Computing*, Volume 171, March 2025.

The third research scope, section 5.2, '*Transformer Hypernetworks for Text-To-Structure Design*' expands on the complementary opportunities between metaheuristics and deep learning models within the context of predictive 3D reconstruction. The hypernetwork method — which was inspired by the radically novel HyperNEAT algorithm — has been applied within this chapter to predict structural solutions to a range of tasks according to user defined input parameters. The design tasks explored within this section range from simple 2D truss structures to 3D girder bridges with interconnected and distinct structural meshes. This marks the first application of the hypernetwork method within the field of structural design, introducing a powerful 3D reconstruction tool for researchers and industry practitioners that is theoretically not limited by topology, resolution, or dimensionality. These hypernetworks were driven by training datasets that were generated using

a metaheuristic approach, allowing historic collections of optimal structural designs to directly inform the predicted hypernetwork outputs. This novel contribution provides a remarkably effective and computationally efficient methodology to build 3D reconstruction datasets with a clear design language and learnable patterns. Additionally, the implemented 'Text-to-structure' framework adopts an interactive approach to structural prediction, allowing users to provide text-based input prompts that contain the necessary design parameters such as the nodal boundary conditions, element span, loading conditions, and structural form.

The fourth research scope, section 5.3, '*Transformer Hypernetworks for Complex Structural Representation*' introduces innovative strategies to scale the representation capacity of these foundational reconstruction models to meet the complexity of applied engineering and architectural design tasks. One of the most significant contributions of this paper involves the training of novel transformer encoder models, which were tasked with predicting a series of translation vectors that manipulate structural geometries within the normalised design space. These vectors effectively allow base structural meshes such as foundations, columns, beams, bracing elements, and floors to be replicated and transformed — constructing intricate and complex structural systems with unprecedented representation capacity. Previous research applications of foundation 3D reconstruction models have conventionally been limited to single asset geometry datasets such as ShapeNet or PartNet. These academic datasets are not illustrative of the scale or complexity of modern design tasks which may consist of hundreds or even thousands of interconnected structural components. To the author's knowledge, prior to this paper there were no demonstrated solutions that could scale these foundational models to meet the practical demands of the AEC industries. This may in turn explain the significant lack of research applying hypernetworks and other foundational reconstruction models within the context of applied architectural design and structural engineering. The results and design visualisations provided by this paper mark a radical shift in reconstruction output for these models, demonstrating compelling performance for high-rise, bridge, and steel-frame structural topologies. This section has been published within the *Journal of Computing in Civil Engineering*, Volume 40, September 2025.

The fifth research scope, section 6.2, '*A neural network based digital twin model for the structural health monitoring of reinforced concrete bridges*', investigates the application of deep neural networks to improve the predictive capabilities of sensor-based structural health monitoring systems. Within the broader context of this thesis, this paper outlines the opportunities for novel computational methods to enrich decision making during the operational phase of an asset's lifecycle, while reflecting on the limitations of existing structural health monitoring practices. Traditional methods of structural health monitoring such as non-destructive evaluation (NDE) are severely limited to routine visual examinations that are highly subjective, and unable to provide continuous information into the long-term behaviour of monitored structures. Modern integrated sensor systems that are designed to capture this long-term structural behaviour have emerged as extremely promising technological advanced within the field of SHM, however the vast structural datasets that they produce are

currently extremely challenging to parse, analyse, and extract value from. In order to revolutionise decision making for asset managers and transition away from insufficient forms of non-destructive evaluation, solutions must be proposed that can allow users to extract actionable insights from these enormous datasets. Neural networks were introduced within this paper to predict the strain response of a monitored footbridge structure as an interactive visualisation, allowing historic discrete sensor data to be translated into comprehensive heatmap simulations. A bespoke user interface was also designed within this project to provide asset managers the ability to parse through the complete historic response of the structure, including the predicted intermediary strain values. This interface furthermore provided a geometric context to the structural analysis, as an entirely novel interactive solution for asset owners. This research serves as a precursor for the compelling hypernetwork approach that was developed in the final research scope of this thesis, which explores novel deep learning architectures for rapid SHM inference and significantly improved representation capacity. This section has been published within the *Journal of Structures*, Volume 57, November 2023.

Finally, the sixth research scope, section 6.3, '*Hypernetworks for Real-Time Structural Health Monitoring of bridges using Embedded Fibre Bragg Grating Sensors*' utilises the precedence established by these previous papers to extend the resolution capacity of modern structural health monitoring systems to new bounds. This was achieved through the development, training, and inference of a novel hypernetwork model, tasked with predicting the strain response of a monitored footbridge at any point within the structure from discrete strain readings recorded by embedded fibre optic sensors. This research demonstrates the unprecedented predictive capabilities of hypernetworks for SHM, outputting implicit neural strain heatmaps with theoretically infinite resolution and rapid inference speeds. In contrast to the broader field of sensor-based structural health monitoring that has become burdened by 'Big data' and an overreliance on low-dimensional forms of analysis, this research introduces a method of leveraging the full collection of historic sensor records for the prediction of high-resolution, context-aware strain heatmaps. While previous research approaches of extracting meaning from modern sensor systems have ignored the geometric component that SHM has historically relied upon, the concept of 'visual inspection' has been directly incorporated into the interactive heatmap visualisations that this thesis has introduced. As a result, this chapter aims to ground itself on existing industry practices, while incorporating cutting edge sensor technologies and deep learning models to empower asset managers. This is directly in service of the underlying objective of this thesis: to bridge the gap between academia and industry that has historically limited the adoption of innovative technologies and hindered productivity. This final research scope serves as a capstone to this thesis, revealing the unexplored potential for interdisciplinary machine learning models and black-box algorithms to reshape modern approaches to our built environment. Each of these research sections was developed to question existing approaches and dispel established research limitations, with the overall objective of reducing the vast gaps between academia and industry.

CHAPTER 4: ALGORITHMIC DESIGN WITH SELF-ADAPTIVE DIFFERENTIAL EVOLUTION STRATEGIES AND MEMORY EFFICIENT NEURAL ARCHITECTURE SEARCH

4.1. Introduction

This chapter documents the development of hybrid algorithms that aim to address key challenges that metaheuristics and machine learning methods inherit by adopting an interdisciplinary approach to algorithmic design. Despite inherent challenges that have limited their adoption within the applied context of the built environment, metaheuristic algorithms and machine learning models have a profound capacity to revolutionise the workflows of practicing architects, engineers, and asset managers. From a design perspective, metaheuristics provide an agnostic black-box approach to navigate complex high-dimensional search spaces and converge at high quality parameter sets. However, the convergence performance of these algorithms is highly sensitive to the initialised hyperparameters, requiring a degree of experimental knowledge and optimisation experience to implement effectively. Adaptive strategies such as Q-learning and dynamic parameters have been proposed to reduce these prerequisite requirements, however the potential for cutting edge machine learning models to guide the metaheuristic exploration-exploitation response is yet to be fully explored. Reinforcement learning techniques such as proximal policy optimisation offer a unique opportunity to extend the observation and action space of agents within these adaptive systems beyond discrete constraints.

Similarly, the performance of complex neural systems is severely impacted by the experience of the model designer. Suitable hyperparameter sets and architectural arrangements are extremely challenging to tune manually, particularly when designing models that are tasked with processing multiple input streams under a single unified multimodal arrangement. This provides a compelling application for metaheuristic algorithms to automate the selection of critical network features such as model depth, layer size, learning rate, and dropout regularisation. To align this research with practical industry hardware limitations and mobilise technological adoption, this opportunity has been framed within the context of memory-efficient multimodal model optimisation, providing a framework for researchers and practitioners to automatically build high-performing, low-footprint machine learning models. Importantly, the comparative benchmarking analysis provided within this chapter has been used to inform the implementation of the differential evolution algorithm throughout the successive chapters within this thesis.

4.2. SELF-ADAPTIVE DIFFERENTIAL EVOLUTION WITH PROXIMAL POLICY OPTIMISATION FOR STRUCTURAL DESIGN

4.2.1. Summary

Within the field of structural optimisation, Differential Evolution (DE) has been applied as a powerful metaheuristic algorithm across a broad scope of diverse engineering design problems. The performance of DE is however predicated on the initialisation and selection of appropriate mutation operators and control parameters, which direct the exploration-exploitation behaviour of the algorithm. Conventionally, these selections are static in nature, often demanding an iterative approach when engineers attempt to deploy these algorithms in practice for structural design problems. The novel algorithm (DE_{PPO}) proposed within this research leverages Proximal Policy Optimisation (PPO) to adapt the mutation operators of the Differential Evolution algorithm throughout the generational loop of the optimisation process, creating a dynamic trade-off between exploration and exploitation that outperforms traditional static methods. To compare the performance of the self-adapting algorithm, DE_{PPO} was applied to five diverse truss optimisation benchmark tasks. Across all benchmarked optimisation tasks, the proposed method outperformed conventional metaheuristics such as the Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO) by an average of 22.27% and 12.53%, respectively. By intelligently alternating between mutation operators which are designed to target either explorative or exploitative behaviour, the self-adapting algorithm outperformed all static DE variations, GA, and PSO algorithms for all averaged benchmark tests.

4.2.2. Introduction

The application of metaheuristic algorithms for structural engineering problems has conventionally involved the optimisation of key design parameters that improve performance with respect to loading capacity, carbon footprint, material expenditure or cost. Research involving these algorithms has grown significantly over the past few decades, empowered by a collection of pin-joint truss structures that have frequently been used as benchmark design problems. The objectives, design parameters, constraints and loading conditions of these benchmark problems are well documented, providing a universal foundation from which novel algorithms may be introduced and evaluated [152]. Since its inception, the Differential Evolution algorithm (DE) has demonstrated exceptional performance within the context of these benchmark problems [153]. Created as an evolutionary population-based metaheuristic with a focus on global optimisation, DE inherits its fundamental structure from the genetic algorithm, leveraging mutation and crossover functions to navigate the search space. The innovations introduced within the Differential Evolution algorithm are its unique mutation equations, which are vector-level operations that control the chosen exploration-exploitation strategy. These unique facets of differential evolution have been contributing factors to the broader adoption of the algorithm, which has outperformed traditional metaheuristics such as the genetic algorithm [154], simulated annealing [155], and evolutionary programming [156] across a range of black-box optimisation tasks.

The performance of Differential Evolution is contingent on the selection and initialisation of its mutation operator, which traditionally produces a static exploration-exploitation response. This limitation has driven research in recent years to explore methods to adaptively control the parameters within the mutation operator, as well as the selection of an appropriate operator itself [157]. One of the first prominent algorithms created for Differential Evolution within this field is JADE, which introduced an adaptive mutation strategy that leverages historic simulation data to promote genetic diversity and drive convergence performance. This algorithm demonstrated an improved or comparable performance to traditional DE, without the need for hyperparameter initialisation to be dependent on user intuition and prior knowledge [158]. This was further explored through the development of the Success-History Based Parameter Adaptation for Differential Evolution (SHADE) algorithm, which stored historic sets of previously successful Crossover rates (CR) and scaling factors (F), to be sampled from in the creation of new combinations for successive generations. These adaptive strategies have since inspired a collection of bespoke modifications, such as the novel L-SHADE algorithm, which provides a linearly decreasing population size to improve search efficiency [159]. The evolution of these algorithms over time demonstrates the degree to which these initialised parameters impact the algorithmic performance of Differential Evolution, driving researchers to investigate novel adaptive variations to well-established metaheuristic algorithms.

More recent research within the field of parameter adaptation has begun to explore the application of reinforcement learning methods to train generalised exploration-exploitation strategies. This has been demonstrated through the introduction of Q-Learning within the generational loop of the DE algorithm, leveraged as a selection agent that dynamically changes the mutation operators and hyperparameters. Studies produced by Kizilay et al. [160], Huynh et al. [89], and Hu et al. [161] have explored the performance of these adaptive QL-DE algorithms for design applications ranging from truss topology optimisation to the fine-tuning of photovoltaic models for the conversion of solar energy, yielding compelling results when compared against traditional forms of differential evolution. The combination of powerful metaheuristic algorithms and reinforcement learning models was furthermore benchmarked by Huynh, Do and Lee, in which the crossover rate and scale factor hyperparameters of a differential evolution (DE) algorithm were directly controlled by a Q-learning model. Notably however, the limitations of Q-learning remain a considerable barrier that hinders the application of reinforcement learning for tasks relating to hyperparameter adaptation. Namely, while Q-learning is a powerful and simple tool, its applications are significantly limited when scaled to high dimensional or continuous environments [162]. Discrete action and state spaces are required to implement Q-learning, confining the adaptation of DE to a limited portfolio of actions.

This research project is targeted at developing self-adapting algorithms that parametrically adjust their own algorithmic behaviour, establishing a dynamic exploration-exploitation response throughout the optimisation

process. The novel self-adapting Differential Evolution algorithm that was trained through this research is capable of out-performing other metaheuristic optimisation strategies when applied to established structural benchmark tests, demonstrating a competitive exploration-exploitation DE response. Historic attempts to introduce self-adapting metaheuristics have relied upon Q-learning to adapt the hyperparameters during optimisation, which is severely limited to discrete and finite observation and action spaces. The introduction of PPO into the field of self-adapting metaheuristic algorithms overcomes the significant limitations of Q-learning itself through the potential for continuous action and observation spaces. The significance of this improved engagement with the exploration-exploitation trade-off within the generational loop of the DE algorithm extends beyond the novel self-adaptive algorithm proposed within this research, suggesting the potential application of leading reinforcement learning methods such as PPO to tune the operators and hyperparameters for algorithms throughout the metaheuristic family.

4.2.3. Proposed self-adapting algorithm

The algorithm developed within this section extends self-adaptive metaheuristic algorithms beyond the limitations of Q-learning by leveraging the capabilities of Proximal Policy Optimisation. This research project posits that the DE_{PPO} algorithm is capable of reconciling the trade-off between explorative and exploitative optimisation procedures. Within the proposed algorithm, PPO has been integrated within the generational loop of the Differential Evolution algorithm, in which the reinforcement learning agent takes a normalised observation as its input, before selecting an appropriate mutation operator that maximises the potential rewards that may be realised through future generations. In this sense, the model parametrically adapts the manner by which it navigates the search space in response to historic observations at distinct phases throughout the optimisation simulation. In order to evaluate the suitability of the proposed optimisation strategy, a series of comparative benchmark tests were conducted against leading metaheuristic algorithms, including Particle Swarm Optimisation, variations of the Genetic Algorithm, and conventional Differential Evolution. By adapting the genetic diversity and convergence rate of the population iteratively in response to these critical observations, solutions generated by this novel self-adapting algorithm exceeded the output from competing metaheuristic optimisation methods.

Agent-based reinforcement learning strategies leverage data collected from their environment in order to develop a strategy, or policy, to optimally collect rewards. This environmental data is referred to as an observation, which must capture the existing state of the environment. The proposed training process is illustrated within Figure 4.1, in which the two distinct environments for both optimisation and training are defined. Within the context of PPO, the feedback loop between observations and rewards are leveraged to train the underlying policy, improving the performance of the agent for future simulations. The proposed algorithm tasks the agent with mediating the mutation operator between explorative (DE/rand/1) or exploitative (DE/best/1) behaviour at each generation, defined according to the two outputs produced by the policy. The observation space used within this research includes the normalised generation count $\frac{t}{T}$ and a normalised count

of the number of generations that have been dedicated to exploitative behaviour throughout the current simulation $\frac{\lambda_{exploit}}{T}$. These observations have been designed to provide a generalised method that allows the policy to be applied to a diverse range of design problems without excessively overfitting to any given fitness landscape. By tracking the current generation relative to the computational budget, this policy was trained with the objective of dividing the simulation timeline into distinct behavioural phases, rather than being confined to a static mutation operator as defined by conventional DE. Furthermore, by providing the policy a historic metric for the degree of exploitative behaviour that has been selected during the simulation, the adaptive method has the capacity to mediate between mutation operators according to distinct phases within the optimisation process.

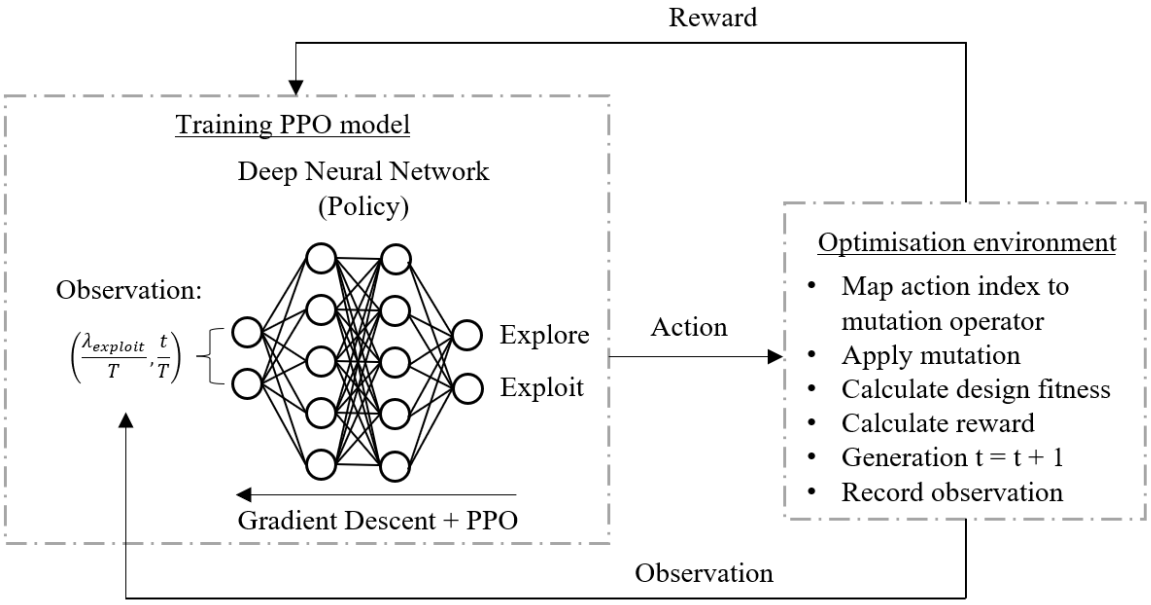


Figure 4.1 – Training of policy network within DE generational loop

In order to direct the learning process of the policy, a carefully designed reward signal is implemented to quantify the performance of the agent when actions are taken. These actions are directed by observations of the simulation state, captured by the agent during the optimisation process. The reward signal that has been created for the training of this reinforcement learning policy has a focus on convergence consistency, such that agents that are capable of reaching high-quality solutions across repeated optimisation simulations will be prioritised. A mass threshold β was introduced during training as a performance benchmark. The desired effect of this reward signal is to create a self-adaptive algorithm that outperforms traditional DE algorithms on average when run across a large number of simulations and separate benchmark tests. The reward for each timestep, according to the performance of the chosen mutation operator for a given generation, is as follows:

$$r_{t-1} = \begin{cases} i + \gamma i & \text{if } best_t < \beta \text{ and } i = I \\ i & \text{if } best_t < \beta \\ 0 & \text{if } best_t > \beta \\ -\gamma best_t & \text{if } best_t > \beta \text{ and } t = T \end{cases} \tag{4.1}$$

Within this context, the reward signal r_t is dictated by the mass of the best solution within the current generation $best_{t+1}$, the current generation count is recorded as t, and i represents the number of times the policy has successfully converged as a suitable design within the current environment. If the policy does not reduce the mass of the structure below the threshold β within the generational limit of $T = 200$ generations, the environment is terminated and a negative reward signal is provided that is proportional to the mass of the optimised structure $best_t$. Under this reward scheme, the policy is penalised significantly for converging prematurely at suboptimal solutions. If the best candidate within generation t satisfies the mass criteria β, a reward of i is provided and the optimisation loop is reset, allowing the agent to continue to collect rewards. In this reward scheme, the base reward signal is increased by one each time the policy successfully converges. Finally, if the mass criteria is satisfied and the optimisation loop has been reset I times, a final large reward is provided and the environment is ended, motivating future iterations of the policy to prioritise convergence consistency. For this research project, the number of resets was capped at 50, with the penalty and reward scalar γ set to 10.

Under the proposed training methodology, the underlying neural network policy has the capacity to take a diverse range of observation inputs and map them to a predicted action space. While conventional methods of adaptive parameter control which rely on Q-learning are limited to discrete environments and actions, the framework introduced within this research can be integrated into both continuous and discrete adaptive tasks. This furthermore removes the requirement of traditional Q-learning implementations to discretise the action and observation spaces, increasing the resolution of both potential actions and states for future research. While this research has focussed on the relationship between explorative and exploitative behaviour through the action output from the policy, the framework outlined in Figure 4.1 is also applicable to other parallel fields of metaheuristic research. By adapting the recorded observations, action mapping, and reward signals that have been used within this research, the policy behaviour itself may be shaped according to the interests of further research, with potential application within fitness-based adaptive methods, population controls, and adaptive metaheuristic algorithms. The underlying principles behind the training process outlined in Figure 4.1 and the deployed workflow of trained models illustrated in Figure 4.2 are broadly applicable beyond the scope of this research.

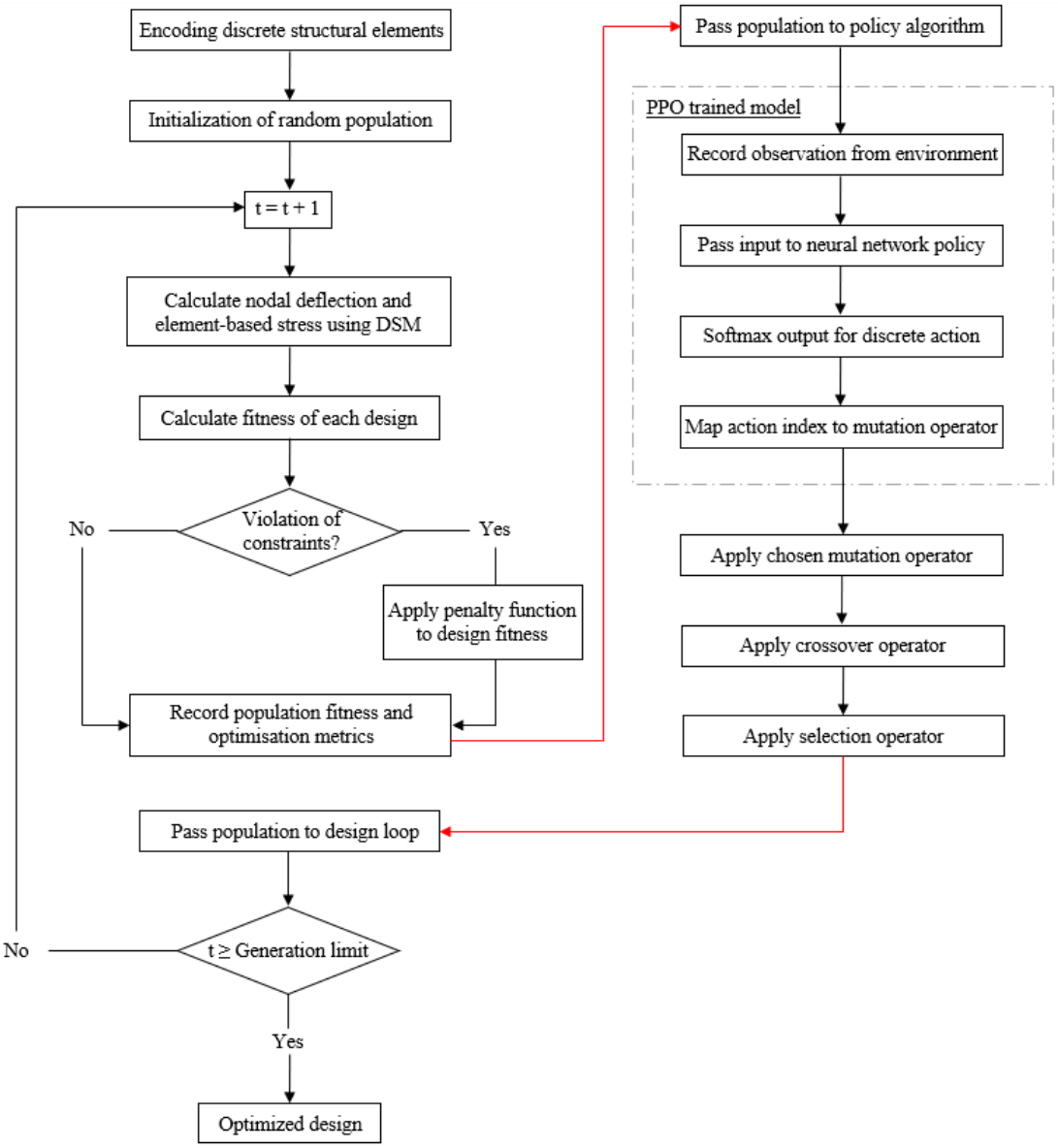


Figure 4.2 – The proposed self-adaptive Differential Evolution optimisation loop

Once trained, the proposed model integrates the adaptive capabilities of reinforcement learning within the generational loop. For each generation, designs within the population are analysed according to their nodal deflections and element stresses. Each solution is then evaluated against the design constraints for the given optimisation task, with a penalty function applied to the fitness of unsatisfactory designs. The observations are then recorded from the current optimisation environment, leveraged as the input to the policy network which in turn provides an action based on the predicted exploration-exploitation response. The action prediction is then mapped to the corresponding mutation operator, which is applied along with conventional DE crossover and selection operators to generate the next generation of designs. This generational loop is repeated until convergence is achieved, or the generational budget is exceeded. Within the proposed algorithm, the policy reward signal feedback loop is aligned with the differential evolution generational loop, such that agent performance is predicated on the quality of solutions generated within each population. The pseudo code for the

proposed implementation of the PPO trained policy to adapt the behaviour of the Differential Evolution algorithm is as follows:

Proposed Algorithm (DE_{PPO})

1. Initialise the initial population (P(t = 0))
2. Initialise trained policy network
3. Define the desired objective function
4. Evaluate the fitness of each member within the initial population $\left(P(t=0)\right)$
5. **while** generation limit has not been reached or convergence criteria is not satisfied
6. Record observations
7. Input observation to trained policy
8. Select mutation strategy based on output policy action index
9. **for** i = 1 to NP
10. Select random unique individuals $x_{R_1}, x_{R_2}, x_{R_3}$ ...
11. Calculate v_i^t using chosen mutation operator
12. $x_{off} = v_i^t$
13. **for** j = 1 to D
14. Generate *rand*(0,1)
15. **if** *rand*(0,1) ≥ CR
16. $x_{off,j} = x_{i,j}$
17. **end if**
18. **end for**
19. **if** $f\left(x_{off}\right) \leq f(x_i)$
20. $x_i = x_{off}$
21. **end if**
22. Update fitness record for x_i
23. **end for**
24. **end**

4.2.4. Benchmark tests

Truss design problems are perhaps one of the most common and widely used categories of benchmark testing throughout the field of algorithmic structural optimisation [163-165]. This is primarily due to the direct application of the analysis for industrial problems, the distinct design constraints, and the computational efficiency of these tasks. Conventionally, these benchmark tests seek to reduce the overall mass of the truss structure while satisfying one or multiple structural design constraints. These often include an evaluation of the induced stress, deflection, and buckling relative to a predetermined design criterion. Within the context of truss optimisation, these constraints dictate that is necessary for the tensile stresses induced within each member of the structure to be less than the maximum allowable tensile stress (σ_{max}), the induced compression to be less than the maximum allowable compressive stress (σ_{min}), and the nodal deflection to be less than the maximum allowable deflection (δ_{max}). Stresses and deflections calculated through either finite element simulations or the direct stiffness method during the optimisation process must therefore satisfy the following design constraints:

$$\sigma_{i,min} \leq \sigma_i \leq \sigma_{i,max} \tag{4.2}$$

$$\delta_j \leq \delta_{j,max} \tag{4.3}$$

For these design constraints, the tensile and compressive stress of element i is defined as σ_i , while the nodal displacement at node j is represented as δ_j. In order to relate this work to pre-existing literature and allow for further standardisation in future research, all truss benchmark problems and specifications that are explored within this research have been selected from existing benchmarks within the available literature [61]. Due to the stochastic nature of metaheuristic algorithms, it is also necessary to collect a representative sample of the performance of each optimisation technique, conducted multiple times across all benchmark tests. For reproducibility, the parameters of each benchmarked algorithm are summarised in Table 4.1.

The 10-Bar Plane Truss

This 2D frame problem consists of 10 bar members organised in a cross-braced truss structure with pinned supports and an applied loading force at the lower nodal interface of each bay. As a black-box optimisation test, this problem requires a given algorithm to adjust the cross-sectional area of each member in order to minimise the overall mass of the structure, while satisfying the defined design constraints. As the complexity of truss benchmark tests is typically defined by the number of elemental parameters available to the optimiser, the 10-bar plane truss problem has 10 dimensions that must be tuned. The broad range of potential solutions that can be generated from this scheme has driven the popularity of the well-known benchmark test to comparatively evaluate the performance of various optimisation algorithms. As a 2D finite element simulation, the computational cost of this benchmark test is relatively low, allowing for a large number of iterations to be executed by each solver within a manageable timeframe, which in turn facilitates the convergence of each algorithm for comparison. This benchmark test also consists of two commonly used loading conditions that have both been used within this analysis, requiring a load to be placed at nodes 5 and 6, respectively. Both load cases are subject to stress constraints, while the second test is also subject to a displacement limitation, the results from which are summarised in Table 4.3. Additional details for each benchmark task are provided within Appendix A of this thesis.

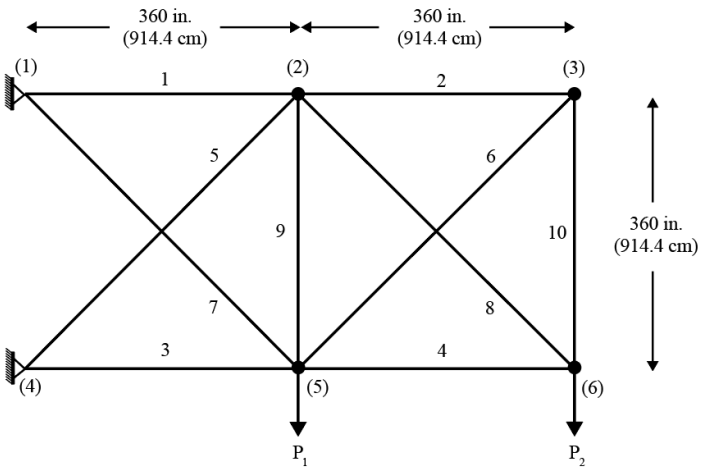


Figure 4.3 - The 10-bar plane truss

The 15-Bar Truss

The 15-bar truss problem is a discrete optimisation problem in which the cross-sectional areas of each bar are selected from a fixed set of 32 sizes. This optimisation problem allows for the position of non-boundary nodes to be adjusted in both dimensions within a constrained area, introducing both size and schematic optimisation. The structure is subject to a stress limitation, with a load applied at node 8. The Young's modulus for the elements is encoded as 10,000 ksi (68.95 GPa), with a material density of 0.1 lb/in^3. (2,770 kg/m^3) Within the allowable profile of cross-sectional areas that may be optimised, the minimum area is 0.111 in^2 (71.6 mm^2) and the maximum area is 19.18 in^2 (12,364 mm^2). The results for this benchmark test are summarised in Table 4.3.

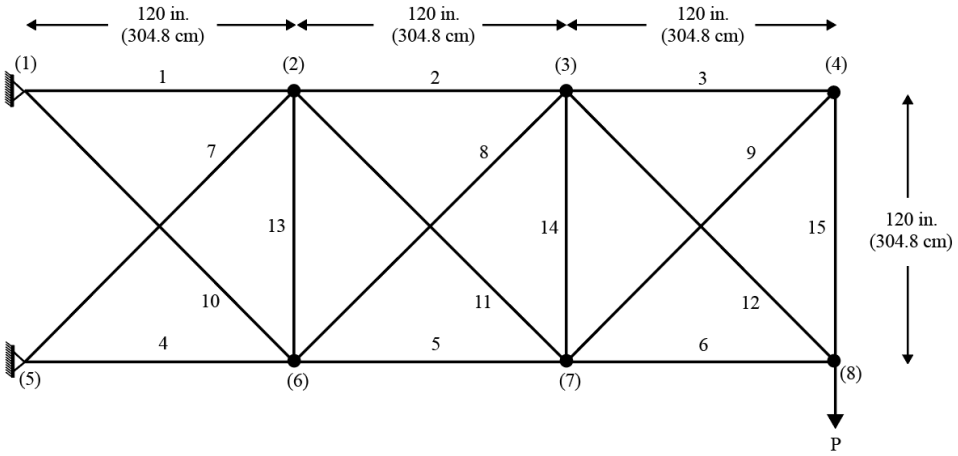


Figure 4.4 - The 15-bar plane truss

The 17-Bar Truss

The 17-bar problem consists of a cantilevered truss, subject to a downwards load of 100 kips (444.8 kN) at its furthest node. A displacement and stress constraint is placed on the truss structure, with a penalty function applied to all solutions which do not satisfy the constraints. The minimum cross-sectional area allowable for each bar element is 0.1 in^2 (64.5 mm^2) with a Young's modulus of 30,000 ksi (206.8 GPa) and a material density of 0.268 lb/in^3 (7414 kg/m^3). Each bay of the frame is spaced at intervals of 100 inches (2.54 m), with cross-bracing elements providing lateral rigidity to the structure.

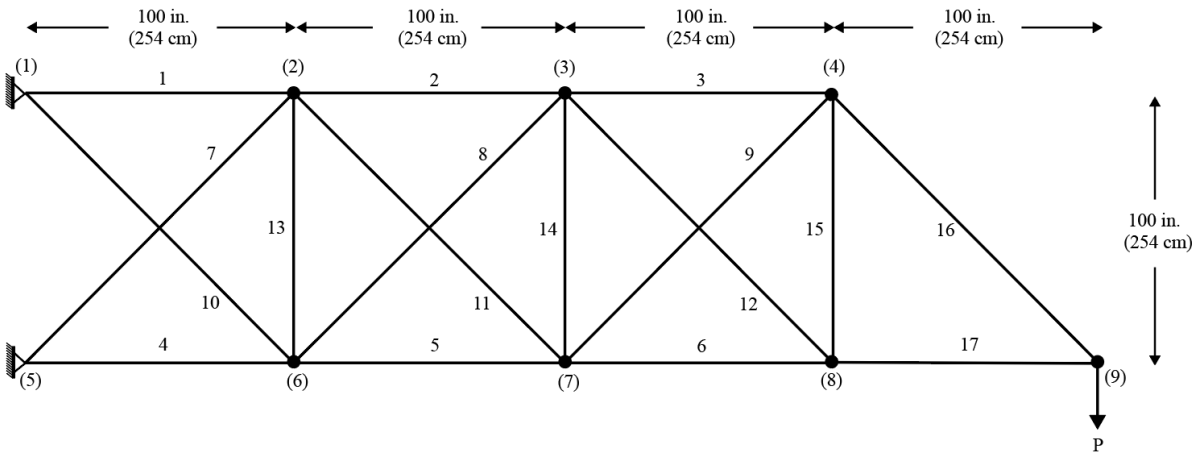


Figure 4.5 - The 17-bar truss

The 25-Bar Transmission Truss

The 25-Bar transmission truss is a three-dimensional structure consisting of 10 nodes supported by 4 pin supports. Due to the symmetry of the structure, it is possible to categorise these elements into eight distinct structural groupings, such that the benchmark test has eight variables which may be algorithmically adjusted. Four loads are applied at nodes 1,2,3 and 6, respectively. Designs generated for this structural problem are subject to displacement and stress constraints. As a discrete optimisation problem, the cross-section for each of the twenty-five members is selected from a list of 34 discrete areas, provided within the appendix of this thesis. The number of function evaluations executed for each of these benchmark tasks is outlined in Table 4.2

Group	Element(s)
1	A_1
2	A_2–A_5
3	A_6–A_9
4	A_{10}–A_{11}
5	A_{12}–A_{13}
6	A_{14}–A_{17}
7	A_{18}–A_{21}
8	A_{22}–A_{25}

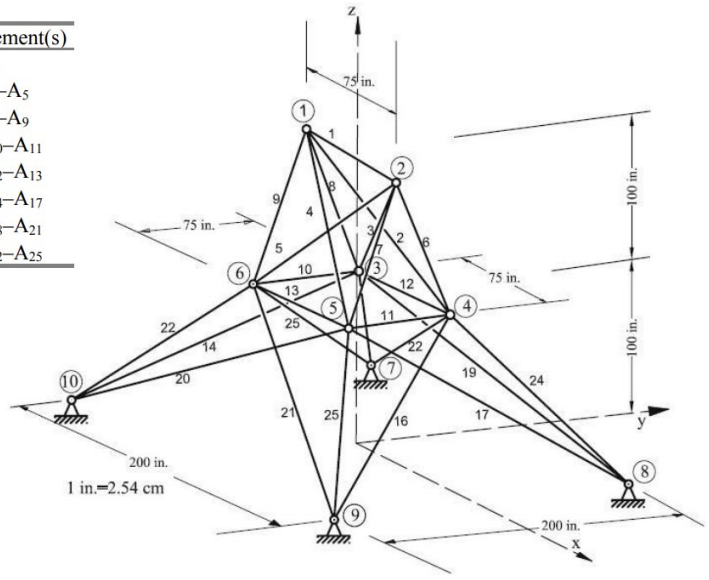


Figure 4.6 - The 25-bar transmission truss [61]

Table 4.1 - Parameter initialisation

Algorithm	Parameter
GA	$P = 30$, $Mr = 0.1$
PSO	$P = 30$, $C_1 = 2$, $C_2 = 2$
$DE_{Explore}$	$P = 30$, $F = 0.5$, $Cr = 0.7$ $v_i^t = x_{R_1}^t + F\left(x_{R_2}^t - x_{R_3}^t\right)$
$DE_{Exploit}$	$P = 30$, $F = 0.5$, $Cr = 0.7$ $v_i^t = x_{best}^t + F\left(x_{R_1}^t - x_{R_2}^t\right)$
DE_{PPO}	$P = 30$, $F = 0.5$, $Cr = 0.7$ $v_i^t \epsilon \left\{x_{R_1}^t + F\left(x_{R_2}^t - x_{R_3}^t\right),\ x_{best}^t\right.$

Table 4.2 - Allocated number of function evaluations

Benchmark Test	Function Evaluations
10 Bar (Constraint case 1)	3,000
10 Bar (Constraint case 2)	3,000
15 Bar	15,000
17 Bar	6,000
25 Bar	3,000

4.2.5. Benchmark results

There is a significant body of research within the field of metaheuristics which has been dedicated to the establishment of a comprehensive empirical framework to evaluate algorithm performance. This conventionally requires the collection of large quantities of trial data for each benchmark experiment, as metaheuristic algorithms are inherently stochastic in their sampling and exploration operators. The random variance induced between individual trials may be overcome by collecting generational recordings of convergence data across a large number of samples in a given test environment, allowing for a more holistic evaluation of an algorithm's performance. The comparative analysis that has been conducted through this research is guided by the precedence established within the field of metaheuristic design. As observed by Rardin and Uzsoy [166], there is a perpetual trade-off between computational efficiency and the quality of the derived solutions when attempting to quantify and compare the results generated by optimisation algorithms. Within this framework, all algorithms are provided the same number of function evaluations for a given benchmark test, which has been selected according to the complexity of the test itself. All optimisation algorithms were run for 100 trials within each benchmark environment, before the average and standard deviation of converged masses was taken according to the recommendations provided by Beiranvand et al. [167]. Distinctions are drawn between each

algorithm according to the average quality of the generated designs, offering a reliable methodology to compare the series of competing algorithms. The standard deviation combined with the averaged converged mass for elite solutions demonstrates the consistency of the algorithm to successfully navigate the search space and converge at a competitive design. The performance of the proposed solution relative to conventional metaheuristic methods is presented in Table 4.3 and Table 4.4. Notably, purely explorative and exploitative mutation schemes failed to consistently outperform each other, while the proposed method DE_{PPO} successfully converged at solutions across all benchmark environments, outperforming all other methods according to average elite mass. This is indicative that an adaptive mutation scheme is a competitive alternative to its static counterparts when prioritising convergence consistency.

Table 4.3 – Average elite mass across 100 trials

Benchmark Test	GA	PSO	$DE_{Explore}$	$DE_{Exploit}$	DE_{Rand}	DE_{PPO}
10 Bar Truss - 1	695.06	559.76	462.33	504.03	473.64	**460.50**
10 Bar Truss - 2	1087.36	855.74	703.46	750.09	725.49	**696.22**
15 Bar Truss	82.16	80.16	79.91	79.48	77.71	**77.62**
17 Bar Truss	2962.45	2711.64	2652.29	2637.73	2626.12	**2620.09**
25 Bar Truss	160.45	150.77	122.54	124.41	122.83	**121.06**

Table 4.4 - Percent improvement in average converged mass achieved by DE_{PPO}

Benchmark Test	GA	PSO	$DE_{Explore}$	$DE_{Exploit}$	DE_{Rand}
10 Bar Truss - 1	33.75%	17.73%	0.40%	8.64%	2.77%
10 Bar Truss - 2	35.97%	18.64%	1.03%	7.18%	4.03%
15 Bar Truss	5.53%	3.17%	2.87%	2.34%	0.12%
17 Bar Truss	11.56%	3.38%	1.21%	0.67%	0.23%
25 Bar Truss	24.55%	19.71%	1.21%	2.69%	1.44%

4.2.6. Discussion

The benchmark tests selected from the available literature provide a diverse collection of discrete and continuous design problems, subject to a variety of stress and deflection constraints. The self-adaptive differential evolution algorithm proposed in this research achieved the best averaged elite fitness across all benchmark tests, demonstrating the viability of the self-adapting mutation strategy when compared against analogous metaheuristic algorithms. Full documentation of the performance metrics recorded across each benchmark test are provided within Appendix A of this thesis. The standard deviation of the mass achieved by DE_{PPO} for each benchmark test are consistently lower or equivalent to the leading competitors, indicating that the spread of masses achieved by the proposed algorithm are tightly grouped around optimal solutions. This is contrasted by

the large standard deviation of masses that are demonstrated by exploitative algorithms such as PSO and $DE_{Exploit}$, due to their premature convergence towards local optima. By adapting the exploration-exploitation response, DE_{PPO} is capable of navigating the search space intelligently — maintaining genetic diversity to avoid premature convergence while conservatively exploiting the most competitive designs in each generation.

By comparing the average fitness of the best designs generated in each trial and the standard deviation of these results, this research posits that the proposed novel algorithm outperformed all other benchmarked algorithms. The self-adapting algorithm leveraged experimental data to alternate between explorative ($DE_{Explore}$) and exploitative ($DE_{Exploit}$) mutation operators, allowing DE_{PPO} to outperform each of these methods individually. On average across all benchmark tests, final designs generated by DE_{PPO} exhibited a consistent reduction in mass when compared to explorative and exploitative mutation operators, respectively. To demonstrate that the policy had developed an efficient exploration-exploitation strategy, a random operator selector (DE_{Rand}) was developed, which was outperformed in each test by the proposed algorithm. DE_{PPO} was also benchmarked against leading metaheuristic algorithms such as the Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO), which it was also capable of outperforming for all of the tested structural optimisation problems. The average mass of converged designs generated by DE_{PPO} was 12.53% less than the results recorded by Particle Swarm Optimisation and 22.27% less than those created using the Genetic Algorithm. Each of the algorithms tested within this research were provided the same experimental setup, with an equivalent number of function evaluations available for each benchmark test. Collectively, these results demonstrate the efficacy of reinforcement learning methods such as Proximal Policy Optimisation to adapt the hyperparameters of metaheuristic algorithms during the generational loop, as an improved approach to algorithmic structural optimisation.

The objective of this research was to train a network that was capable of consistently converging at high quality solutions within a set computational budget. The training environment was constructed such that agents that were capable of repeatedly generating designs that were proximate to the optimal fitness were rewarded according to the number of times this convergence was achieved. Training was terminated if the computational budget was exceeded for a given trial without achieving the required fitness threshold. As such, the designed reward signal was predisposed towards producing a network that emphasised consistency in convergence within a defined timeframe, which is highly aligned with the objectives of practicing designers and engineers. This reflects the reasoning behind measuring algorithmic performance on average, rather than simply referring to the algorithm that generated the best solution across 100 trials for each benchmark test. The stochastic nature of metaheuristic algorithms can obfuscate algorithmic performance if the analysis is limited to a small sample of optimisation outcomes. Notably, the average and standard deviation of purely exploitative algorithms across the 100 trials reveal the inconsistency with which these algorithms converge at high-performing results, demonstrating the exploration-exploitation trade-off for practicing designers and engineers.

It must be noted here that this research has prioritised convergence consistency over speed, achieved through the careful construction of the reward signal used during the training of the created reinforcement learning policy. As such, the exploration-exploitation response of DE_{PPO} attempts to strike a balance between mutation operators according to the stage in the optimisation timeline. Additionally, the selection of computationally efficient design problems for benchmarking reflects the prioritisation of design efficiency over convergence speed within this research. In this sense, the mediation between exploration and exploitation must also account for the complexity of the task in practice, which will need to be evaluated on a case-by-case basis. If a project requires a methodology that is more aggressively exploitative, the reward signal may simply be altered in the training phase of PPO to favour agents that achieve convergence speed over convergence consistency. For these applications, the policy will need to be retrained and the reward signal must be reconstructed. This limitation provides opportunities for future research to explore potential variations to the proposed algorithm, in which the action and observation spaces, reward signals, and training environment may be modified to account for separate research objectives.

4.2.7. Summary of key findings from Section 4.2

The comparative benchmark tests recorded within this research demonstrate that the dynamic exploration-exploitation response generated by DE_{PPO} consistently outperforms static Differential Evolution mutation operators. This research furthermore argues that reinforcement learning methods such as PPO provide a suitable framework to intelligently mediate the exploration-exploitation response for structural design tasks. In order to construct a testing environment that was conducive towards iterative experimentation, this research was primarily focused on leveraging the Direct Stiffness method for truss optimisation problems. These optimisation problems are well suited for the development of novel optimisation techniques due to the simplicity of the structural calculations that are required to determine the suitability of candidate solutions, significantly reducing the computational cost of the generational optimisation loop. The benchmark problems selected for this research were drawn from the extensive literature investigating metaheuristic algorithms for structural optimisation. For each benchmark problem, the converged mass of elite solutions averaged across 100 individual simulations was used as a primary indicator of algorithmic performance. Designs generated by the proposed method outperformed both static variants of the differential evolution algorithm for all benchmark problems. Furthermore, DE_{PPO} outperformed all alternative metaheuristic algorithms that were benchmarked, outperforming PSO and GA by an average of 12.53% and 22.27% respectively. This improved engagement with the exploration-exploitation trade-off within the generational loop extends beyond the novel algorithm proposed within this research, suggesting the potential application of leading reinforcement learning methods such as PPO to self-adapt the operators and hyperparameters for algorithms across the metaheuristic family.

4.3. OPTIMISING MEMORY-EFFICIENT MULTIMODAL NETWORKS FOR IMAGE CLASSIFICATION USING DIFFERENTIAL EVOLUTION

T. Hielscher[1*] , S.A. Hadigheh[1]

[1]School of Civil Engineering, Faculty of Engineering, The University of Sydney, Sydney, New South Wales 2006, Australia

*Corresponding Author: Tommy Hielscher

Applied Soft Computing, https://doi.org/10.1016/j.asoc.2025.112714, Published in March 2025

4.3.1. Abstract

This paper presents a novel approach to developing tuned model architectures and hyperparameter sets for multimodal classification, leveraging Differential Evolution to optimise for memory-efficient design (MEMODE). The evolved model combines image channel data with vectorised word matrices generated through trained Word2Vec embeddings, as a resourceful convolutional processing of text and image inputs. Additionally, to enrich the semantic representations drawn from text-based inputs, a pretrained ALBERT module is included to process a parallel input stream. The custom Differential Evolution algorithm employed utilises a penalised objective function that directly accounts for model size and Mean F1 performance. We focus on the critical task of hyperparameter optimisation and neural architecture search, while accounting for a set design limit of 100 MB. Significantly, the network evolved through the MEMODE algorithm achieved a validation mean F1 score of 0.8621, outperforming over 1,600 competing architectures, including manually constructed CNNs, AlexNet, ResNet variations, Inception modules, and models generated by alternative algorithms such as Bayesian Optimisation, Particle Swarm Optimisation and the Genetic Algorithm. These results emphasise the capacity of Differential Evolution for autonomous network design and hyperparameter tuning when training memory-efficient models.

4.3.2. Introduction

Efficiency in neural network design is a critical challenge within the landscape of modern machine learning, where resource constraints are often a significant limiting factor impacting model performance. Memory limitations within training environments and deployed devices during inference have demonstrated a growing need to design network architectures that are resourceful and efficient with respect to their computational requirements. Central to the performance of these models is a careful trade-off between model size and accuracy, conventionally achieved through iterative experimentation with network architectures and hyperparameter tuning. This challenge is compounded by the increasing complexity and size of deep learning systems in recent years, with modern multimodal models managing multiple streams of inputs that are propagated through architectures containing millions of parameters. The field of Neural Architecture Search (NAS) has emerged within this context to provide an autonomous method of network design, leveraging techniques such as

reinforcement learning [168], metaheuristic algorithms [169], and Bayesian optimisation [170] to supplement manual network design.

Within the field of deep learning, metaheuristic algorithms have a rich history of optimising network architectures and hyperparameters [171-173]. Population-based metaheuristic algorithms such as Differential Evolution (DE), the Genetic Algorithm (GA), and Particle Swarm Optimisation (PSO) have been used for neural architecture search and hyperparameter tuning with compelling results [174]. Wang et al. benchmarked the performance of the Differential Evolution algorithm to evolve the architecture of convolutional neural networks across six widely used deep learning datasets for image recognition tasks [175]. These included the MNIST Basic, MBI, MDRBI, MRB, MRD, and Convex datasets. Twelve state-of-the-art models were compared against the converged network across these benchmark datasets, allowing for a comparative analysis of the performance of the Differential Evolution algorithm to generate competitive hyperparameters. The networks evolved through the proposed method outperformed all other models across the MBI and MRB datasets, while achieving the second best One Sample T-Test in the MDRBI and MRD datasets. When evaluated against a competing Particle Swarm Optimisation algorithm (IPPSO), the Differential Evolution method outperformed it in five out of the six benchmark datasets. These results demonstrate the competitive capabilities of Differential Evolution to converge populations of networks towards high-performing architectures and hyperparameter sets relative to other competitive metaheuristic algorithms. Notably however, NAS research investigating these algorithms has predominantly focused on maximising performance metrics such as classification accuracy, ignoring design considerations such as model size and parameter count that have become increasingly critical to modern deep learning systems.

4.3.2.1 Research Gaps

The field of Neural Architecture Search has largely ignored the importance of memory-efficient network design, particularly within the context of complex multimodal models that require several converging architectural branches. These regions each have bespoke requirements to process their respective modalities, such as text, video, images, audio, and signal data. This in turn presents a unique series of fine-tuning challenges that are extremely difficult to solve *a priori*. The additional computational complexity of multimodal models relative to their unimodal counterparts furthermore affirms the importance of research developing methods for automating memory-efficient multimodal design. Given their broader application within fields of neural evolution, hyperparameter tuning and black-box optimisation, this research posits that Metaheuristic algorithms such as Differential Evolution offer a black-box approach to generating memory-efficient architectures across diverse design spaces. Within the context of increasing demand for computationally efficient network design, this research paper introduces Memory-Efficient Multimodal Optimisation using Differential Evolution (MEMODE), designed to balance the trade-off between network performance and memory efficiency in terms of model size. MEMODE has been applied to the complex task of organising and fusing multiple modalities

into a single classification model while satisfying a size limitation of 100 MB. This has been achieved through the application of a penalty factor within the single objective fitness function, with the intention of incentivising high performing, memory-efficient architectures.

Recent advances within the field of NAS are summarised in Table 4.5, outlining competing algorithms, the novelty of the proposed method, and the critical research gaps that MEMODE aims to address. Previous NAS approaches have primarily focused on exclusively maximising performance, fixing the fitness function to metrics such as classification accuracy and regression loss [176-178]. The MEMODE algorithm diverges from the established literature by introducing a penalty function that disincentivises populations of networks from generating models that exceed a predefined memory footprint. In this sense, the MEMODE algorithm strikes a balance between inference performance and memory efficiency—a critical challenge that has been previously ignored within the available literature and has compounding complexity when applied within the context of multimodal design. Due to the challenges associated with optimising branching streams of input, complex multimodal tasks have also been ignored within the available literature, with recent research prioritising simpler unimodal applications [175, 179-182]. The few related works that have aimed to investigate algorithmic multimodal design have themselves been limited by computationally inefficient fixed training schedules, optimising over a reduced generation count ($G \leq 10$), or limiting the total computational budget to only 100 trained models [183, 184]. These compromises in turn lead to a reduced optimisation duration, terminating the algorithm prior to fitness convergence and thereby significantly limiting the performance of the final model. The proposed MEMODE algorithm overcomes this limitation of previous NAS approaches through the implementation of penalised fitness tracking during optimisation, terminating a training instance if performance stagnation is observed over a defined number of epochs. This early stopping function effectively mitigates overfitting, discards invalid network architectures, and significantly reduces excessive training epochs during optimisation. A total of 1,600 multimodal networks were generated within this research project including 400 MEMODE models that were evolved over 20 generations. To ensure a fair comparison, non-parametric benchmarking was repeated across 10 subsets, using repeated holdout validation with shuffled splitting.

Table 4.5. Summary of related works and gaps addressed by the proposed method

Model	Search algorithm	Input Capacity	Fitness criterion	Novelty	Limitations
IPPSO [179]	PSO	Unimodal (image)	Mean accuracy	PSO for Architecture + Parameter search	Limited by binary parameter encoding.
PSO-CNN [176]	PSO	Unimodal (image)	Classification accuracy	DenseNet Architecture search	Focuses exclusively on convolutional blocks.
DECNN [175]	DE	Unimodal (image)	Mean accuracy	Differential Evolution based Architecture search	Optimises 6 architecture-specific parameters.
IDECNN [180]	DE	Unimodal (image)	Classification loss	Architecture + Parameter search	CNN layers are grouped in chromosomal encoding.
Genetic CNN [181]	GA	Unimodal (image)	Classification accuracy	Binary Architecture search	Binary representation is not suited for hyperparameter tuning.
CNN-GA [182]	GA	Unimodal (image)	Classification accuracy	Encoded Architecture search	FC layers are discarded in encoding strategy.
SLE-CNN [177]	GA	Unimodal (signal)	Classification accuracy	Architecture search for sleep state classification	Encoding strategy limited to pre-established CNN units.
ENAS-B [178]	BO	Unimodal (image)	Classification accuracy	Efficient Neural Architecture search	Performance is dependent on two-phase Bayesian Optimisation.
MOBON [185]	BO	Unimodal (image)	Classification accuracy	Bayesian Neural Architecture Search	Suffers from the curse of dimensionality.
GA based CNN [186]	GA	Unimodal (image)	Classification accuracy	Efficient Neural Architecture search	Single modality (images), no memory threshold/target value.
BO CNN [183]	BO	Multimodal (image + sensor data)	Mean Squared Error	Bayesian Neural Architecture Search	Only performed 100 function evaluations.
DC-NAS [184]	GA	Multimodal (image + text)	Classification accuracy	Differential Evolution based Architecture search	Computationally expensive, limited generation count (G = 10).
IPSO [187]	PSO	Multimodal	Classification accuracy, F1	PSO for Neural Architecture search	Bisection method stop condition (BsM).
MEMODE (proposed)	DE	Multimodal (image + text)	Mean F1 score + Memory penalty	Memory-efficient Multimodal Architecture + Parameter Search, Penalised Fitness Tracking, and Word2Vec convolution	Convolution fixes Word2Vec matrix to the same shape as the input image.

4.3.2.2 Research Significance

Within the field of deep learning, the task of selecting the suitable network size, layer counts, activation functions, regularisation methods, and learning rate has a significant impact on the resulting performance of the model. This task is compounded by the potential size of the design space for multimodal architectures and the stochastic nature of model performance over successive runs. Across deep learning tasks and dataset sizes, the suitable parameter set that maximises the performance of the network will also vary significantly. The traditional manual tuning of these hyperparameters will often require past experimental knowledge and a time-consuming iterative approach that is unlikely to result in a network that is proximate to global optimal performance. Additionally, as cutting-edge architectures have become larger and increasingly complex, memory efficiency has become critical to address concerns relating to hardware limitations, cloud services costs, training and inference speeds, latency, power consumption, and the processing of large datasets. Within this paper,

Differential Evolution has been introduced to generate a population of candidate networks that are optimised over successive generations for memory-efficient performance—an essential and largely overlooked requirement for modern deep learning models. The contributions of this paper are as follows:

1. We propose a novel MEMODE algorithm that optimises network architecture and hyperparameters, utilising a targeted penalty function to guide model evolution according to a predefined size limitation of 100 MB. This algorithm was designed to solve a significant gap within the available NAS literature, which has ignored memory-efficient model design, particularly within the context of complex multimodal tasks.
2. MEMODE is tasked with evolving complex multimodal architectures, including convolutional blocks, natural language branches, fully connected layers, and classification regions of the model, in addition to critical hyperparameters such as the dropout and learning rates.
3. We propose a memory-efficient method to convolutionally process image and text modalities through generated Word2Vec matrices. Early stopping has also been introduced to track historic fitness performance, prevent overfitting, and significantly reduce the computational cost of the MEMODE implementation.

4.3.3. Methods and Techniques

4.3.3.1 Word2Vec embeddings

Advances within the field of network embeddings have provided innovative approaches to process text inputs with traditional convolutional architectures. Word2Vec was created in 2013 as a natural language processing technique that captures the semantic similarity of words within a given vocabulary of text. The principal concept behind Word2Vec as proposed by Mikolov is the hypothesis that similar words within a given context may be expressed as proximate through a vectorised representation [188]. For CNN based implementations, these vectors are stacked to form a vectorised word matrix that can then be convolutionally processed [189-192]. This method was comprehensively evaluated by Jang et al., in which captions for news articles were used to train Word2Vec embeddings, designed to generate vectorised representations. The CNN model that was trained to evaluate these synthetic matrix images was used for a sentiment-analysis task, with performance measured according to the mean F1 score. According to the results of this research, models trained with stacked vector inputs generated through a pretrained Word2Vec embedding had a significantly improved performance, demonstrating that the learned semantic relationships captured by Word2Vec embeddings are viable inputs for convolutional neural networks when applied to conventional NLP tasks [193].

The capability of Word2Vec models combined with the image processing capabilities of convolutional neural networks has furthermore been examined extensively in related works. Kale et al. explored the application of the Word2Vec networks to generate vectorised word matrices, processed with traditional convolutional neural network architectures to categorise forms of Malware [194]. An initial shallow model was trained according to

the input vocabulary, generating a collection of Word2Vec embeddings. The index of each entry within this embedding object represents the vectorised map of each word within the training vocabulary. The vectorised outputs are then used as features for the downstream classification task, with the assumption that these vectorised word matrices provide additional semantic information that may be interpreted by traditional convolutional neural network architectures. According to the results from this paper, the Word2Vec-based model attained the best results across the benchmarked architectures, demonstrating the competitive capabilities of Word2Vec text transformations as processed by CNN models.

There are several key advantages of Word2Vec embeddings that are well-suited to the proposed multimodal architecture and optimisation framework when compared against competing methods such as GloVe and FastText. The proposed metaheuristic optimisation method requires a fast and efficient model to capture localised contextual relationships, providing meaningful embeddings for the convolutional branch of the multimodal model. Word2Vec models are capable of learning detailed semantic relationships between words and their context within a sentence, which can be trained and fine-tuned to large task-specific datasets. In contrast, GloVe prioritises global connections between words across the entire dataset vocabulary instead of detailed local syntactic relationships. GloVe also requires the creation of a memory-intensive co-occurrence matrix, limiting its application for training tasks that require fast, memory-efficient models. Similarly, FastText incorporates subword information to enrich the learnt embeddings, which is slower to train and is much more memory intensive. The speed of training Word2Vec models and their memory efficiency relative to competing embedding methods makes them a compelling candidate within the context of memory-efficient multimodal design and computationally expensive metaheuristic optimisation.

The Word2Vec model implemented within this paper consists of a shallow fully connected neural network with an input layer containing a neuron for each possible word within the training vocabulary. This model was initially trained to predict the context word for a given input word, such that the hidden layer captures the semantic relationships between words within the training vocabulary. The weights from this pretrained hidden layer then serve as the embedding dictionary, translating each word within an input caption into a vectorised form. The embeddings of this network are stored as a dictionary to map each word within the training vocab into a corresponding vector with a length of 224. In order to ensure continuity for each input channel to the convolutional region of the model, the output vector dimension was fixed to match the width of the input images after pre-processing. A total of 224 vectors were stacked together to form the vectorised word. An example diagram of vectorised word processing with a traditional convolutional neural network architecture is provided in Figure 4.7. This structure was selected to ensure that the Word2Vec matrix matched the shape of the resized input images. The new vectorised synthetic images are then added as a new channel to the RGB image data, allowing both input types to be consumed by the same network architecture, with the goal of providing a memory-efficient method to convolutionally process multiple input modalities. As model size is a limitation

that has been applied to the given design task, this proposed architecture has the significant advantage of leveraging text and image inputs within the convolutional region of the architecture, while still reserving an additional portion of the model size purely for conventional natural language processing.

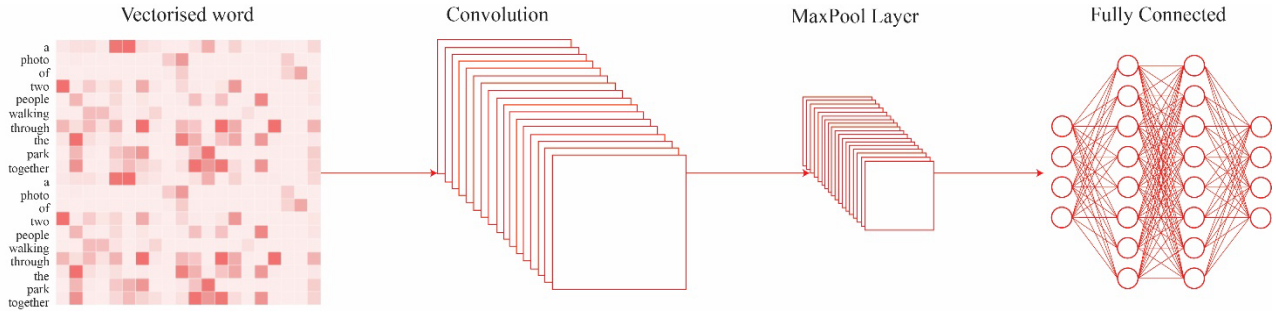


Figure 4.7 – Processing word vector matrix with traditional CNN architecture

4.3.3.2 Memory-Efficient Multimodal Optimisation using Differential Evolution

Within the proposed MEMODE algorithm, each gene of the initial population was randomly assigned within a unique predefined range, outlined in Table 4.6. For each model $x_i^{t=0}$ in an initial population of size N, the genetic vector of length D for each design variable is given by the following:

$$x_i^{t=0} = \{x_{i,1}^0, x_{i,2}^0 \dots, x_{i,D}^0\} \tag{4.4}$$

$$\text{with} \quad x_{i,j}^{t=0} = x_{min,j} + rand_{i,j}[0,1](x_{max,j} - x_{min,j}) \tag{4.5}$$

where $rand_{i,j}[0,1]$ is a random float chosen between 0 and 1. The maximum and minimum design bounds for each hyperparameter are given by $x_{max,j}$ and $x_{min,j}$ respectively. For this implementation, the tuned parameters include the learning rate, dropout probability, the size of fully connected layers throughout the multimodal model, and the number of layers for each convolutional grouping. A full summary of these parameters is provided in Table 4.6, which were selected based on their precedence within the field of Convolutional Neural Architecture Search [195] and their significant impact on the memory footprint of the multimodal model. The range of acceptable values defined for architectural parameters such as the number of Conv/FC layers and their respective sizes was determined through preliminary experimentation and prior knowledge of the general characteristics of designs that are appropriate for the defined memory requirements, excluding ranges that excessively exceeded the allowable memory footprint of 100 MB. This ensured that the initialised population contained networks that satisfied the necessary penalty threshold and that these parameters would be passed to successive generations. Within the MEMODE algorithm, these parameters are encoded as a unique genetic representation for each network, referred to as the 'chromosome'. Models within a given population were evaluated according to the mean F1 fitness calculated on the validation dataset, subject to a memory penalty function described in equation 4.8. The next generation of candidate solutions was then created through a

process of mutation and crossover, maximising the mean F1 fitness of generated candidate solutions while accounting for the imposed memory limitations for the given task.

Table 4.6. Hyperparameters tuned through MEMODE

Hyperparameter	Type	Bounds
Learning rate Lr	Continuous	0 – 2e-5
Dropout rate Dr_{CNN}	Continuous	0 – 0.9
Dropout rate Dr_{Final}	Continuous	0 – 0.9
Neurons per layer FC_{CNN}	Discrete	100 –1000
Neurons per layer FC_{Final}	Discrete	100 – 1000
No. F_{64} feature map blocks	Choice	[1,2,3]
No. F_{128} feature map blocks	Choice	[2,3,4]
No. F_{256} feature map blocks	Choice	[2,3,4]
No. F_{CNN} layers	Choice	[1,2]
No. F_{Final} layers	Choice	[1,2]

The mutation operator implemented within the MEMODE algorithm is outlined in equation 4.7, in which $u_{i,g}$ is the *ith* network, created within generation g out of G. In order to align the proposed MEMODE algorithm with previous implementations of metaheuristics for neural architecture search, the population size and generation count have been taken from the precedence set by Genetic CNN and CNN-GA ($N = G = 20$) [181, 182]. Each network within a given population evolved using the mutation function, generating an alternative mutated model $v_{i,g}$. The variable F serves as a scaling factor that controls the degree of genetic mutation, while the crossover of genetic code between candidate and mutated solutions for each parameter j was mediated by the crossover rate Cr. To exploit the genetic code of the best candidate observed so far (x_{elite}), the mutation operator retains the network chromosome set that achieved the greatest fitness after accounting for the memory penalty function. This in turn allows the collective genetic pool of each population to converge towards high-quality parameters over successive generations. Explorative behaviour was induced through the random selection of candidate solutions $x_{1,g}$ and $x_{2,g}$, which collectively contributed towards the genetic code of the next population of candidate networks according to the DE/best/1 mutation operation [157]. Crossover is induced at the parameter level, in which the crossover rate Cr provides the probability that a given hyperparameter j is replaced using the corresponding gene from the mutated chromosome, outlined in equation 4.7. For this task, the mutation scaling factor F and crossover rate Cr were both initialised to 0.5, based on the Convolutional Neural Architecture Search implementation provided by [196], which demonstrated competitive DE performance using static optimisation parameters and an initial population size of 20.

$$v_{i,g} = x_{elite} + F \times (x_{1,g} - x_{2,g}) \tag{4.6}$$

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & if\ rand(0,1) < Cr \\ x_{j,i,g} & if\ rand(0,1) \geq Cr \end{cases} \tag{4.7}$$

Algorithm 1: Population initialisation

Input size of initial population N
$|P_0| = 0$, $i = 0$
While $|P_0| < N$
    Randomly initialise learning rate within bounds, $0 \leq Lr \leq$ 2e-5
    Randomly initialise FC dropout rate within bounds, $0 \leq Dr_{Final} \leq 0.9$
    Randomly initialise CNN dropout rate within bounds, $0 \leq Dr_{CNN} \leq 0.9$
    Randomly select neurons in classifier layers within bounds, FC_{Final} [100 – 1000]
    Randomly select neurons after conv region within bounds, FC_{CNN} [100 – 1000]
    Randomly select no. F_{64} feature map blocks within bounds, F_{64} [1 – 3]
    Randomly select no. F_{128} feature map blocks within bounds, F_{128} [1 – 3]
    Randomly select no. F_{256} feature map blocks within bounds, F_{256} [1 – 3]
    Randomly select no. fully-connected layers in conv region, F_{CNN} [1 – 3]
    Randomly select no. fully-connected layers in classifier, F_{Final} [1 – 3]
    Create a network ($u_{i,0}$) with selected parameter set
    $P_0 \leftarrow P_0 \cup u_{i,0}$
    $i = i + 1$
Return P_0

Algorithm 2: Mutation + Crossover

Input population P_g of size N
Record elite candidate x_{elite}
For each network i in P_g
    Generate random integers $r_1, r_2 \in (1, N)$
    Select random candidate $x_{1,g}$, $x_{2,g}$ using random indexes r_1, r_2
    Mutate new network chromosome $v_{i,g} = x_{elite,g} + F \times (x_{1,g} - x_{2,g})$
    For each parameter j
        If $rand(0,1) < Cr$
            $u_{j,i,g} = v_{j,i,g}$
        Else if $rand(0,1) \geq Cr$
            $u_{j,i,g} = x_{j,i,g}$
    Record new network chromosome $u_{i,g}$ after mutation and crossover
    $P_{g+1} \leftarrow P_{g+1} \cup u_{i,g}$
Record new population of networks P_{g+1}
Return P_{g+1}

To achieve a balance between memory efficiency and model performance, the fitness criterion used within MEMODE is a single-objective function with a penalty factor. The goal of the implemented algorithm is to maximise the mean F_1 score of the multimodal model while disincentivising the genetic pool from evolving models that exceed the allowable memory footprint (D). This behaviour was induced within the objective function through the custom a penalty parameter (k), which reduces the fitness (f) of networks that exceed the allowable model size proportional to the network size (S). This allows the MEMODE algorithm to effectively regulate the trade-off between model performance and memory efficiency, which may be adapted to meet the specific requirements of other deep learning tasks. For this research task, the multimodal model was restricted to a size of $S \leq D$ = 100 MB. The penalisation coefficient k = 0.01 was selected to scale the impact of the model size (S) within the fitness function (f). By selecting a value of 0.01 for the penalty factor, the degree of penalisation due to the model size was effectively downscaled to align with the range of F_1 scores calculated

for each network, $F_1 = \{ F_1 \in \mathbb{R} \mid 0 \leq F_1 \leq 1 \}$. The mean F1 score and the expected allowable model sizes for each model were scaled within the same range [0,1] to prevent either factor from excessively dominating the fitness function.

$$f = \begin{cases} F_1 & \text{if } S \leq D \\ F_1 - k \cdot S & \text{if } S > D \end{cases} \tag{4.8}$$

Algorithm 3: Fitness evaluation function

```
Input population P_g
Initialise empty fitness array f_{g,all}
For each network i in P_g
    Initialise network u_{i,g}
    termination == False
    While termination == False
        For epoch in training epochs
            Train network u_{i,g}
            Calculate validation mean F_1 score for network u_{i,g}
            Update stagnation value if F_1 stagnation is observed
            If stagnation > patience
                termination = True
        termination = True
    Record size S of network u_{i,g}
    If S ≤ D
        f = F_1
    Else
        f = F_1 − k · S
    f_{g,all} ← f_{g,all} ∪ f
Return f_{g,all}
```

One significant limitation impacting previous implementations of metaheuristic algorithms for neural architecture search is the computational cost of training populations of networks for a sufficient number of generations [181, 182, 197]. To overcome this common challenge, MEMODE implements a stagnation metric that tracks the best mean F_1 score achieved on the validation dataset during the training process and terminates the training if performance does not improve over a chosen number of epochs. This patience parameter allows the algorithm to explore a larger number of potential model designs for the same computational budget by training until the maximum number of epochs is reached or the patience threshold is exceeded, and early stopping is triggered. The implementation of this early stopping termination condition is described in Algorithm 3, detailing how the validation mean F_1 score for the current network $u_{i,g}$ is tracked during training. At the end of each epoch, the validation mean F_1 score is compared against the best value recorded from previous epochs and the stagnation value is increased if no improvement is observed. When the stagnation value exceeds the patience threshold, training is terminated, the metrics of the training process are recorded, and the results are

passed to the optimisation function. Conversely, if the best recorded validation mean F_1 score is exceeded, the stagnation value is reset. For this research, training was executed for 50 epochs using a patience of 5, with the aim of achieving a desirable 'price-performance' ratio during the training process [198]. Stagnation tracking significantly improves the training efficiency and inference performance of MEMODE models relative to traditional approaches. It prevents overfitting by terminating training as soon as validation performance begins to degrade, enhances generalisation across unseen data, and reduces training time by eliminating unnecessary training epochs on low-performing models generated through the evolution process. A diagrammatic representation of the proposed MEMODE algorithm is outlined in Figure 4.8. The final model achieved through the proposed MEMODE algorithm is illustrated in Figure 4.11 and is detailed in Table 4.8. All 400 models evolved through the MEMODE algorithm were trained on an intel(R) Core(TM) i7-11700, with an NVIDIA GeForce RTX 4090 over five GPU days. Alternative algorithms developed within this research project for benchmarking purposes were also afforded the same computational budget.

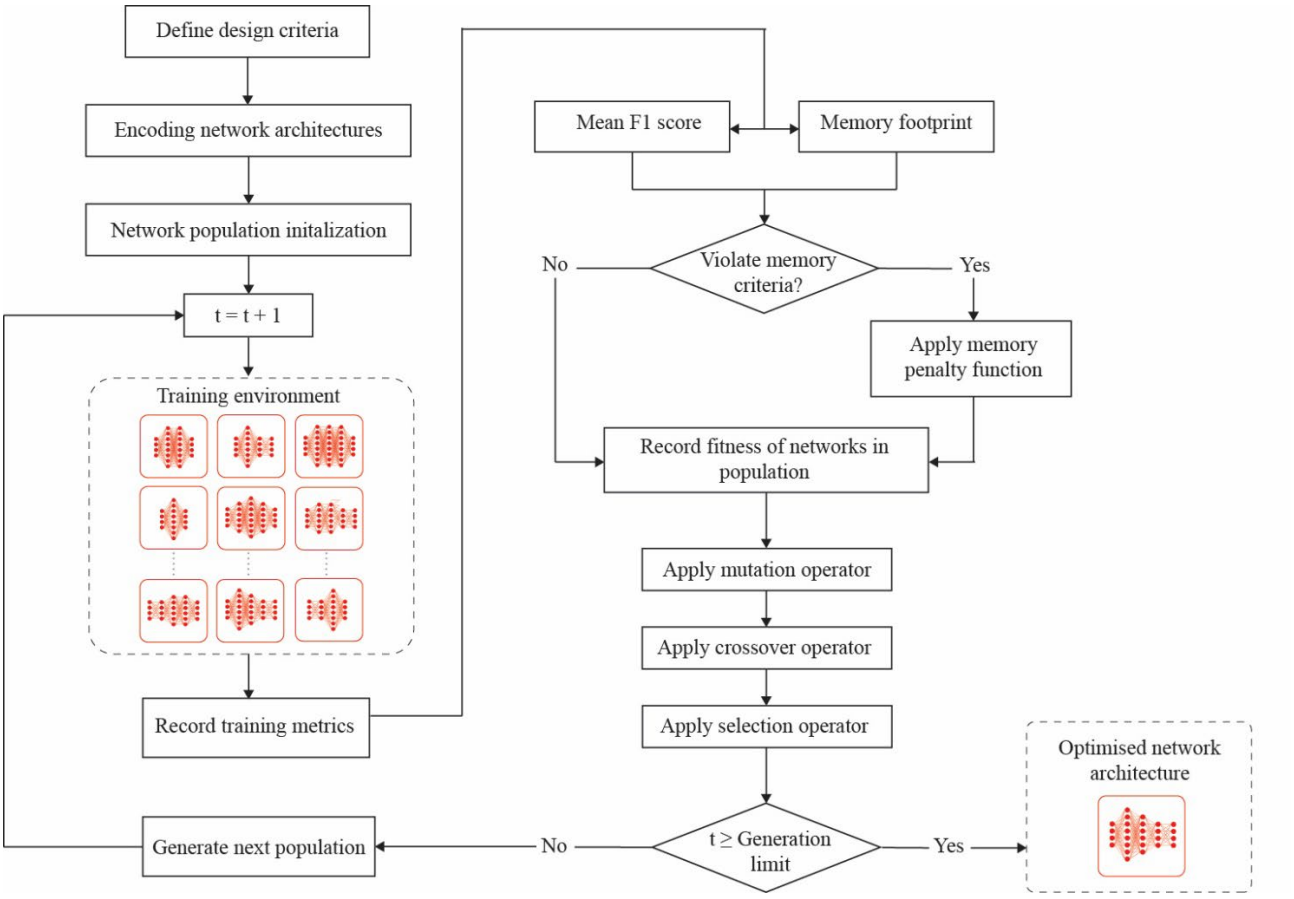


Figure 4.8 – Proposed MEMODE optimisation procedure

Algorithm 4: MEMODE

Generate preliminary population P_0
For each network $u_{i,0}$ in P_0
    Train and evaluate performance of given network
    Record mean F1 score and network size
    Calculate fitness and apply memory penalty
Record elite candidate x_{best}
For each generation g in G
    Mutate genetic code of all networks within P_g
    Crossover genetic code of all networks within P_g
    For each network in P_g
        Train and evaluate performance of given network
        Record mean F1 score and network size
        Calculate fitness and apply memory penalty
    Keep or reject new candidate solutions based on fitness comparison
    Check status of elite candidate x_{best} and update if performance has improved
End

4.3.3.3 Evolving Architectures

Traditional implementations of Differential Evolution for NAS will encode hyperparameters as discrete integer values, which are indexed within a vector for each individual solution. These typically include variables such as the number of layers for a given section of the model and the size of each layer. Similarly, DE implementations focusing entirely on hyperparameter optimisation will conventionally rely on continuous values to represent the entire chromosome, such as the dropout and learning rate. These fixed encoding strategies are particularly inefficient when considering the unique challenges of optimising memory-restricted multimodal models, comprised of various architectural branches that are effectively competing for parameter allocation. The proposed MEMODE algorithm allows for three distinct encoding techniques, incorporating discrete, continuous and mapped index values, summarised in Table 4.6. By incorporating a mixed encoding strategy at the parameter level, the proposed MEMODE implementation effectively reconciles discrete architectural search and continuous hyperparameter optimisation within a single algorithm. This is critical, as these parameters are often interdependent—the necessary degree of regularisation for a given region will naturally vary according to the size of the architecture, which is compounded when designing complex multimodal models with distinct, interdependent input branches.

The final multimodal model evolved within this research project uses two distinct branches to process the text and image inputs individually, before fusing the final signals from each stream into a fully connected classifier. The convolutional branch of the model consists of a CNN region that infers its architecture from the genetic chromosome evolved during optimisation. The first convolutional layer takes 4 channels of input, including the pre-processed RGB image and the synthetic Word2Vec matrix. These are applied to 64 filters with a kernel size of 3, followed by a ReLU activation function. This layer pattern is repeated according to the genetic parameter F_{64}, followed by a max pooling operation. This relationship is also applied for 128 and 256 convolutional blocks, defined by the F_{128} and F_{256} genes respectively. The final layer in the CNN region outputs 512 feature maps,

which are flattened before being passed to the fully connected layers. The number of neurons per layer and the number of layers within this dense region of the CNN branch is defined according to the FC_{CNN} and F_{CNN} genes. For all convolutional operations, a stride size and padding of 1 are used.

To process the caption data with a more conventional methodology, the NLP region of the model tokenises the caption data before passing it into a lite version of a bidirectional encoder representation from transformers model (ALBERT), which is a variant of the BERT series of models. The BERT transformer model was developed by Google in 2018 [199] and has demonstrated competitive performance across a generalised range of NLP tasks [200]. The advantage of using a light version of the BERT architecture is that the model size and memory requirements are significantly reduced [201]. Specifically, the ALBERT model discussed within this multimodal architecture is only approximately 45 MB in size, leaving roughly 55 MB of the model size budget for the convolutional neural network and classifier. ALBERT effectively decreases the size and memory requirements of the base BERT model by grouping the layers of the neural network and sharing network parameters across its layers, reducing the number of trainable parameters required. This parameter sharing contributes significantly towards the efficiency of the ALBERT model, making it an excellent memory-efficient candidate for the given design task. Within this implementation, the NLP region outputs 19 values that are fused with the output from the traditional Convolutional-FC architecture (FC_{CNN}). A final fully connected region (FC_{final}) is then used as a classifier to categorise the fused signals, predicting 19 final outputs which directly map to each class index.

4.3.4. Experiment

4.3.4.1 Dataset and Algorithms

The dataset utilised for this research is a subset of the COCO Caption dataset [202], designed for multilabel classification tasks. This dataset was selected due to its multimodal requirements, providing both image and text inputs for each sample. The dataset itself contains a total of 40,000 images and text captions, with each sample containing one or more subjects belonging to 19 possible classes. The training dataset for this task contains 24,000 image-caption pairs, constituting 60% of the data used in this research project. To evaluate the hyperparameter tuning process used within the MEMODE algorithm, 10,000 (25%) samples were reserved for testing, while 6,000 (15%) samples were used for validation to track the training process. For all testing, training and validation datasets, input data was divided into separate modalities that were processed by two distinct branches of the model architecture. The NLP region of the model processed the caption text, while the convolutional module received four channels of data as input, including the three RGB tensors and a vectorised word matrix generated through trained embeddings. To further benchmark the performance of Differential Evolution for memory efficient hyperparameter tuning and architecture search, additional variations on the proposed implementation were also developed using the Genetic Algorithm (MEGA), Particle Swarm

Optimisation (MEPSO), and Bayesian Optimisation (MEBO). Details regarding these individual implementations are available in Appendix B of this thesis, including the selected control parameters for each algorithm.

4.3.4.2 Pre-processing

The training dataset was subject to a range of transformations that were selected to improve model performance for the given task. Min-max normalisation and zero mean unit variance normalisation (MVN) was applied to the input data [203], effectively transforming the range of features for the input dataset and rescaling the features such that the mean (μ) was zero and the standard deviation (σ) was one across all three channels of the input image. The synthetic image created by the Word2Vec embeddings was furthermore standardised with the same base formula.

$$x_{stand} = \frac{x-\mu}{\sigma} \tag{4.9}$$

To improve the generalisation capabilities of the model and prevent overfitting, data augmentation was also used to pre-process the input images prior to being passed to the convolutional neural network. First, the input image was resized to a square image of size 300×300 pixels. A random crop was then taken of the input image, transforming the image to a size of 244×244 pixels. This effectively allowed for a randomised capture of the input image that varied across successive epochs. Finally, a horizontal flip was taken of the input image with a probability of 0.5, augmenting the input signal to prevent overfitting [204]. The captions were converted into vectors of length 244 using the trained Word2Vec network, before being stacked to form a 224×224 synthetic image as an additional channel to the image data tensor. This representation was then processed by the convolutional portion of the network, an example of which is provided in Figure 4.9. An embedding length of 224 was selected to provide a uniform tensor size across each of the four input channels processed by the convolutional branch of the multimodal model. Notably, Convolutional Neural Networks employ parameter sharing, such that the same convolutional kernel is applied across each of the input channels. In order for this sharing to meaningful and effective, the input image and the word vector matrix generated from the caption were fixed to the same shape.

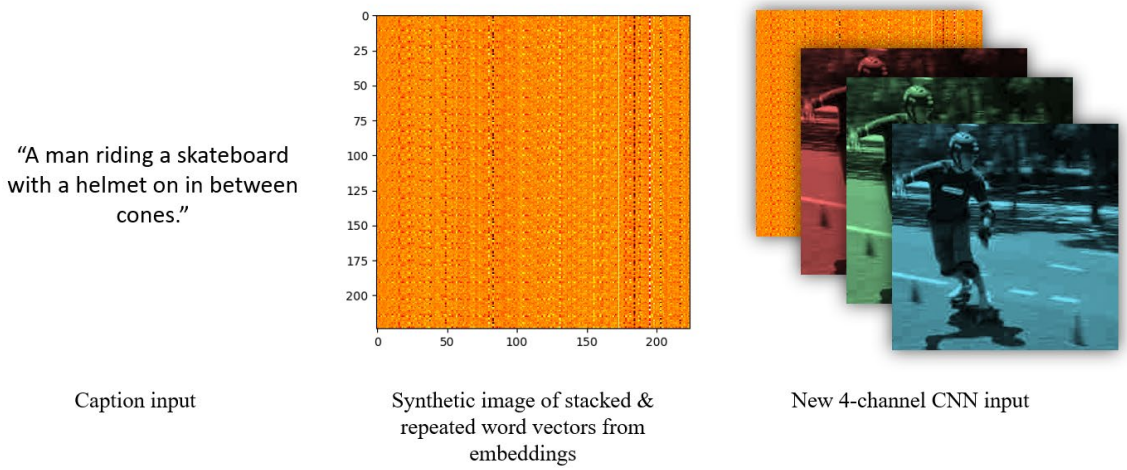


Figure 4.9 – Using trained embeddings to generate vectorised word matrices for CNN processing.

4.3.4.3 Loss Function

For the given multilabel classification task, cross entropy with logits loss has been used to quantify the error of the multimodal model. Within the training dataset, each input sample belongs to one or more classes, with the ground-truth vector represented in binary format. Cross entropy with logits loss combines the functionality of the sigmoid activation function with conventional cross entropy, providing a suitable framework to normalise the network output and compare the predictions against the ground truth for any number of classes. The advantages of using cross entropy with logits loss for multi-label classification is that it has the capability to compare two probability distributions while allowing for multiple true labels for a given sample [205]. The loss function used within this implementation follows the following equation:

$$L = -[\, t \cdot \log \sigma(x) + (1 - t) \cdot \log\big(1 - \sigma(x)\big)] \tag{4.10}$$

Within this context, t represents the ground truth values in binary format, while x is taken as the predicted value output by the network. The sigmoid function which is used to normalise the predictions from the network is represented by σ, scaling predicted values within the desired range of the binary ground truth.

4.3.4.4 Mean F1 Score

Within the context of multi-label classification, the F1 score is a metric that measures a model's precision and recall capabilities [206]. Precision (p) specifically measures the fraction of true positives (tp) to total positive predictions (tp + fp), while recall (r) measures the fraction of true positives to total relevant elements. A relevant element itself refers to the sum of true positives and false negatives. Collectively, the mean F1 score is interpreted as a measurement of model performance across classes, with values ranging from zero to one [207]. When applied to the given multi-label classification task, the mean F1 score is first calculated for each label

before the average is taken across the possible categories. When evaluating the performance of the elite candidates generated by each algorithm, the fitness is taken as the validation mean F1 score if the model size does not exceed the predefined design limitation of 100 MB. For each of the 19 classes investigated within this research paper, the F1 score for a given class is given by:

$$F_1 = 2\frac{p \cdot r}{p+r} \text{ , where } p = \frac{tp}{tp+fp} \text{ , } r = \frac{tp}{tp+fn} \tag{4.11}$$

4.3.5. Results

For each algorithm developed within this research, 400 neural networks were evolved, trained, and evaluated according to their fitness. A total of 1,600 models were trained for benchmarking, using Differential Evolution (MEMODE), Particle Swarm Optimisation (MEPSO), the Genetic Algorithm (MEGA), and Bayesian Optimisation (MEBO). The population metrics for each generation of the optimisation process are provided in Table 4.7, highlighting the evolution of networks over successive generations (G), the number of invalid networks recorded in each generation (In) the genetic diversity within each population (D_{norm}), the elite fitness achieved by each algorithm ($F1_{val}$), and the elite networks model size (S_{elite}). The final architecture, hyperparameter set, and fitness of each network generated by these competing algorithms is outlined in Table 4.8.

Table 4.7. Performance of elite network for each algorithm during optimisation

	MEPSO				**MEGA**				**MEMODE**				**MEBO**			
G	In	D_{norm}	$F1_{val}$	S_{elite}	In	D_{norm}	$F1_{val}$	S_{elite}	In	D_{norm}	$F1_{val}$	S_{elite}	In	D_{norm}	$F1_{val}$	S_{elite}
1	12	1.6187	0.8590	66.44	10	1.6230	0.8595	75.56	11	1.4000	0.8577	84.36	1	1.8124	0.8577	65.36
2	5	0.9136	0.8590	66.44	1	1.5413	0.8595	80.70	10	1.2652	0.8577	84.36	7	1.8056	0.8577	65.36
3	6	0.8767	0.8590	66.44	0	1.4498	0.8604	75.78	7	1.2947	0.8584	83.88	3	1.7214	0.8599	94.30
4	5	1.5049	0.8590	66.44	0	1.2870	0.8606	72.38	0	1.1561	0.8599	81.71	4	1.8986	0.8599	94.30
5	8	1.4530	0.8590	66.44	0	1.2539	0.8606	72.38	0	1.0747	0.8599	81.71	6	1.8038	0.8599	94.30
6	6	1.3219	0.8610	89.95	0	1.0387	0.8606	72.38	0	1.0813	0.8599	81.71	5	1.7320	0.8599	94.30
7	11	1.5926	0.8610	89.95	0	1.1570	0.8606	72.38	0	1.0084	0.8605	77.31	5	1.7495	0.8599	94.30
8	7	1.2276	0.8610	89.95	0	1.1591	0.8607	74.47	0	0.8688	0.8605	77.31	4	1.6739	0.8599	94.30
9	8	1.2894	0.8610	89.95	0	1.2257	0.8607	74.47	0	0.9795	0.8611	81.38	4	1.7555	0.8599	94.30
10	8	1.2332	0.8610	89.95	0	1.2809	0.8607	74.47	0	1.0005	0.8611	81.38	8	1.5876	0.8599	94.30
11	4	1.1750	0.8610	89.95	0	1.2831	0.8607	74.47	0	1.0047	0.8611	81.38	4	1.5865	0.8599	94.30
12	7	1.1909	0.8610	89.95	0	1.2770	0.8607	74.47	0	1.1191	0.8611	81.38	5	1.6496	0.8599	94.30
13	9	1.1528	0.8610	89.95	0	1.3029	0.8607	74.47	0	0.8692	0.8621	87.77	1	1.6871	0.8599	94.30
14	5	1.2502	0.8610	89.95	0	1.1571	0.8607	74.47	0	0.9057	0.8621	87.77	6	1.6451	0.8599	94.30
15	5	1.2666	0.8610	89.95	0	0.7937	0.8607	74.47	0	0.9444	0.8621	87.77	2	1.6277	0.8599	94.30
16	8	1.3500	0.8610	89.95	0	1.1833	0.8607	74.47	0	0.9479	0.8621	87.77	3	1.6873	0.8617	71.26
17	9	1.2262	0.8610	89.95	0	1.0774	0.8607	74.47	0	1.0056	0.8621	87.77	7	1.6557	0.8617	71.26
18	8	1.1746	0.8610	89.95	0	0.9155	0.8607	74.47	0	0.8641	0.8621	87.77	5	1.638	0.8617	71.26
19	10	1.4137	0.8610	89.95	0	0.8729	0.8607	74.47	0	0.9022	0.8621	87.77	6	1.6217	0.8617	71.26
20	11	1.5502	0.8610	89.95	0	0.7472	0.8607	74.47	0	0.9377	0.8621	87.77	3	1.6651	0.8617	71.26

To evaluate the diversity of populations generated by each algorithm, the Euclidean distance between each network of a given population has been calculated and summed. Genes have been normalised within this analysis to provide equal weight to each parameter. Populations evolved through the MEMODE algorithm experienced a 33.02% reduction in their genetic diversity over 20 generations when measured by the normalised

Euclidian distance (D_{norm}) summed for each gene. This illustrates the sequential convergence of the MEMODE algorithm within the provided computational budget. A similar result is demonstrated in the number of invalid networks present within each MEMODE population, with invalid networks being completely removed from the genetic pool by the fourth generation. This is also reflected by the results generated by the Genetic Algorithm, MEGA, which reduced the genetic diversity of the initial population by 53.96% and eliminated invalid networks from the population by the third generation. In contrast, populations generated by the MEPSO algorithm exhibited a larger genetic diversity at the cost of an increased proportion of invalid models. This indicates that the control parameters used for the Differential Evolution and Genetic Algorithm implementations have resulted in exploitative navigation of the search space, while the Particle Swarm Optimisation algorithm has prioritised explorative behaviour. Conversely, while the MEBO algorithm retained its genetic diversity over the 20 generations, the black-box solver was still capable of discovering high performing architectures and parameter sets, achieving the second-best validation fitness observed during optimisation. Notably, this architecture was also the smallest of the algorithmically generated final elite models, containing 18,237,498 parameters within an overall model size of 71.26 MB.

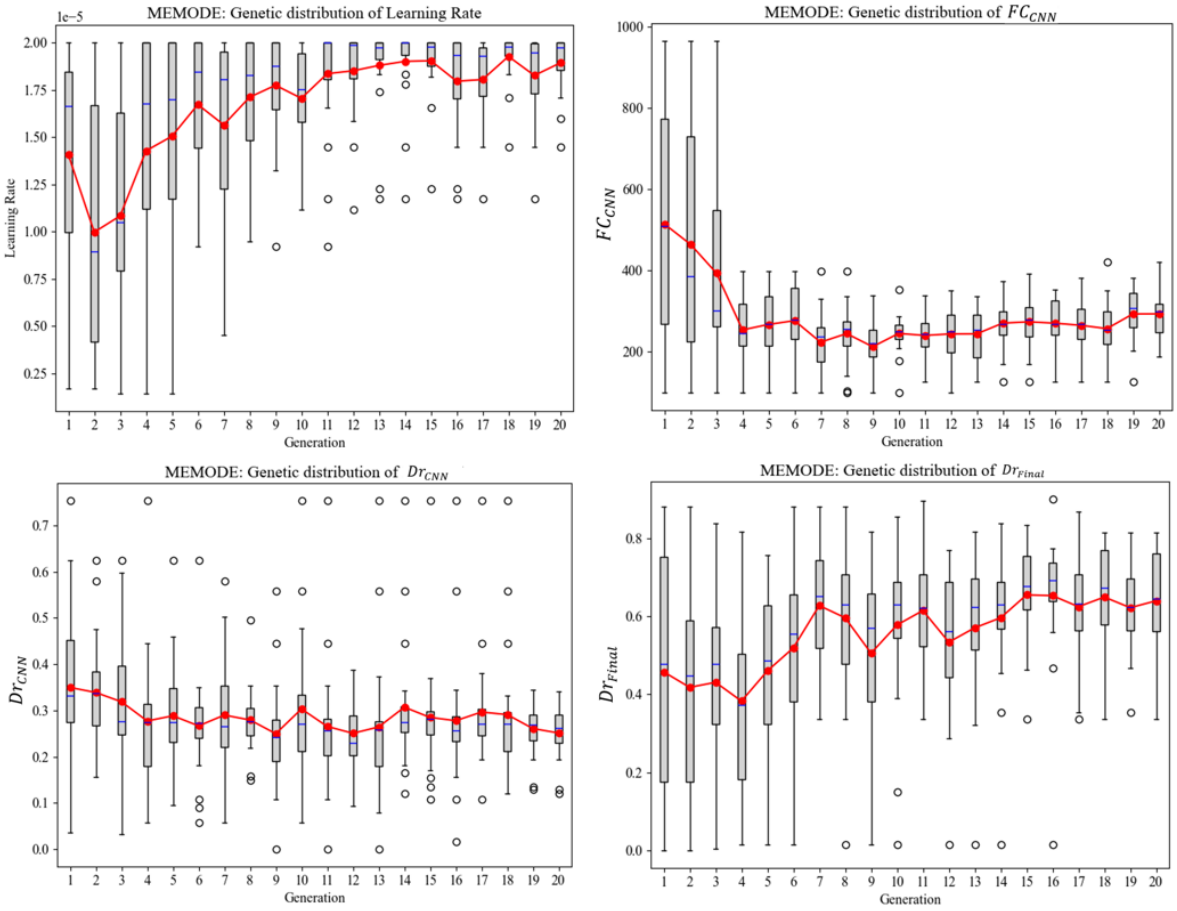


Figure 4.10 – Genetic distribution over successive generations of MEMODE

The genetic diversity of parameters optimised using the MEMODE algorithm is explored further in Figure 4.10, where the distribution of genes and their average value within each generation is graphed as overlayed boxplots and line charts, respectively. These diagrams reveal the progression of the optimisation process and the convergence of each population's genes, with the final evolved architecture utilising the maximum learning rate of 2×10^{-5}. The tightening quartiles of these boxplots over sequential generations furthermore illustrate how the MEMODE algorithm eliminated invalid networks from the genetic pool and optimised distinct regions of the multimodal model. By the end of the fourth generation, the fully connected region of the image processing branch FC_{CNN} was significantly reduced to less than 400 neurons for all individuals within the population, effectively controlling the size of image-processing branch without compromising the performance of the entire model. The regularisation patterns in the convolutional and classification FC layers were also mediated to account for this architectural shift. Through the MEMODE implementation, the dropout probability within the CNN region (Dr_{CNN}) was reduced to a value of 0.2626, while the dropout within the final classification layers (Dr_{Final}) was increased to 0.6939. This is in part due to the varying signal-processing tasks that each of these regions is responsible for. The fully connected region of the CNN region captures the convolutional feature maps, while the final FC layers are tasked with processing a fused signal from both CNN and NLP regions before outputting a consolidated classification signal. Notably, the MEMODE algorithm favoured smaller final classification regions, offering more of the budgeted parameter count to upstream architectures.

Due to the number of generations available to the optimisation algorithm and the time required to train each network within a given population, the exploration-exploitation behaviour exhibited by the MEMODE algorithm is highly desirable. If a more explorative navigation of the search space is required, the scaling penalty factor k may be reduced, encouraging the MEMODE algorithm to retain models that exceed the desired threshold and increasing the genetic diversity during optimisation. The final elite candidate evolved by the MEMODE algorithm is illustrated in Figure 4.11, including the CNN, NLP and classification regions of the evolved architecture. The image processing branch of the model takes four channels of input, processed by eight convolutional blocks and two fully connected layers. A max pooling operation is introduced after the second and sixth convolutional block, reducing the spatial dimensions of the processed feature maps and decreasing the computational complexity of downstream operations, thereby improving the model's memory footprint. The fully connected region of the CNN architecture consists of two layers, each containing 308 neurons, while the final classification layer consists of 562 neurons. The combined size of the proposed architecture after 20 generations of optimisation is 87.77 MB, satisfying the defined memory limitations imposed by the penalty coefficient. A detailed breakdown of the model's architecture and hyperparameters is provided in Table 4.8.

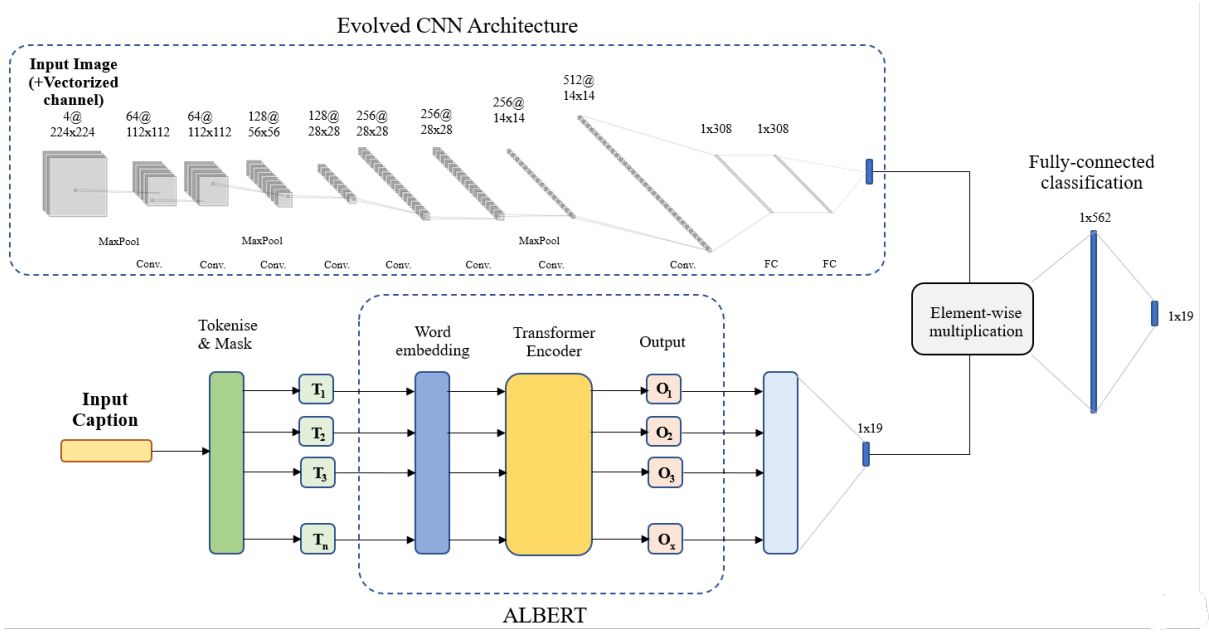


Figure 4.11 – Evolved multimodal architecture converged through MEMODE

Table 4.8. Evolved parameters of final models for each implemented algorithm

Algorithm	Learning rate	No. 64 conv	No. 128 conv	No. 256 conv	F_{CNN} layers	Neurons/ FC_{CNN}	Dropout Dr_{CNN}	F_{Final} layers	Neurons/ FC_{Final}	Dropout Dr_{Final}	Model Size (MB)	Fitness $F1_{val}$
MEMODE	1.98×10^{-5}	2	2	4	2	308	0.2626	1	562	0.6939	87.77	**0.8621**
MEGA	1.68×10^{-5}	3	2	3	1	198	0.5610	1	685	0.3483	74.47	0.8607
MEPSO	2.00×10^{-5}	2	2	3	2	353	0.6707	1	243	0.5354	89.95	0.8610
MEBO	1.19×10^{-5}	2	3	3	1	136	0.1259	2	787	0.3302	71.26	0.8617

To further benchmark the performance of the MEMODE algorithm against leading architectures within the field of multimodal deep learning, a series of competing models were trained and evaluated under the same framework. Competing baseline models were selected according to their relevance and demonstrated performance within the broader field of deep learning [128]. The results of this comparative analysis are provided in Table 4.9 and the performance of elite competing architecture is plotted in Figure 4.12 (a), including models generated by MEMODE, MEGA, and MEPSO algorithms. The final architecture evolved through the proposed MEMODE method outperformed all models that were tested against it with respect to model fitness, achieving a mean validation F1 score of 0.8621. In comparison, the highest performing ResNet and Inception modules achieved a mean validation F1 score of 0.8574 and 0.8544 respectively. The performance distribution of all models generated by the benchmarked algorithms is furthermore outlined in Figure 4.12 (b). The average validation mean F1 score of models generated by the MEMODE, MEGA, MEPSO, and MEBO algorithms was 0.8457, 0.8313, 0.8341, and 0.7497 respectively. The validation mean F1 scores recorded during training for all 1,600 algorithmically generated model architectures are plotted in Figure 4.13, with the final elite candidate for each algorithm marked in bold. The terminating mean F1 scores in each run are marked with an "x", indicating the starting trigger of the early stopping condition. Notably, the MEMODE algorithm outperformed each of the alternative algorithms in terms of elite and average model performance across the validation dataset, while the

Genetic Algorithm marginally outperformed Differential Evolution on the testing dataset. These results accumulatively highlight the capacity of the implemented MEMODE algorithm to navigate the design space and converge at high-performing architectures and hyperparameter sets.

Table 4.9. Comparative final performance of all multimodal models

Model	Tuning	Training Loss	Validation Loss	Fitness $F1_{val}$	Testing $F1_{test}$	Model Size (MB)	No. Parameters
AlexNet	Fixed	0.0679	0.0877	0.8513	0.8022	269.54	68,997,922
ResNet_{small}	Fixed	0.0488	0.0983	0.8567	0.8073	65.74	16,816,930
ResNet_{large}	Fixed	0.0656	0.0868	0.8574	0.8088	95.55	24,439,842
Inception	Fixed	0.0425	0.1058	0.8544	0.8046	69.98	17,904,658
ManualCNN	Manual	0.0524	0.0995	0.8533	0.8025	83.17	21,288,502
MEBO	Automatic	0.0822	0.0856	0.8617	0.8125	71.26	18,237,498
MEGA	Automatic	0.0609	0.0922	0.8607	**0.8137**	74.47	19,059,404
MEPSO	Automatic	0.0644	0.0885	0.8610	0.8101	89.95	23,021,940
MEMODE	Automatic	0.0560	0.0931	**0.8621**	0.8112	87.77	22,464,811

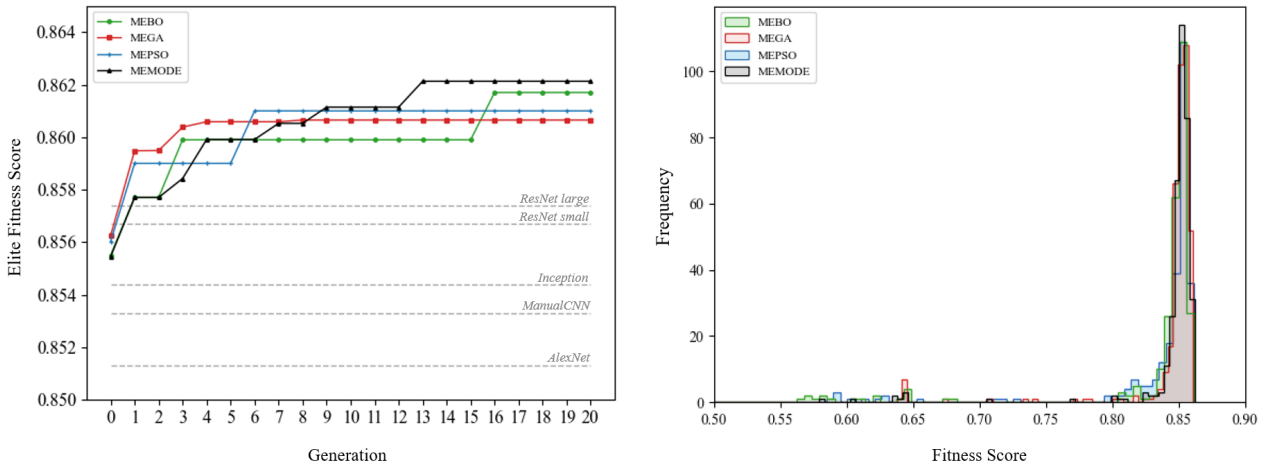


Figure 4.12 (a) – Tracking elite fitness score achieved after each generation of optimisation, 4.12 (b) – Performance distribution of generated valid models

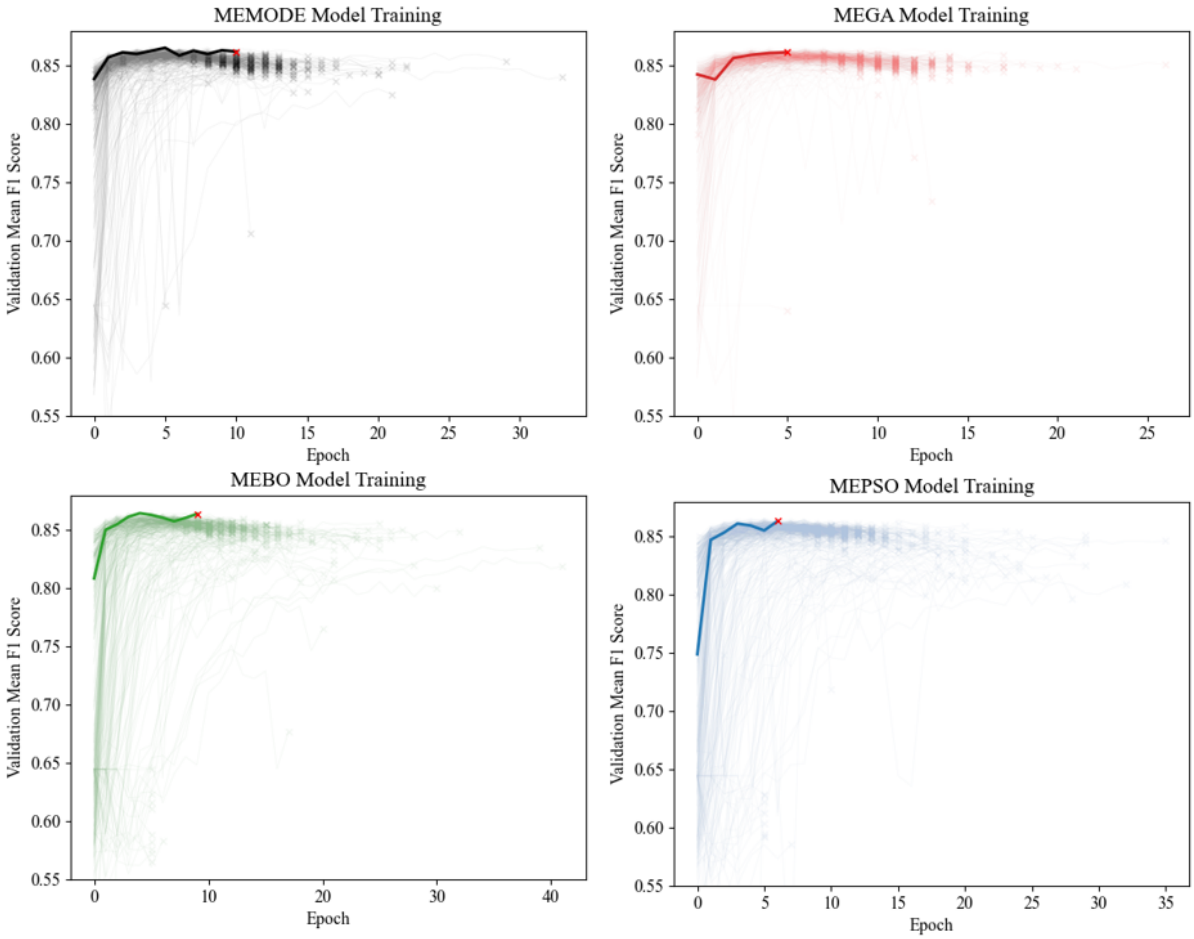


Figure 4.13 – Validation performance of final models compared against historic optimisation results

In evaluating the performance of each developed algorithm, a Friedman test was utilised to identify which method had the greatest statistical difference relative to the manually tuned and fixed models, outlined in Table 4.10. This non-parametric test is particularly suitable for evaluating the competing algorithms, as it does not assume that the recorded mean F1 scores follow a normal distribution. To conduct this statistical analysis, repeated holdout validation with shuffled splitting was used to divide the underlying validation dataset into 10 equally sized datasets. Competing architectures were then tasked with making predictions across these datasets, and the mean F1 scores was aggregated for each model. The statistical significance of each algorithm was then determined by comparing the corresponding F1 results recorded for the fixed and manually tuned models, included within the calculation of the Friedman Test statistic.

$$F_r = \left[\frac{12}{nk(k+1)}\sum_{i=1}^{k} R_j^2\right] - 3n(k+1) \tag{4.12}$$

In this context, n is taken as the number of datasets (10), k is the number of models being compared (6), and R_j is the sum of ranks across the j-th model. The Differential Evolution variant recorded a Friedman Test statistic of 18.74, the largest value observed within this analysis. This indicates that the MEMODE algorithm achieved a more statistically significant performance improvement when compared against the non-parametric results calculated for the Particle Swarm Optimisation, Bayesian Optimisation, and the Genetic Algorithm

implementations. Furthermore, the recorded p-value of the MEMODE algorithm was 0.0019, the lowest of the compared black-box optimisers, reaffirming the competitive performance of the MEMODE algorithm. Notably, all the algorithms developed within this research project achieved fitness results that were statistically significant relative to non-algorithmic competitors ($p < 0.05$), despite not benefitting from powerful residual connections or computationally efficient inception modules.

The mean F1 scores of each model across these 10 datasets were then evaluated through a structured pairwise win-loss comparison, outlined in Table 4.11 and summarised in Table 4.12. This pairwise test involved the construction of a performance matrix, in which each cell (i, j) records the number of times model i outperformed model j, providing a quantitative representation of relative model performance. This analysis reveals that the MEMODE algorithm achieved the highest average ranking of 2.8 when benchmarked against the competing models. Additionally, the MEMODE algorithm recorded the most pairwise wins (62) and the fewest pairwise losses (18), resulting in the strongest pairwise win-loss differential observed within this research paper (+44). The results of this non-parametric test and pairwise win-loss comparison indicate that the MEMODE algorithm achieved the most statistically significant fitness results when benchmarked against all competing fixed, manually tuned, and algorithmically evolved models.

Table 4.10. Comparison of algorithmic performance relative to manually tuned and fixed architectures

Model	Friedman Test Statistic	p-value
MEBO	18.0000	0.0029
MEGA	17.9429	0.0030
MEPSO	17.0286	0.0044
MEMODE	**18.9714**	**0.0019**

Table 4.11. Pairwise Win-Loss comparison

Model	*AlexNet*	$ResNet_{small}$	$ResNet_{large}$	*Inception*	*ManualCNN*	*MEBO*	*MEGA*	*MEPSO*	*MEMODE*
AlexNet	-	2	1	2	5	0	0	0	0
$ResNet_{small}$	8	-	3	6	6	2	2	1	1
$ResNet_{large}$	9	7	-	6	7	5	3	4	3
Inception	8	4	4	-	6	1	2	2	2
ManualCNN	5	4	3	4	-	1	1	2	1
MEBO	10	8	5	9	9	-	5	6	5
MEGA	10	8	7	8	9	5	-	5	3
MEPSO	10	9	6	8	8	4	5	-	3
MEMODE	10	9	7	8	9	5	7	7	-

Table 4.12. Summary of Pairwise Win-Loss Model Benchmarking

Model	Average Ranking	Total wins	Total Losses	Win-Loss Differential
AlexNet	8.0	10	70	-60
ResNet_{small}	6.1	29	51	-22
ResNet_{large}	4.6	44	36	+8
Inception	6.1	29	51	-22
ManualCNN	6.9	21	59	-38
MEBO	3.3	57	23	+34
MEGA	3.5	55	25	+30
MEPSO	3.7	53	27	+26
MEMODE	**2.8**	**62**	**18**	**+44**

4.3.6. Discussion

The selection of Differential Evolution as the proposed algorithm for this paper is predicated on its balanced navigation of the search space, its competitive validation performance, and its leading results observed from pairwise win-loss benchmarking. Its controlled balance between exploration (global search) and exploitation (local search) makes the algorithm effective at escaping local optima and converging at high-performing designs. According to the comparative analysis summarised in Tables 4.7, 4.9, and 4.12, the final model evolved using the Differential Evolution algorithm outperformed 1,600 algorithmically generated architectures, in addition to leading competitive models that were implemented. Despite being 67.76% smaller in size than the AlexNet architecture, the elite MEMODE model achieved the highest validation mean F1 score for the chosen multimodal task. Remarkably, the multimodal model evolved through the MEMODE algorithm achieved a greater validation mean F1 score than the largest competing ResNet model, despite containing approximately 2 million fewer parameters and not benefiting from the powerful residual connections offered to the ResNet variants. Similarly, despite the factorised convolutions and compact convolutional modules enabled by the Inception method, the MEMODE model also outperformed the competing Inception model implemented within this research paper. This observation is similarly reflected across MEPSO, MEGA, and MEBO implementations, revealing the competitive performance of algorithmic optimisation, Neural Architecture Search, and Hyperparameter tuning when compared against manually constructed architectures. It is important to note that the performance of metaheuristic algorithms is significantly influenced by their initialised optimisation parameters. Despite the stochastic nature of these algorithms and their dependency on optimal control parameter sets, the range of algorithms developed and tested within this research project demonstrate competitive. Each of the final networks generated by the MEMODE, MEGA, MEBO, and MEPSO algorithms outperformed the fixed and manually tuned architectures on validation and testing datasets to a statistically significant degree, while satisfying the imposed model size limitations.

When investigating the elite fitness of networks generated across each of the implemented algorithms, Differential Evolution outperformed the Bayesian Optimisation, Particle Swarm Optimisation, and the Genetic Algorithm variants, which achieved a validation mean F1 score of 0.8617, 0.8610 and 0.8607 respectively. This

may be attributed to the efficiency of the exploration-exploitation trade-off offered by the DE algorithm, reducing the genetic diversity of the network population over 20 successive generations by 33.02% and eliminating invalid networks from the population by the fourth generation. Notably, the population diversity of the Genetic Algorithm indicates that it underwent a rapid population convergence, decreasing its genetic diversity by 53.96% and eliminating invalid networks from the population by the third generation. This exploitative behaviour contributed to the performance stagnation of the Genetic Algorithm observed in Figure 4.12 (a) and Table 4.7. While the PSO algorithm maintained a significant degree of exploration and an average normalised Euclidian distance of 1.2891, 38% of PSO generated models exceeded the imposed memory constraint of 100 MB. In contrast, DE and GA implementations generated invalid networks at rates of 7% and 3%, respectively. This population analysis reveals that Particle Swarm Optimisation exhibited excessively explorative behaviour, while the Genetic Algorithm exploited its elites and converged on its final elite candidate in its eighth generation, thereby limiting its capacity to generate diverse networks at the upper fitness threshold. In contrast, the DE algorithm continued to improve the fitness of its generated networks over the duration of the optimisation budget. The Differential Evolution algorithm mediated its population diversity while sufficiently capitalising on high-performing networks, demonstrating the competitive performance of Differential Evolution for Convolutional Neural Architecture Search within the context of memory-efficient network design.

4.3.7. Limitations and future research

There are several limitations to the proposed method that must be considered for future research projects. Notably, the parameter sharing feature of the convolutional process required the dimensions of the input channels to be the same, as illustrated in Figure 4.9. Deep learning tasks involving alternative modalities or larger embedding lengths may require these input streams to be separated, which was not required within the scope of this research project. While the combined input is efficient with respect to its memory footprint, tuning the embedding length will also require a specialised architecture that may be investigated in future research. Additionally, the described vectorised matrices generated from Word2Vec embeddings are used to manage text-based captions. As a result, this proposed Word2Vec framework is primarily suitable for tasks relating to Named Entity Recognition (NER), document retrieval, semantic search, and sentiment analysis. Furthermore, while early stopping has been implemented to reduce the computational cost of the optimisation process, the proposed MEMODE algorithm took approximately five GPU days to complete. This significant challenge is inherent to modern large-scale algorithmic network design, which currently relies on limited optimisation runs, reduced population sizes, and lower generation counts. Given the current optimisation framework ($N = G = 20$), future research may seek to improve the proposed implementation by exploring methods to reduce the computational complexity of the multimodal design task, speed up algorithmic convergence, reducing the number of necessary training episodes, or leveraging substitute surrogate functions.

Despite limitations imposed by the Word2Vec matrices, the proposed MEMODE algorithm may still be applied across alternative tasks that also require memory efficient architectures, such as Visual Question Answering (VQA), Phrase Localisation, Image-Text Retrieval, Multimodal Named Entity Recognition, and Content-Based Image Retrieval (CBIR). The competitive performance of the implemented Differential Evolution algorithm described within this paper also indicates that the proposed algorithm may be applied to unimodal architectures for Hyperparameter Optimisation and Neural Architecture Search, in addition to deep learning tasks that require alternative modalities such as video [208], audio [209] or time-series [210] datasets. Analogously, due to the generalised nature of metaheuristics and other black-box optimisation algorithms, the additional competing algorithms developed within this paper (MEBO, MEPSO and MEGA) may also be applied across these various modalities according to the unique requirements of the problem and the desired exploration-exploitation trade-off during optimisation. Notably however, Differential Evolution conventionally requires fewer parameters to be fine-tuned when compared against leading competitive metaheuristics. The selection of the DE mutation factor and crossover rate also intuitively impacts the exploration-exploitation mutation operation. In contrast, Particle Swarm Optimisation depends on the careful selection of distinct parameters that have a less intuitive impact on the exploration-exploitation response, such as the personnel and social particle coefficients, velocity clamping, and the inertia weight. Each of these parameters requires careful selection and experimentation, which is inefficient for optimisation tasks that contain computationally expensive operations. Algorithms like PSO are also typically better suited for continuous optimisation problems and simple fitness functions with smooth gradients. The proposed penalty function generates discontinuities in the fitness landscape that hinder swarm-based convergence.

The algorithms introduced within this paper may also be repurposed within the context of a multi-objective optimisation problem in future research, which would require a Pareto dominance comparison, a method of maintaining the set of non-dominated solutions along the Pareto Front, and a selection mechanism that accounts for Pareto dominance such as Pareto ranking, crowding distance, or other diversity preservation techniques. Furthermore, instead of implementing a penalty coefficient to regulate memory efficiency, constraints may be imposed within the Pareto dominance comparison for comparing feasible models. As this research project was interested in maximising performance while confining the allowable network size to a predefined range, multi-objective Pareto measures were not required for the proposed implementation of MEMODE.

Notably, the performance of the proposed method of memory-efficient optimisation is highly sensitive to the choice of the penalty coefficient k. If the factor is initialised to a value that is too small, the algorithm may produce infeasible network architectures that do not satisfy the desired memory constraints. Alternatively, if the penalty coefficient is too large, search space exploration will be significantly hindered, reducing the performance of converged models. For this research project, a penalisation coefficient of 0.01 was chosen to downscale the fitness function to match the range of values expected for the mean F1 score [0,1], thereby

balancing the variables used within the fitness function. While the sensitivity of the penalisation coefficient was not a significant factor within this research project, alternative deep learning applications may require careful penalty tuning if they use unbound performance metrics that are difficult to anticipate, or if they do not benefit from a clearly defined memory criterion. Furthermore, the effectiveness of metaheuristic algorithms, such as Differential Evolution, relies on the selection of suitable algorithmic parameters, like the crossover rate and mutation factor. In the proposed implementation, these parameters were kept constant during optimisation to ensure fair benchmarking against competing foundational algorithms. This limitation may be overcome by modifying the proposed DE algorithm using variants that adapt their control parameters over the duration of the optimisation process, such as Adaptive Differential Evolution (ADE) [211], Self-Adaptive Differential Evolution (SADE) [212], Adaptive Differential Evolution with Optional External Archive (JADE) [158], Success-History Based Adaptive Differential Evolution (SHADE) [213], and Fuzzy Self-Tuning Differential Evolution [214].

4.3.8. Summary of key findings from Section 4.3

This research introduces a framework for evolving memory-efficient multimodal models to process text and image inputs for multilabel classification. As a population-based evolutionary algorithm, DE uses mutation and crossover modules to search for optimal network architectures and hyperparameter sets, while penalising models that exceed the model size threshold of 100 MB. The final evolved architecture achieved a mean validation F1 score of 0.8621 with a size of 87.77 MB, outperforming 1,600 algorithmically generated models. Furthermore, despite not benefitting from factorised convolutions or skip connections, the MEMODE model also outperformed leading Inception and ResNet models trained within this research project. When benchmarked against alternative NAS algorithms across 10 validation subsets, the MEMODE algorithm achieved the highest average ranking of 2.8, the most pairwise wins (62), and the fewest pairwise loses (18), resulting in the strongest pairwise win-loss differential observed within this research paper (+44). The results within this paper reflect the competitive performance of the proposed Differential Evolution algorithm for the given multimodal task, converging towards highly efficient model architectures and hyperparameter sets that satisfied a predefined model size limitation. Notably, the Particle Swarm Optimisation, Bayesian Optimisation, and Genetic Algorithm variations that were developed within this paper also outperformed traditional competitive networks, demonstrating the value of metaheuristic algorithms and black-box solvers as valuable tools to automate the design of memory-efficient multimodal architectures and hyperparameter sets.

4.4. Conclusion

This chapter documents the benchmarking and development of innovative hybrid differential evolution algorithms, which utilise complementary features of metaheuristics and neural networks to reduce the impact of human parameter selection on design performance. Remarkably, the demonstrated capacity of neural networks to mediate the exploration response of the differential evolution algorithm indicates the potential for

autonomous algorithmic initialisation and dynamic hyperparameter adaptation during the optimisation process. Future research may seek to incorporate additional observations and actions within the proposed reinforcement learning framework to further improve algorithmic performance and reduce the theoretical background knowledge required to effectively apply metaheuristics across industry projects. This practical interdisciplinary design focus is expanded upon through the development of the MEMODE algorithm, which allows machine learning practitioners to define the memory requirements of the final model design — directing the evolution and convergence of the DE algorithm according to real-world task specific inference memory requirements. By grounding this chapter in addressing practical challenges surrounding autonomous hyperparameter selection and control, this research aims to significantly reduce the impact of human error, human subjectivity, and human inefficiency within the context of algorithmic design.

CHAPTER 5: SCALING PREDICTIVE STRUCTURAL DESIGN MODELS FOR IMPROVED PRECISION AND SCENE COMPLEXITY USING TRANSFORMER HYPERNETWORKS

5.1. Introduction

The interdisciplinary opportunities between hybrid metaheuristic and machine learning systems may also be revealed when examining the limitations that have inhibited the field of predictive design, which has been largely characterised by object-based training datasets and academic model frameworks. Importantly, open-source datasets such as ShapeNet and PartNet have been used to develop foundational predictive approaches and benchmark the reconstruction accuracies of models such as convolutional voxels, mesh deformations, and hypernetworks. While these datasets are incredibly useful from a benchmarking perspective, they are not representative of the scene complexity required for practical design applications. Consequently, there is also very little published research that proposes potential solutions to scale existing foundational models to meet the demands of practicing architects and engineers. This is illustrative of a pervasive void between academia and industry, which this thesis argues has significantly inhibited the broader adoption of machine learning innovation throughout the built environment. The growth of the predictive design field beyond academic contexts must be driven by research that aims to directly reconcile these challenges. This chapter introduces an interactive 'text-to-structure' predictive framework that encourages an iterative text-based design process that is particularly suited for rapid preliminary modelling and massing — grounding the model design with a focus on practical user interactions. Section 5.3 expands upon this strategy through novel model-predicted translation vectors which manipulate geometric outputs to produce scenes with hundreds or even thousands of interconnected structural elements — vastly improving the representation capacity of foundational reconstruction models.

5.2. TRANSFORMER HYPERNETWORKS FOR TEXT-TO-STRUCTURE DESIGN

5.2.1. Summary

This research presents a novel 'text-to-structure' approach for generating structural models from text-based design prompts. The proposed architecture consists of a fine-tuned transformer model and a deep neural hypernetwork tasked with mapping design criteria to a tensor of weights that are assigned to a primary network. Structures are constructed by the primary network by extracting negative signed distance predictions from a collection of queried points within the design space. As the output geometry is produced from the predicted implicit neural field output by the primary network, the resolution and shape of the generated structure are theoretically unlimited, enabling the potential application of this method across a diverse range of structural engineering problems and design tasks. This adaptability in structural representation also allows for the proposed hypernetwork model to learn from traditional optimisation methodologies and previously generated

structures, as a computationally efficient method of capturing large model databases for inference. Applications demonstrated within this section range from two-dimensional and three-dimensional truss design, gridshell forms, and structural component prediction within an ensemble hypernetwork arrangement. The code is available at: https://github.com/thie7042/Text-To-Structure-Hypernetworks

5.2.2. Introduction

This chapter introduces a novel method for producing manifold structures from text-based design criteria, training from datasets that draw from a diverse range of traditional optimisation strategies. This has been referred to as a 'text-to-structure' methodology within the context of this research, utilising the compelling representation capacity of hypernetworks and the semantic mapping of transformer encoders. The term 'hypernetwork' was introduced by Ha et al. [109], inspired by advances in Evolutionary computation and the HyperNEAT algorithm, and is typically defined by two distinct networks. One network, referred to as the hypernetwork, is tasked with taking a representative input of the target geometry and predicting a set of weights for a secondary network, referred to as the primary network. While hypernetworks have been used for 3D shape reconstruction and generation, their application within the fields of structural engineering and architecture for design purposes has not been explored. Furthermore, to our knowledge, the potential application of transformers is yet to be fully realised for text-to-geometry tasks, with previous hypernetwork implementations adopting a convolutional architecture and image-based inputs to represent the desired geometry [119, 215].

Recent state-of-the-art advances in natural language processing such as the Bidirectional Encoder Representations from Transformers (BERT) model have demonstrated competitive performance across a broad range of tasks, with an emphasis on question answering [216, 217], classification [218, 219] and natural language inference (NLI) [220, 221]. As a pretrained transformer-based architecture, BERT leverages the self-attention mechanism to calculate attention scores for each token within the input sequence, determining the weighting of individual tokens relative to its context. Multiple attention heads within the BERT architecture allow the model to capture semantic and contextual relationships in the input data, enhancing the representation capacity of the final output embeddings [222]. Despite its recent adoption across a diverse range of natural language processing tasks in recent years, the application of transformer encoders such as BERT within hypernetwork architectures remains a significant gap within the available literature. This presents a unique opportunity to investigate the performance of transformer-based hypernetwork architectures to construct designs from user text inputs.

The compelling performance of hypernetworks and transformer models such as BERT provides a unique opportunity to explore novel solutions that empower designers and engineers. By fine-tuning a lightweight transformer encoder for the named entity recognition (NER) task of extracting design parameters from a text-

based user input, the proposed model pipeline has been developed as an interactive tool for rapid inference that complements preliminary design experimentation. The proposed 'text-to-structure' model pipeline also offers an automated design workflow that is trained on bespoke structural design datasets, allowing engineers to develop and iterate over predicted structures according to highly specific design requirements. A broad range of optimisation strategies have been used to generate these structural datasets, furthermore demonstrating the versatility of hypernetwork reconstruction. Once trained, these hypernetworks are capable of predicting 2D and 3D geometries of unseen structures based on learnt relationships between the input design parameters that are difficult for designers to intuitively detect *a priori*. Changing parameters such as the structural configuration, geometric depth and width, loading conditions, or node boundary conditions can have a profound effect on the final design. These relationships can be learnt and accurately predicted by the proposed hypernetworks, allowing designers and engineers to rapidly iterate over preliminary design ideas and derive deeper insights into the patterns that emerge between interdependent design variables.

Within the context of this research, structures are represented as continuous functions referred to as implicit neural fields, which are taken as either binary or signed distance values that have been predicted by a neural network. The proposed implicit neural fields enable high-resolution and continuous reconstructions of complex structural topologies that are conditioned by user defined design requirements for a given design task. The experiments conducted within this research examine the capacity of hypernetworks to learn complex two dimensional and three-dimensional implicit neural field structural representations, using a diverse range of structural parameters that were selected based on common preliminary design practices. These structural designs have been optimised using a broad range of traditional optimisation techniques that are characterised by their distinct topological limitations or lack thereof, demonstrating the capability of hypernetworks to bridge the gap between academic optimisation subfields that have largely remained data-independent and distinct. The text-based input used to drive the proposed 'text-to-structure' method also demonstrates the capacity of these models to learn from a wide range of design requirements, using input criteria relating to member sizing, structural form, boundary nodes, and loading conditions.

To account for complex design tasks that require the capacity to model and isolate a range of distinct structural elements, this research project also introduces ensemble text-to-structure hypernetworks. Within this context, an ensemble model refers to the combination of multiple hypernetworks that are each trained to predict the form of specific structural elements that cumulatively contribute to the final structure. By training individual hypernetwork models to learn specific subcomponents of a given design, the outputs from each network can be meshed separately and assembled to construct a structural form composed of a variety of classes. This process improves the capacity of the method to capture complex interconnected structural elements and allows the final output geometry to capture class distinctions that are essential from a design perspective. This is reflected within the concrete girder bridge design task chosen for this research project, in which the ground truth bridge models

were divided into foundations, columns, crossbeams, girders, abutments, decks, and parapets. An ensemble arrangement was then implemented, consisting of seven individually trained hypernetworks responsible for reconstructing each of the structural components, allowing the final predictive output to consist of a range of distinct meshes with individual class labels. This furthermore allows the final output geometries to be rendered with different material properties and manipulated individually according to the needs of the chosen design task.

5.2.2.1 Research Significance

Since its inception, the field of structural optimisation has developed a collection of highly specialised techniques that, due to their unique advantages and limitations, occupy distinct subdisciplines within academia. Historically, metaheuristic algorithms such as Differential Evolution, the Genetic Algorithm, and Particle Swarm Optimisation require the discretisation of designs into a collection of parameters which are initialised within a confined design space [223-225]. This inherently limits the scope of these algorithms to schematic, shape, and sizing optimisation problems. Furthermore, these methodologies are largely data independent, relying purely on search-space navigation and the querying of a chosen objective function for each design task [226]. In this sense, the traditional application of these algorithms cannot learn from other methods, or even from past optimisation results generated by the same underlying process. This research project is aimed at introducing a novel text-to-structure transformer-based hypernetwork model, leveraging datasets that have been generated using conventional optimisation algorithms.

We posit that the distinct representation capability of the proposed hypernetwork method offers a unique opportunity to unify the historical results that are generated across optimisation disciplines. By representing designs through the predicted decision boundary of a binary classification task or signed distance function, the hypernetwork method can replicate topological forms present within the field of structural optimisation with theoretically unlimited resolution [227]. This is unique, as conventional optimisation methods sacrifice constructability for topological freedom or vice-versa, without the capability of adapting this trade-off according to the needs of a given task. Conventional SIMP approaches are incapable of producing uniform cross-sectional elements in the same discrete manner as Differential evolution, while DE is restricted to the standardised parameters encoded prior to optimisation. In contrast, hypernetworks can learn from a diverse range of structural datasets and can condition the output form according to commands from the user. As traditional optimisation methods provide a well-defined and repeatable method for converging on optimal designs, they are highly suitable for the purposes of data generation when curating a hypernetwork training dataset. By emphasising convergence consistency, the variance present within the training dataset may be directly controlled, mitigating one of the leading challenges in training hypernetworks. The distinct optimisation strategies discussed within this research offer unique methods to generate consistent structural topologies as training data for a given structural task, improving the capacity of the neural network to map design criteria to weight tensors.

The purpose of this research is not to undermine the viability of traditional optimisation techniques, but to demonstrate that they each occupy unique and disparate spaces within the field of structural optimisation. The proposed hypernetwork method provides an opportunity to leverage data generated across these traditional optimisation techniques, in a unified text-to-structure model that is not limited by topology, resolution, or dimensionality. In this sense, by developing a database of optimal solutions through conventional optimisation methods, implicit neural representations of designs may be learnt via the hypernetwork method. These may be generated at inference time without the need to directly query an objective function or navigate the search space, allowing for rapid iteration during the preliminary design phase. From an architectural perspective, this allows designers to visualise structures that are potential solutions for a given set of design requirements, enriching preliminary planning and allowing for rapid and interactive design iteration. From an engineering perspective, by training hypernetwork models off historic designs, engineers may leverage a collective database of previous projects to iterate through alternative solutions for future preliminary design tasks.

Due to its generalisation capabilities, resolution capacity and flexibility in topological representation, the potential application of the proposed transformer-based hypernetwork architecture extends to a broad range of design fields, including iterative design experimentation, architectural and engineering modelling, rendering, computer aided design, additive manufacturing, and 3D printing. By sampling the implicit neural field generated from the hypernetwork and converting the designs to polygon mesh-based geometry, structures generated through this novel method may be directly introduced into existing computer graphics and computer-aided design software, allowing the proposed model to be adopted within conventional design workflows. Designs constructed using hypernetworks also offer a lightweight approach to rapidly experiment and iterate through various design alternatives, allowing for a highly memory efficient and compact method to reproduce structures from asset databases that may be orders of magnitude larger than the hypernetwork itself. Given the broad potential application of the proposed model to synthesis training data across diverse datasets into a consolidated design methodology and integrate into existing design practices, hypernetworks provide a unique opportunity for engineers and architects that is yet to be fully realised.

5.2.3. Methods and Techniques

5.2.3.1 Data collection and representation

The hypernetwork models utilised within this research require a collection of either 2D or 3D coordinates that are sampled from the ground truth design space, with each point capturing either a binary label or signed distance value. Across a range of design tasks, collected designs have been scaled to fit within a unit cube such that the coordinates of sampled points are normalised within the range [0-1] for each axis. During the training process, the performance of the hypernetwork f is evaluated for two-dimensional tasks according to the binary cross-

entropy loss function, with sampled points labelled across two distinct states [228]. The output geometry is represented by the predicted state of each point p as either existing within (1) or outside (0) of the structure. The binary cross-entropy loss of trained hypernetwork weights θ_f given some set of design criteria D is calculated by the following equation:

$$H(\theta_f, D) = -\frac{1}{n}\sum_{i=1}^{n}(y_i \cdot \log(g(p, f(D, \theta_f))) + (1 - y_i) \cdot \log(1 - g(p, f(D, \theta_f)))) \quad (5.1)$$

where the hypernetwork f outputs the weights of the primary network g according to the input design criteria D. The ground truth tensor y_i represents the target state of each point p for sample i within a batch of size n. Points are sampled for 2D structural problems within the normalised domain $(x, y) \in \{(x, y) | \, 0 \leq x \leq 1, 0 \leq y \leq 1\}$, while 3D structural engineering design tasks require the sampling of points within the range $(x, y, z) \in \{(x, y, z) | \, 0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1\}$.

Beyond binary representations, recent works have also explored the application of Signed Distance Functions (SDF) to capture the distance between each sampled point within the design space to the target manifold geometry [229-231]. Signed distance functions provide a continuous representation of the 3D structure, encoding the distance between any point in the design space and the target structure's surface. The surface itself is captured within the zero-level set, in which the SDF value is exactly zero. Points with a negative SDF value lie within the enclosed volume of the underlying structure, while sampled points with a positive SDF value lie outside the structure within the design space. Signed Distance Functions provide a structural encoding strategy that informs the proximity of all points relative to the target geometry, while binary representations offer a discrete categorisation of each point as either existing inside or outside the structure. An example Signed Distance Function heatmap is visualised in Figure 5.1 (b), illustrating the implicit neural field that the proposed hypernetwork architecture aims to reproduce. Signed Distance Functions provide a holistic representation of the target geometry and the surrounding design space, which translates exceptionally well to the training of hypernetworks [232]. Within this research, both binary and SDF representations have been examined across various structural tasks. The loss function utilised for these applications is the mean absolute error between the predicted signed distance and the ground truth dataset, calculated through the following equation:

$$H(\theta_f, D) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - g\left(p, f\left(D, \theta_f\right)\right)\right| \quad (5.2)$$

Once the optimal designs were established for each design task using a variety of optimisation algorithms, the structures were sampled to collect either binary or SDF datasets. A combination of random sampling and boundary-proximate sampling was used to capture a point-based representation of the desired geometry, visualised in Figure 5.1 (a). Points that are closer to the boundary surface of the structure provide more useful information regarding the detailed shape of the converged solution, while randomly sampled points allow for a more general depiction of the prediction space.

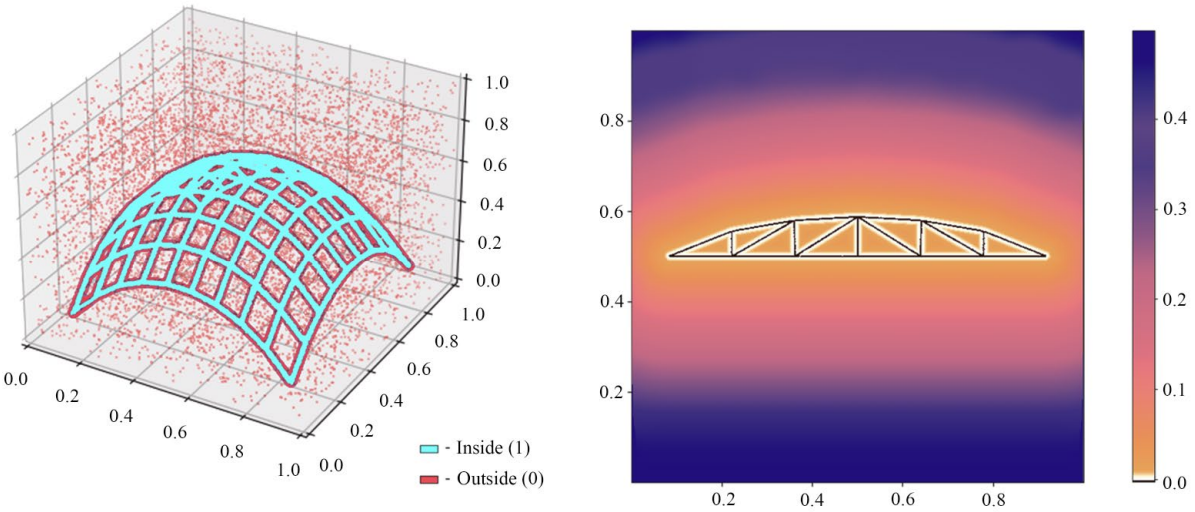


Figure 5.1 (a) – Binary sampling example, Figure 5.1 (b) – Signed Distance Function example

5.2.3.2 Marching Cubes Algorithm

To apply the proposed model pipeline within a traditional modelling design context, a manifold geometry must be extracted from the predicted implicit neural field. Final mesh designs are produced after training by passing a grid of points to the primary network according to the desired resolution. From these structured points, a manifold surface is generated using the Marching Cubes algorithm from predicted internal and boundary points [233]. The Marching Cubes algorithm is a sequential-traversal method that is widely applied within the field of 3D reconstruction, due to its straightforward and structured processing of localised boundary details [234, 235]. This process effectively divides the design space into a set resolution of cubes, often referred to as voxels. If a given voxel intersects with the surface of a scalar field, the algorithm determines an appropriate triangle configuration to approximate the boundary surface using a precomputed lookup table. Once complete, these triangles are assembled and smoothed to form a mesh-based depiction of the primary network's implicit neural field. An example of this mesh-reconstruction process from sampled hypernetwork implicit neural fields is provided in Figure 5.15. Further documentation of the Marching Cubes algorithm is provided in Appendix C of this thesis.

5.2.3.3 Model Architecture

The proposed methodology consists of three distinct signal-processing blocks; the transformer module, the hypernetwork, and the primary model. The first region of the signal processing takes a tokenised representation of the user's input, which is passed to the BERT architecture to capture the semantic meaning of each token ID within the input. Positional embeddings are added to the word embeddings to provide positional information regarding the order of the input tokens, as transformer-based models are driven by self-attention and therefore do not directly inherit structural information through recurrently processing the input signal as LSTM networks

do. These tensors are then processed by several stacks of transformer encoder blocks, which themselves consist of a multi-head self-attention module followed by feedforward neural networks. The base pretrained BERT model used within this model pipeline consists of 12 transformer blocks [236], which are fine-tuned according to the Named Entity Recognition (NER) task of extracting the relevant design criteria from the user's input. These structural specifications vary according to the given design task, including requirements such as the span, width, loading conditions, and structural form.

The extracted and normalised design criteria D are then processed by a fully connected hypernetwork, tasked with mapping the numerical design specifications into a tensor of weights for the primary network. For all applications tested within this research, the hypernetwork f consists of five fully connected layers, the first four of which consist of 1024 neurons and utilise the ReLU activation function. The final activation is a hyperbolic tangent function (Tanh), effectively centring the signal at zero with values ranging between -1 and 1. For the three-dimensional variation, the hypernetwork output is a tensor of 3297 weights, which each map to a parameter within the primary network. The hypernetwork contains the only set of trainable parameters θ_f required within the proposed pipeline, as the weights of the BERT transformer were fine-tuned separately. Notably, smaller hypernetwork architectures that improve inference speed are often limited by a reduced reconstruction accuracy, limiting the structural complexity that smaller networks are capable of learning from [237]. The primary network consists of five layers, the first four of which use ReLU activations and contain 32 neurons that inherit their weights and biases from the hypernetwork model. The primary network either takes a 2D or 3D coordinate input depending on the given task, outputting a single value that serves as either a binary classification or a prediction of the signed distance value at the given point. The hypernetwork text-to-structure process is broadly outlined in Figure 5.2, including the architecture implemented across each design task.

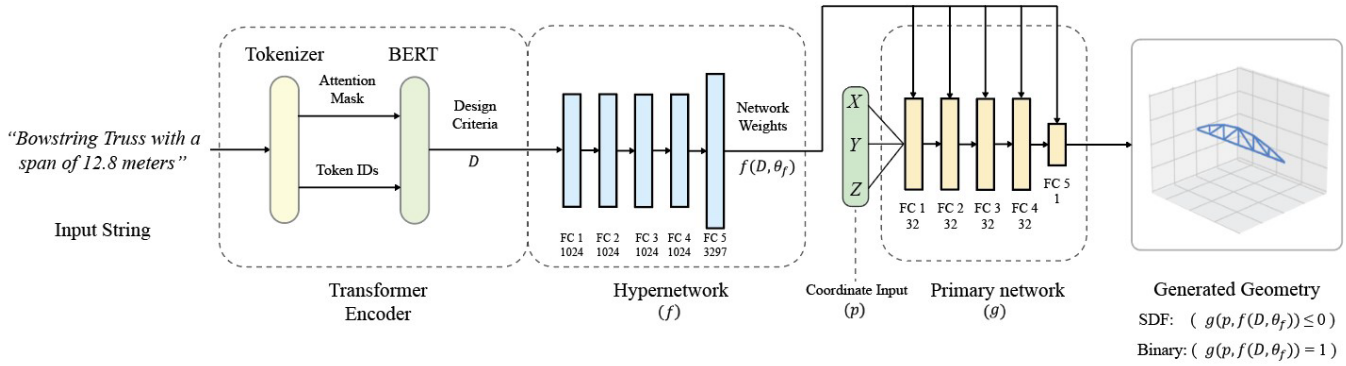


Figure 5.2 – Proposed text-to-structure hypernetwork architecture

The primary network generates a neural representation according to the grid of input points, producing the predicted shape of the target structure for the given user input. Notably, the decision boundary between points existing within or outside of this structure defines the topology of the generated solution, such that the resolution of the final design is entirely dependent on the chosen grid size. As the decision boundary is learnt during the

training process, it can theoretically take on any desired shape present within the training data given adequate sampling, sufficient model capacity, and an appropriate training method. This is demonstrated within this research through the diverse range of topologies produced by the proposed hypernetwork method for each given design task. While implicit neural representations do not have a direct resolution limitation, the visualisation of the three-dimensional design applications discussed within this research require the final output structure to be processed as a polygon mesh-based geometry. To extract a mesh-based geometry from the hypernetwork output, the sampled points are first converted into a scalar field representation, before being processed by a Marching Cubes and mesh smoothing algorithm.

To evaluate the performance of the proposed hypernetwork methodology for three-dimensional tasks, the Intersection over Union (IoU) metric has been used to calculate the similarity between the structures extracted from the primary network's implicit neural fields and the ground truth training geometry [238]. Intersection over Union is a performance metric that is widely used to evaluate the accuracy of models within the field of 3D object construction, providing a scaled measurement of the spatial overlap between the generated output and the ground truth geometry. The popularity of IOU is partially due to its scale-invariant normalisation, penalisation of false positives, interpretability, and consistency across benchmark tasks. These advantages have contributed towards its extensive use as a defining benchmark metric used to evaluate the reconstruction accuracy of novel algorithms and deep learning models [239-242]. For two-dimensional examples, the Intersection over Union between the predicted structure and the ground truth design is simply calculated on a pixel basis. The IOU value is calculated by measuring the volume of a predicted structural element against the target geometry as follows:

$$IoU = \frac{|V_{pred} \cap V_t|}{|V_{pred} \cup V_t|} \tag{5.3}$$

where V_{pred} is taken as the volume (or area) of the predicted structural element and V_t is the equivalent measurement for the ground-truth target geometry. The intersection between these two geometries ($\cap$) is divided by their corresponding union ($\cup$) to provide a standardised measurement of hypernetwork performance.

5.2.4. Experiments

To analyse the performance of the proposed text-to-structure method, six unique design tasks were developed across a range of unique topological forms and design requirements. A total of 54,596 individual structural mesh models were created within this research project across the six design experiments, using a diverse range of strategies including Differential Evolution, Genetic Algorithms, Solid Isotropic Material with Penalisation, and parametric CAD models, demonstrating the versatility of the proposed hypernetwork method to learn from a variety of training structures. The distribution of these models is entirely dependent on the complexity of the design task, the dimension of the output geometry (2D/3D), the number of design parameters that define the

underlying structure, and the range of possible values that these parameters can take. The specifications of the ground truth dataset are further documented in Table 5.2, while the structural parameters and constraints used across each experiment are provided in Table 5.1. All design tasks utilising a finite element approach have been analysed under the assumption of static equilibrium, such that time-dependent effects are not considered. A consistent data partitioning strategy was used across each experiment, in which 80% of structural models were reserved for training, 10% for validation, and 10% for testing. A learning rate of 2×10^{-5} was used for each hypernetwork implemented within this training framework, using the ADAM optimiser with a batch size of 10. Each structural model developed for training, validation, and testing purposes was assigned a unique text-based user command, containing the design criteria specific to that model. These user commands serve as the basis for the NER task executed by the transformer region of the text-to-structure architecture, after which the extracted design criteria are normalised and passed to the hypernetwork model. Examples of these user commands and the corresponding network outputs are provided in Figures 5.4, 5.6, 5.8, 5.10, 5.12, and 5.14.

Table 5.1 – Design specifications for each task

Task	**Dimension**	**Imposed Load**	**Design Limitation**	**Objective Function**
Cantilever - DE	2D	10 kN	$\Delta_{(x,y)} \leq 64\ (cm)$	Structural mass minimisation
Cantilever - SIMP	2D	1 kN	$\kappa = 0.3$, $p = 6$	Compliance minimisation
SSB - SIMP	2D	1 kN	$\kappa = 0.5$, $p = 1$	Compliance minimisation
Truss	3D	2.4 kN/m	L/180	Structural mass minimisation
Gridshell	3D	4.8 kN/m	L/360	Structural mass minimisation
Concrete Girder Bridge	3D	-	-	-

Table 5.2 – Ground truth element breakdown

Design Task	Structural Element	Dimension	Design Parameters	Parameter Ranges	Structural Variations
2D Cantilever Truss (DE)	Truss	2D	Loading position	[50-300] (x,y)	208
2D Cantilever Truss (SIMP)	Truss	2D	Depth, Span	[20-100], [20-71]	4210
2D SSB	Beam	2D	Boundary conditions, Depth, Span	[Fixed/Pinned], [20-140], [120-840]	3650
Form Conditioning	Truss	3D	Form, Span	[Pratt, Bowstring], [5-15]	410
Gridshell	Gridshell	3D	Span	[5-15]	310
Bridge	Parapet Girders Foundation Deck Crossbeam Column Abutment	3D	Span, Width (Bridge)	[30-60], [8.5-16]	6544

5.2.4.1 2D Cantilever Truss Design

The first two-dimensional structural optimisation problem evaluated within this experiment consists of a fully connected 3×3 truss matrix, cantilevered with fixed supports. Each non-boundary node has two axes that can be offset by an entire span length, while the 39 bars within the truss itself are added or removed during the optimisation process. A total of 63 structural parameters were tuned using a Differential Evolution algorithm to generate an optimal design for each loading position, with the objective function defined to reduce the overall weight of the structure. The generalised framework for this 2D design task is illustrated in Figure 5.3. For each training solution, the position of a 10 kN downward point load was first randomly defined within a cantilevered matrix. The closest node to this point was then fixed to these coordinates to ensure that the converged design was subject to the load. The optimisation loop was then run for 1,000 iterations, with each simulation itself being repeated 100 times to ensure consistency within the training dataset. The loading position along with the parameters of the solution with the greatest performance were then recorded, allowing for the designs to be reconstructed for sampling purposes. The design displacement limitation ($\Delta_{(x,y)}$) for this optimisation problem is outlined in Table 5.1, in addition to the criteria and objective function for each experiment discussed within this section. The structural analysis was conducted for each truss candidate using the direct stiffness method, using a Young's modulus of $E = 68.95 \times 10^3$ MPa and a material density of 2,768 kg/m^3. The hypernetwork trained to learn these optimised structures was trained over 20,000 epochs using the normalised load location as its input. Examples of these hypernetwork structures for various descriptions of loading positions are provided in Figure 5.4, in which the greyscale predicted truss is compared against a red overlay outline of the target geometry.

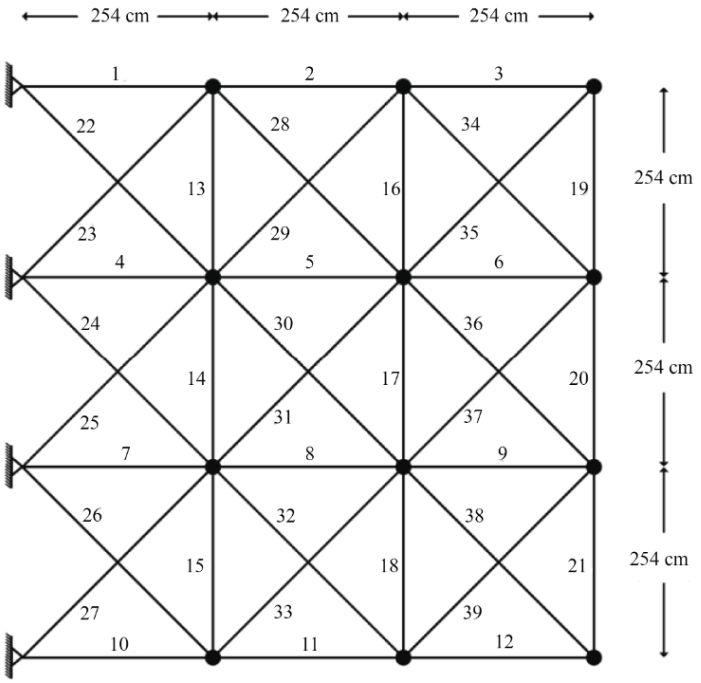


Figure 5.3 – Uniform cantilevered truss design task

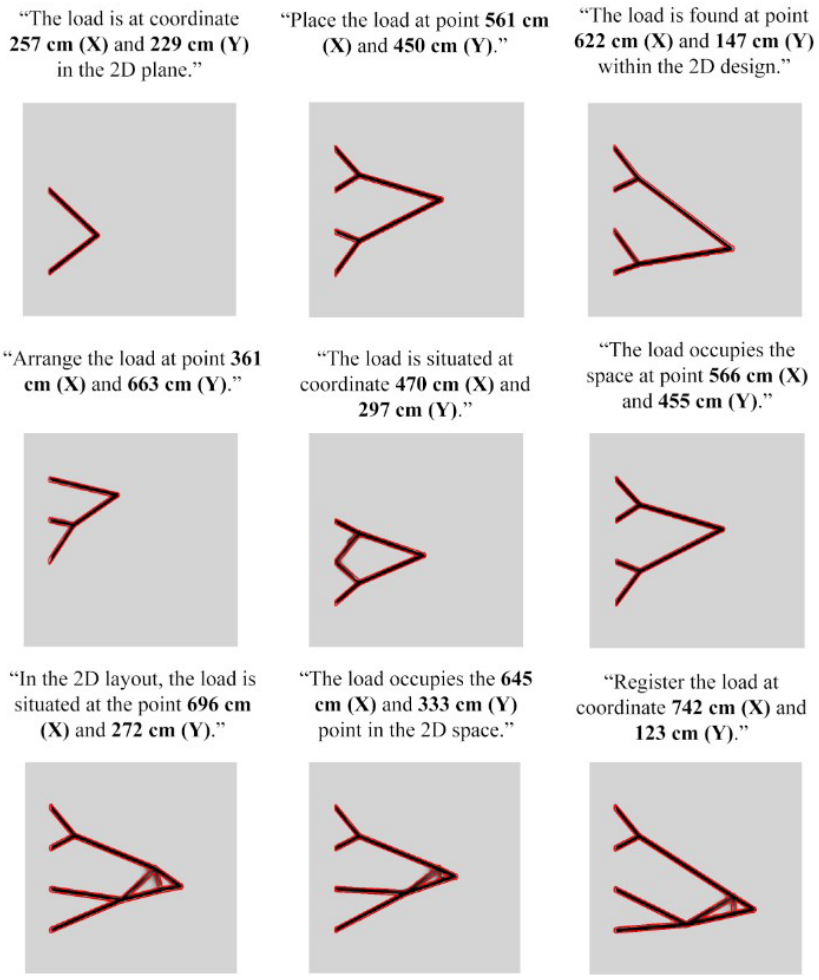


Figure 5.4 – Hypernetwork cantilever trusses for various descriptions of loading positions

To demonstrate the versatility of the hypernetwork method, a similar cantilevered truss framework was established using the Solid Isotropic Material with Penalisation (SIMP) approach. This is a method of topology optimisation, in which material distribution within a specific design space is optimised for a given objective function [243]. In this example, the design domain consists of a grid of 2D bi-linear finite elements cantilevered from a fixed axis, with a defined number of cell elements in the horizontal N_x and vertical N_y directions. A downward load $P = 1$ kN was applied at the centre of the cantilevered edge. The objective function for this design task involved the minimisation of the structural compliance, while satisfying a specified volume constraint. For each 2D truss generated using the SIMP method, the volume fraction (κ) was initialised as 0.3 and the penalty scalar factor (p) was set to 6. To align this task with established SIMP cantilever benchmark problems and prevent the stiffness matrix from becoming singular, a unit Young's modulus and a Poisson's ratio of $v = 0.3$ were used [244, 245]. An illustration of the general design conditions for this task is provided in Figure 5.5. As a benchmark application of SIMP optimisation, the method of moving asymptotes (MMA) was used to optimise these design parameters and update the density variables within the design space over a maximum of 200 iterations [246]. The design criteria passed to the hypernetwork to construct trusses that satisfy these conditions included a description of the cantilevered design space, including the number of bi-linear cell elements in the x (width) and y (height) axis. Example trusses reconstructed by the trained hypernetwork model

for various descriptions of cantilever sizing are provided in Figure 5.6, with the red outline of the target geometry overlayed in red.

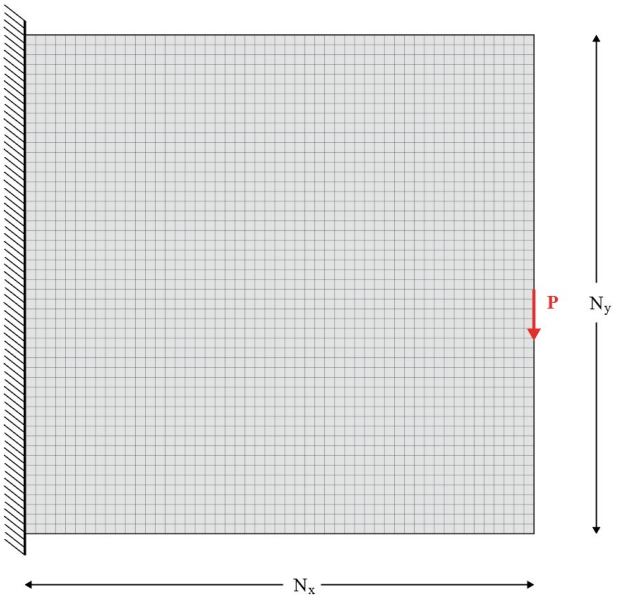


Figure 5.5 – Freeform cantilevered truss design task

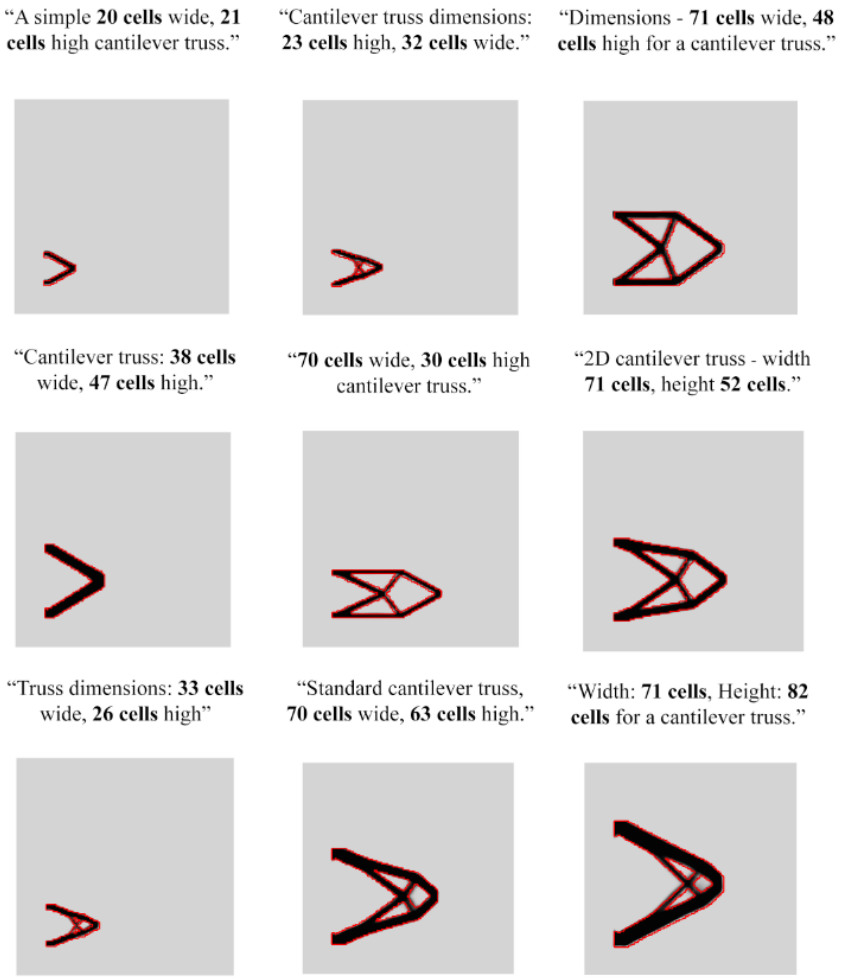


Figure 5.6 – Hypernetwork structures for various descriptions of cantilever sizing

5.2.4.2 2D Simply Supported Beam Design

The proposed hypernetworks offer a unique and versatile method for learning underlying patterns present in datasets of optimised structural models. To demonstrate the flexibility of these hypernetworks, an additional 2D design task was developed under the constraints of a symmetric simply supported beam. This task is driven through cell-based structures that were generated using the Solid Isotropic Material with Penalisation (SIMP) approach, consisting of 4 potential boundary nodes at each corner of the base beam and an array of cells that can vary along the x and y axis. Optimised structures were labelled according to their depth (N_y), span (N_x), and boundary conditions, which can be defined by the user at inference through a text-based input command. To establish an appropriately sized design domain, structures were generated in increments of 5 cells, with spans ranging from 120 to 840 cells, and depths ranging from 20 to 140 cells, including all symmetric SSB boundary variations. 4-node quadrilateral elements were used for this SIMP-based finite element analysis, developed as a 2D design problem with 2 degrees of freedom per node. The material properties have been consistently defined across SIMP-based design tasks. A point load of 2 kN was positioned at the top surface of the beam, halfway between the boundary support nodes. Notably, this problem may be simplified using the method of sections and equivalent force systems, symmetrically bisecting the beam at its midpoint. The objective function implemented for this task required compliance minimisation, subject to a volume fraction constraint (κ) of 0.5, a filter radius of 1.5, and an initial penalisation faction of 1, with each structure optimised over 50 iterations to develop the final database for training. For this task, reciprocal approximation and a grey-scale filter (GSF) were used to process the optimised beam structures during the dataset generation process. The base experimental framework for this optimisation task is outlined in Figure 5.7, while the predicted hypernetwork outputs for given user commands are provided in Figure 5.8. For each of these predicted output heatmaps, the target structural geometry is outlined as a red overlay.

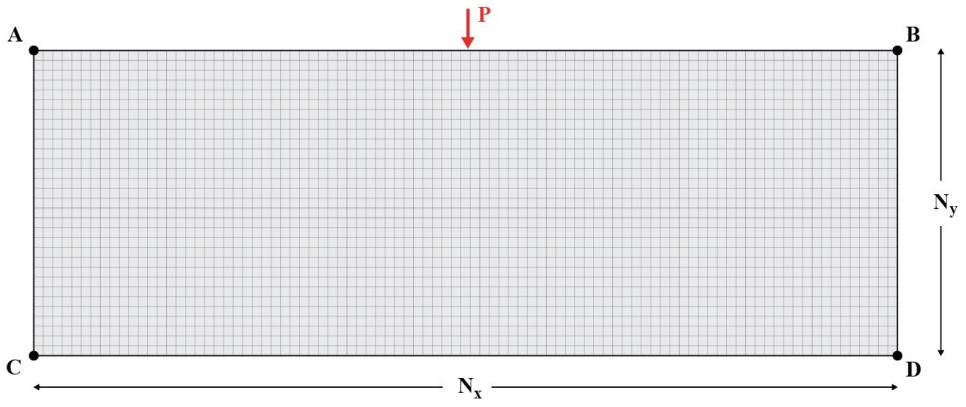


Figure 5.7 – Simply supported beam design task

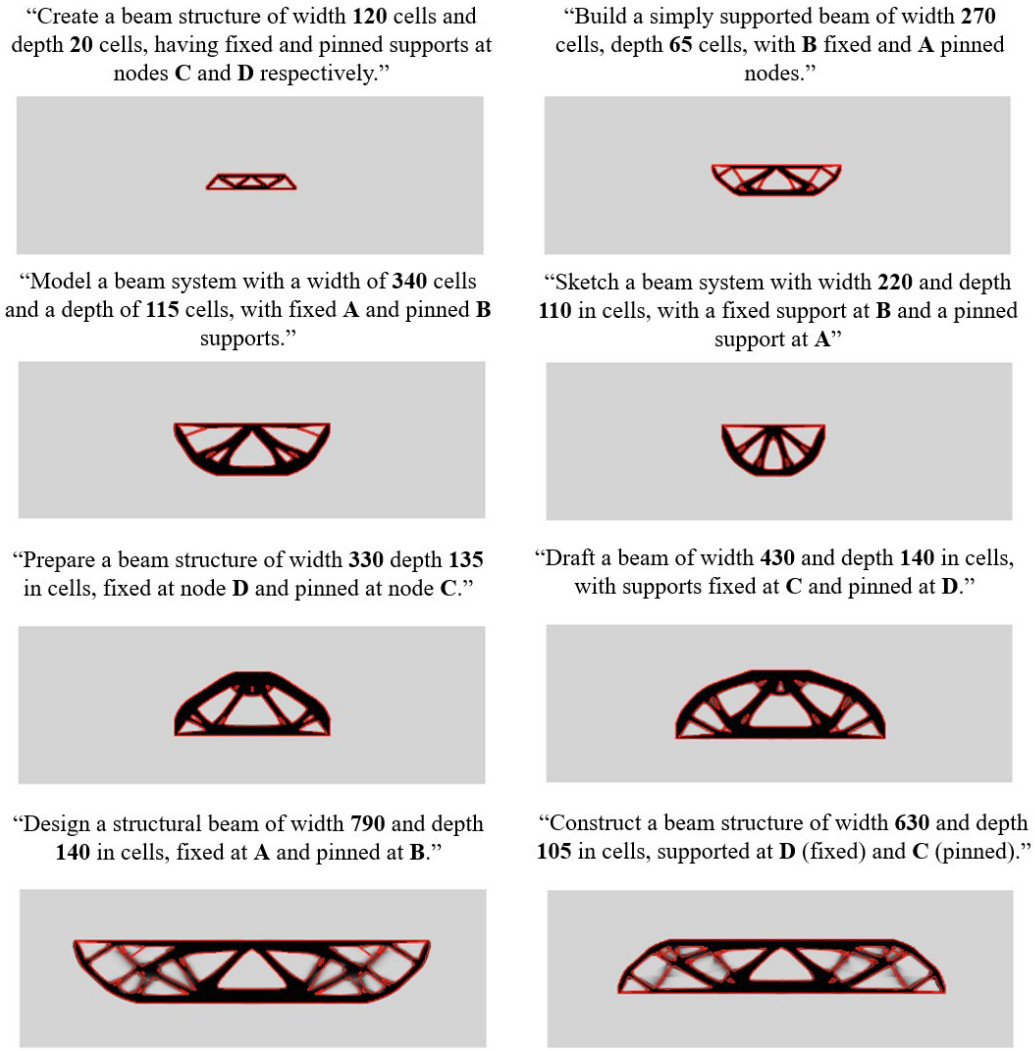


Figure 5.8 – Hypernetwork structures for SSB user requirements

5.2.4.3 Truss Form Conditioning

The representation capabilities of implicit neural fields extend beyond two-dimensional models, providing a unique opportunity to apply the proposed text-to-structure hypernetwork method within the context of three-dimensional geometries. For all three-dimensional output structures examined within this section, a hypernetwork model was developed to process points sampled from a 3D unit cube, utilising the Signed Distance Function to represent the underlying ground truth geometry. Truss form conditioning was selected as the first design task to investigate this methodology. Optimal truss designs were developed for Bowstring and Pratt truss variations using the Galapagos Genetic Algorithm interface [65], subject to a uniformly distributed load of 2.4 kN/m and a deflection limitation of L/180. The structural response of each generated truss was calculated using the Karamba3D finite element analysis toolkit, which implements first-order calculations of compressive and tensile forces within the truss in addition to the resulting displacements [247]. The truss structural elements themselves consisted of hot rolled round steel bars, with a standardised diameter of 100 mm and a Youngs modulus of 210×10^3 MPa. Each optimal truss design was assigned a text-based command

containing the desired structural form and the span of the truss, ranging between 5 – 15 meters. These inputs represent the potential prompts that may be provided to the hypernetwork model within an interactive 'text-to-structure' design context. To numerically distinguish the requested form of the structures, extracted 'Pratt' commands were mapped to a value of 1 while 'Bowstring' requests were assigned a value of -1 before being included with the normalised span as inputs for the hypernetwork. The hypernetwork itself was trained over 50,000 epochs before being benchmarked within the proposed text-to-structure model framework. Example structures generated by the final text-to-structure hypernetwork are summarised in Figures 5.9 and 5.10.

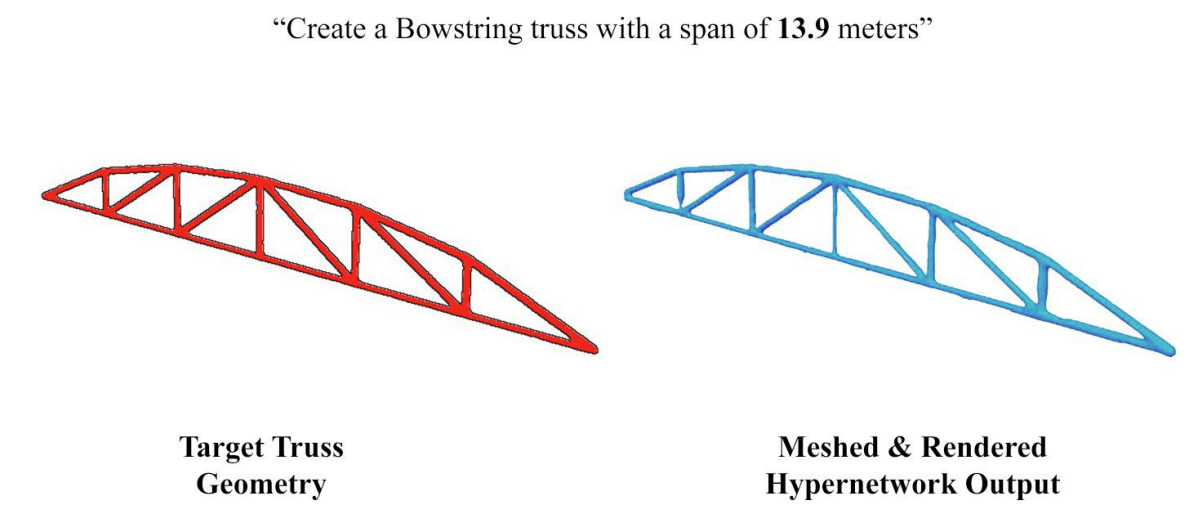


Figure 5.9 – Example comparison of ground truth truss and meshed hypernetwork reconstruction

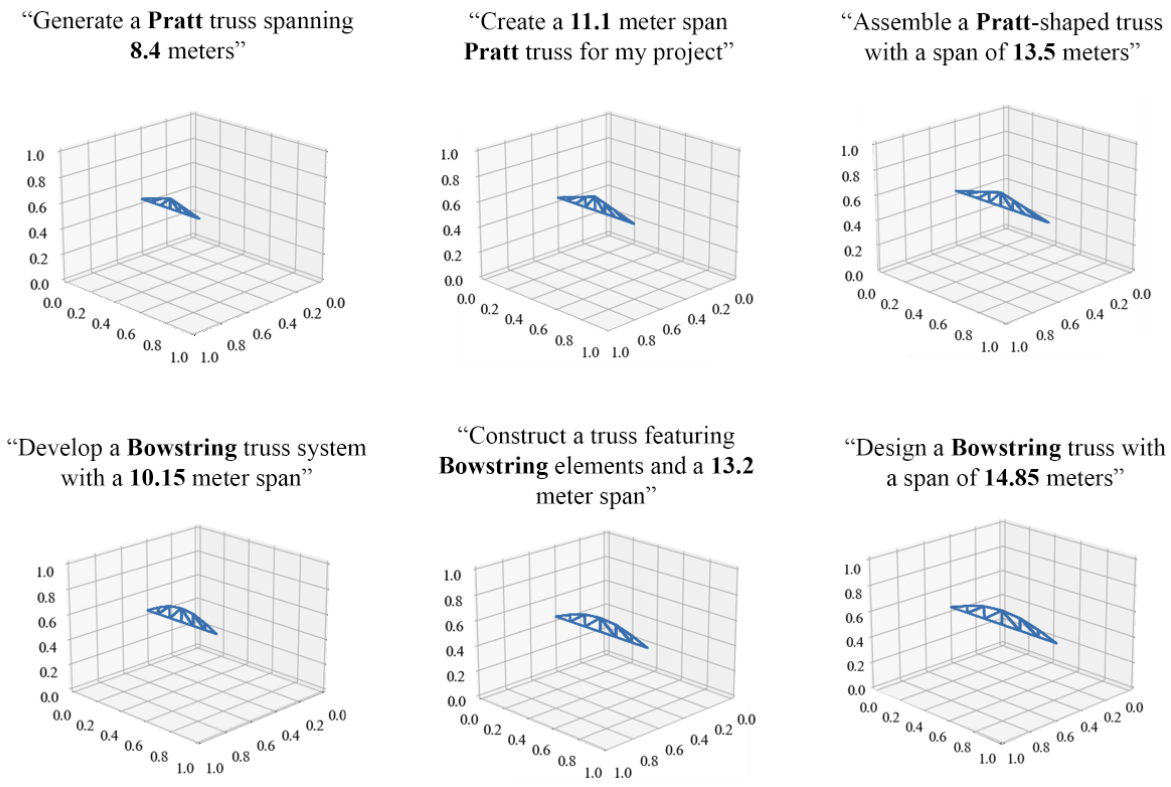


Figure 5.10 – Hypernetwork trusses for varied descriptions of length and structural form

5.2.4.4 Gridshell Design

To further examine the modelling capabilities of the proposed text-to-structure hypernetwork method, an additional design task was developed that required the reconstruction of three-dimensional gridshell structures. A parametric gridshell framework was developed to generate a range of optimal gridshell designs for the corresponding training, validation, and testing datasets, with spans ranging between 5 to 15 meters. The number of spanning gridshell members was tuned to minimise the overall weight of the structure while satisfying a maximum deflection limitation of L/360 and an imposed distributed load of 4.8 kN/m. The individual structural elements used to construct the gridshell form consisted of a steel circular cross-section with a diameter of 140 mm and a minimum yield strength of 235 MPa. The form of each gridshell was developed by scaling the structural span to define the edge catenary length. A hanging chain equation, modelled using the hyperbolic cosine function, was then solved. The horizontal distance between boundary nodes was taken, before the free-form sag of the curve under its weight was fit to satisfy the defined catenary length. The Karamba3D finite element component performed first-order structural calculations considering compressive and tensile forces in addition to the resulting displacements that each element was subject to. The supporting boundary nodes have fixed translational movement in the horizontal plane, effectively anchoring the gridshell, while rotational movement across all axes are enabled to allow the structure to adapt to loads and maintain overall stability. Each optimal gridshell was assigned a corresponding text-based command, containing the span of the underlying ground truth geometry. These commands were used as inputs to the proposed text-to-structure model pipeline, which outputs a predicted signed distance value for each sampled point within the 3D unit cube. Due to the complexity of the ground truth structures relative to previous experiments, the hypernetwork implemented for this task was trained over 250,000 epochs. Example predicted hypernetwork gridshells constructed after training are visualised in Figures 5.11 and 5.12 by extracting coordinates that yielded negative SDF values and meshing via the Marching Cubes algorithm. Cumulatively, these experiments illustrate the representation capabilities of hypernetworks within the context of text-to-structure design across a range of topologies and structural forms.

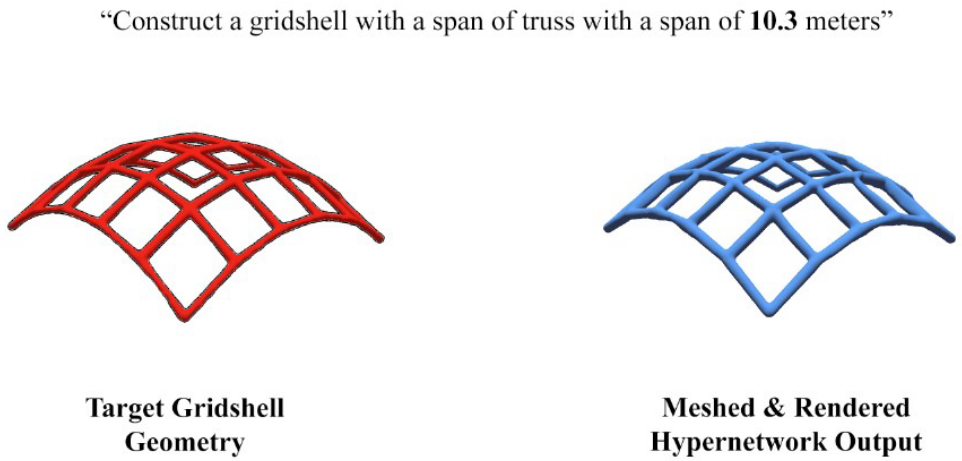


Figure 5.11 – Example comparison of ground truth gridshell and meshed hypernetwork reconstruction

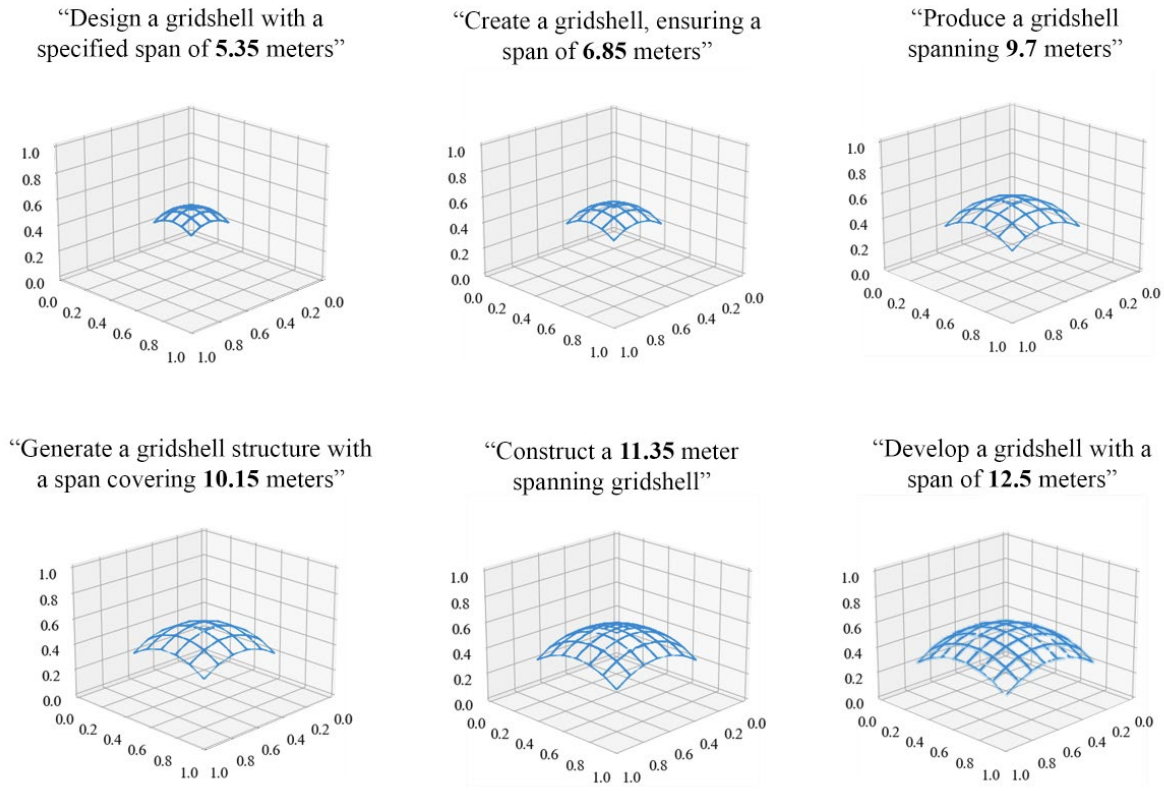


Figure 5.12 – Hypernetwork gridshells for varied span descriptions

5.2.4.5 Ensemble Bridge Design

Previous applications of hypernetworks for shape reconstruction have primarily focused on outputting a single class representation, commonly taken as the signed distance function of the ground truth object. This is a limiting factor when applied within the context of modern design tasks, which often require the modelling of complex, interconnected elements across primary and secondary structural systems. For these complex design tasks, it is necessary to model distinct structural components that can be individually isolated and modified according to the needs of the project. To account for these unique requirements, this research introduces an ensemble hypernetwork structure in which each individual hypernetwork is trained to reconstruct a specific structural form. Within this implementation, all hypernetworks inherited the same underlying architecture as previous three-dimensional examples, allowing the same primary network to be used to predict the signed distance function of each structural component. This ensemble architecture was applied to the task of generating concrete girder bridges according to the width and span dimensions specified within the input prompt. The bounds of these user defined structural parameters are provided for each design task in Table 5.2. Seven hypernetworks were individually trained within the ensemble over 3,500 epochs, targeted at reconstructing the concrete deck, parapets, abutments, girders, crossbeams, columns and the foundation of the ground truth structure. Each of the mesh-based training models used in this task was developed within a parametric Grasshopper environment. Final hypernetwork designs for given user commands are included in Figures 5.13 and 5.14.

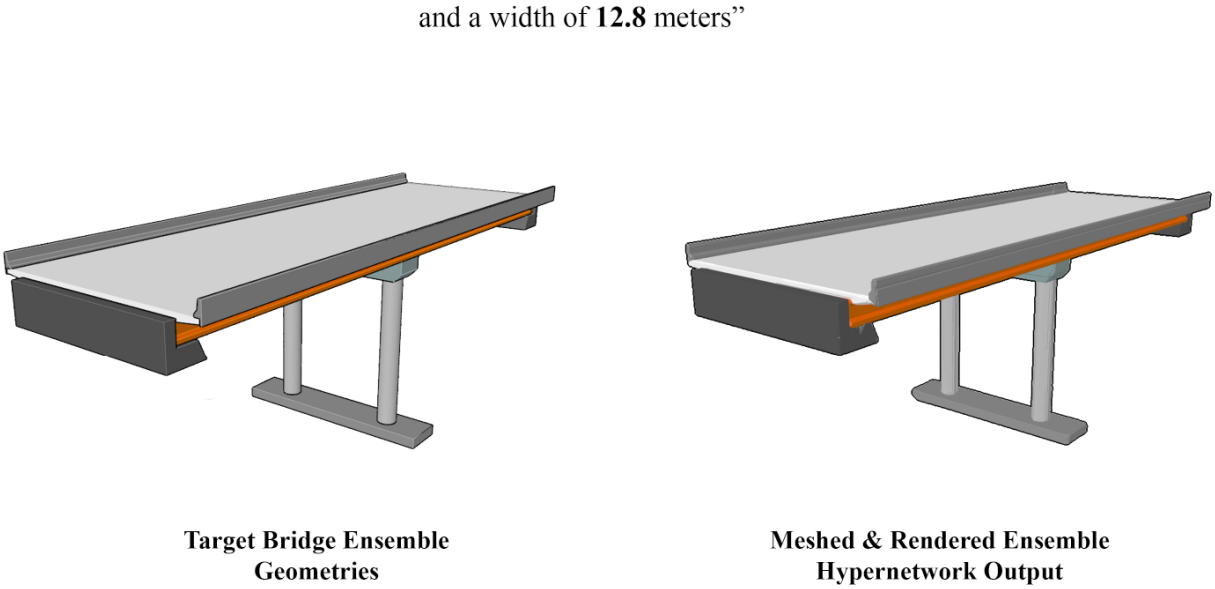


Figure 5.13 – Example comparison of ground truth bridge and meshed hypernetwork reconstruction

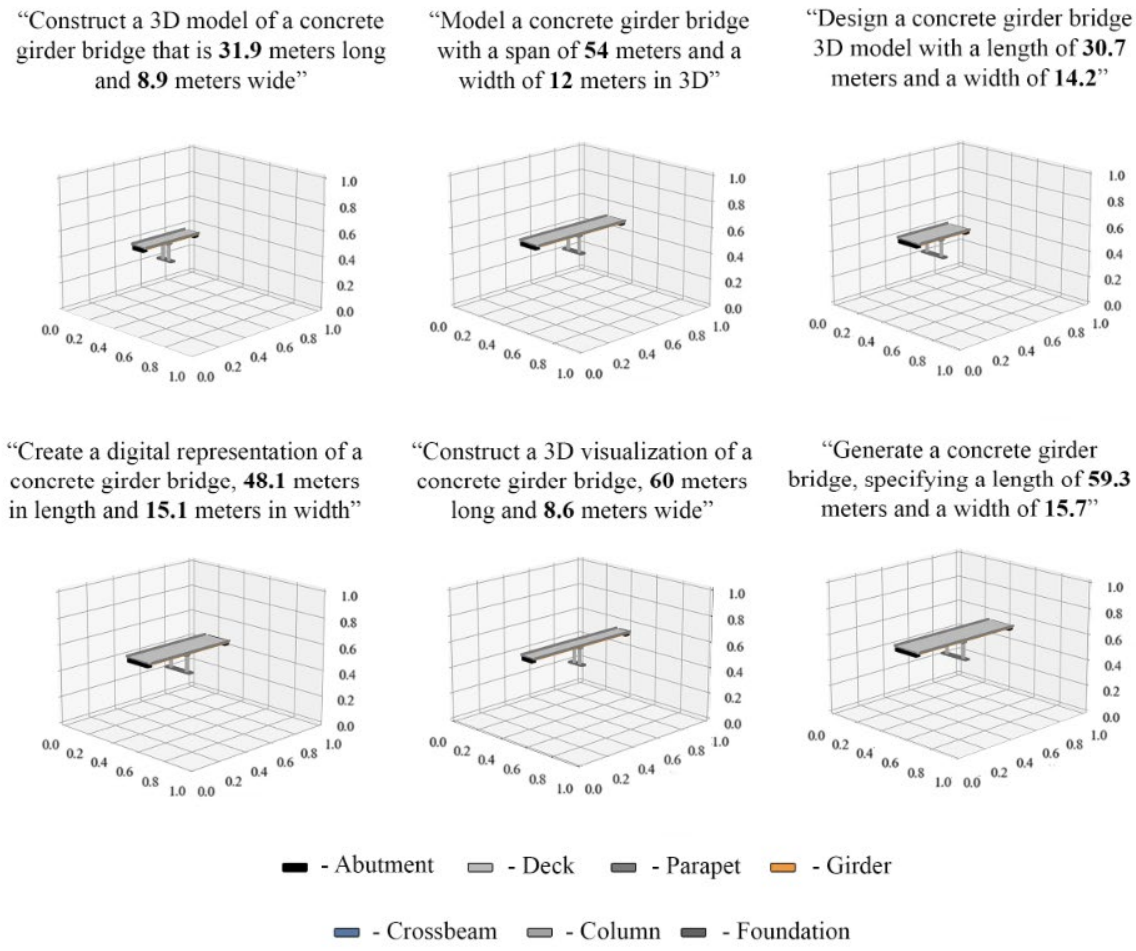


Figure 5.14 – Hypernetwork structures for varied descriptions of girder bridges

5.2.5. Results

To quantify the performance of the implemented hypernetworks, the generated structures have been evaluated against the ground truth models across each design task. According to the IOU scores of generated structures summarised in Table 5.3, hypernetworks appear to achieve the best performance within an ensemble arrangement. Under this training arrangement, complex structures were divided into their fundamental elements and distributed across a collection of hypernetworks. This method of breaking down models into their structural components effectively simplifies the topological forms that each network is required to learn. Hypernetworks trained to generate structural components for the ensemble bridge design task achieved an average IOU score of 0.8358 on the testing dataset, while the gridshell and three-dimensional truss hypernetworks achieved an average IOU of 0.6440 and 0.7088. Across all benchmarked structures, the best average testing IOU scores were recorded by the foundation, crossbeam, and column hypernetworks, recording values of 0.9493, 0.9288 and 0.9277, respectively. These results indicate that slender truss designs and their complex signed distance functions may be more challenging to learn when compared to bulkier structural forms.

Table 5.3 – Performance of the hypernetwork method for each design task

Design Task	Function	IOU_{Train}				IOU_{Val}				IOU_{Test}			
		Max	Mean	Min	STD	Max	Mean	Min	STD	Max	Mean	Min	STD
Truss – DE	Binary - 2D	0.8662	0.8112	0.6950	0.0297	0.8543	0.6696	0.3690	0.1516	0.8327	0.5639	0.2310	0.1978
Truss – SIMP	Binary - 2D	0.9032	0.8024	0.5282	0.0773	0.9028	0.7963	0.5281	0.0844	0.9032	0.8008	0.5253	0.0816
Truss – SSB	Binary - 2D	0.9921	0.8920	0.6709	0.0739	0.9818	0.8822	0.6980	0.0765	0.9910	0.8844	0.6875	0.0764
Truss – Pratt	SDF - 3D	0.7169	0.6869	0.2551	0.0512	0.7137	0.6915	0.6611	0.0135	0.7164	0.6987	0.6599	0.0126
Truss – Bowstring	SDF - 3D	0.7327	0.7192	0.4796	0.0196	0.7326	0.7225	0.7125	0.0064	0.7289	0.7211	0.7118	0.0049
Truss – Combined	SDF - 3D	0.7327	0.7034	0.2551	0.0417	0.7326	0.7059	0.6611	0.0189	0.7289	0.7088	0.6599	0.0149
Gridshell	SDF - 3D	0.8776	0.6662	0.3942	0.1523	0.8705	0.6453	0.0816	0.1878	0.8703	0.6440	0.0860	0.2078
Bridge Column	SDF - 3D	0.9439	0.9277	0.8921	0.0079	0.9421	0.9275	0.9019	0.0079	0.9431	0.9277	0.8955	0.0079
Bridge Abutment	SDF - 3D	0.8826	0.8802	0.8776	0.0012	0.8825	0.8802	0.8776	0.0013	0.8825	0.8802	0.8776	0.0012
Bridge Parapet	SDF - 3D	0.6549	0.5385	0.4586	0.0587	0.6516	0.5390	0.4580	0.0586	0.6550	0.5402	0.4613	0.0581
Bridge Crossbeam	SDF - 3D	0.9570	0.9298	0.9028	0.0146	0.9565	0.9286	0.9031	0.0144	0.9566	0.9288	0.9031	0.0147
Bridge Girders	SDF - 3D	0.8738	0.7168	0.4889	0.0779	0.8706	0.7272	0.4954	0.0736	0.8714	0.7225	0.4909	0.0776
Bridge Foundation	SDF - 3D	0.9603	0.9497	0.9226	0.0067	0.9601	0.9493	0.9341	0.0068	0.9602	0.9493	0.9326	0.0069
Bridge Deck	SDF - 3D	0.9438	0.9011	0.8344	0.0233	0.9427	0.9026	0.8369	0.0227	0.9429	0.9020	0.8357	0.0228

When comparing the optimisation strategies used to develop the structural datasets for each of the two-dimensional cantilever design tasks, a similar result is observed. The hypernetwork trained on structures that were generated using topology optimisation performed 42% better with respect to mean testing IOU and 8.47% better in terms of maximum IOU when compared against the hypernetwork that was trained on the Differential

Evolution database. This is due in part to the consistency and simplicity in the topological forms generated through the SIMP method, which experienced less stochastic variance between optimisation runs when compared against Differential Evolution. This improved the capacity of the network to generalise and predict weight tensors that best suited the design criteria extracted from the user input. Furthermore, the training trusses produced through the SIMP method exhibit a degree of topological freedom, while those generated through the Differential Evolution algorithm had an emphasis on constructability and uniformity of member sizing that induced more variance in the training dataset. Training trusses produced by the SIMP method also consist of elements that occupy a larger area within the design space when compared against the standardised members that were optimised through Differential Evolution, reducing the effect of incorrect predictions proximate to the prediction boundary.

The fine-tuning of the pretrained BERT models was executed prior to the training of the hypernetwork and is summarised in Table 5.4. These models have been evaluated separately from the geometric output generated by the text-to-structure workflow to allow for a holistic analysis of the proposed model pipeline. The transformer models achieved a near-perfect accuracy applied to the Named Entity Recognition task for each of the tested design tasks, successfully extracting the necessary design criteria from the user commands present within the training, validation, and testing datasets. The separation of the proposed workflow into two distinct training tasks also provides the opportunity to substitute each module with alternative networks without requiring a complete end-to-end model to be retrained.

Table 5.4 – Performance of transformer-based design parameter extraction

Design Task	**Extracted parameter**	**Training**		**Validation**		**Testing**	
		Loss	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**
Truss – DE	Load location	0.0007	0.9996	0.0002	1.0	0.0009	0.9686
Truss – SIMP	Truss depth & span	0.0004	0.9860	0.0004	0.9857	0.0003	0.9810
Truss – SSB	Boundary conditions, depth & span	0.0009	0.9971	0.0007	1.0	0.0014	1.0
Pratt & Bowstring	Span & truss type	0.0002	1.0	0.0067	0.9634	0.0022	0.9875
Gridshell	Gridshell span	0.0003	1.0	0.0003	1.0	0.0003	1.0
Bridge	Width and span	0.0001	0.9948	0.0001	0.9962	0.0001	0.9954

5.2.6. Discussion

This research posits that hypernetworks are capable of learning highly specific structural representations, generating clearly defined topological forms through an interactive ‘text-to-structure’ design workflow. Due to the representative capabilities of implicit neural fields, the hypernetwork methodology can be applied to a diverse range of structural design tasks with compelling performance. Notably, training dataset consistency is a critical challenge to consider when training hypernetworks for structural reconstruction, achieved within this research through the implementation of various conventional optimisation strategies and parametric

Grasshopper scripts. The results recorded across each of the six experiments indicate that datasets that had a consistent structural topology were reconstructed with greater accuracy when compared against datasets with a larger degree of structural variation. Ground truth training models with a uniform structural form yielded significantly greater IOU reconstruction performance, suggesting that consistent and algorithmic frameworks for dataset generation are uniquely viable when training hypernetworks. By creating a uniform structural representation within each training dataset according to the chosen design task, each hypernetwork model was able to learn and approximate the target geometries from a short text-based description of the design requirements, despite using a diverse range of optimisation methodologies to generate the training datasets.

The trained hypernetworks perform exceptionally across validation and testing datasets, which were not available to the models during training. Importantly, however, the validation and testing datasets are not radically different in their structural presentation. This is by design, as SIMP and DE dataset generation methods were selected for reproducibility to allow for consistent and reliable convergence during training. As the weights that are being predicted by the hypernetwork are for a base feed-forward model, the structures predicted by the proposed hypernetwork method are directly limited by the underlying distribution of the training dataset. Fully connected networks generalise well within the specific domain of the provided training data. The proposed hypernetworks are capable of recognising patterns in the underlying target dataset and can apply these patterns to predict structures within this domain. In this sense, the trained hypernetworks are capable of generalising within the learned design space. However, these designs should be interpreted as an approximation of the underlying optimisation strategy used to generate the training dataset such as DE or SIMP, rather than entirely novel generative solutions.

5.2.7. Future research and design applications

As the structures generated by the primary network are continuous manifold surfaces with a theoretically unlimited resolution and variation in topological form, the proposed hypernetwork method is applicable across a range of additive manufacturing and design tasks. The Marching Cubes algorithm has been used to process coordinates sampled from the predicted implicit neural, outputting a mesh-based geometry that may be used for 3D printing, computer-aided design, modelling, and design rendering. An example of this final output geometry is provided in Figure 5.15, demonstrating the capabilities of transformer-based hypernetworks to map text design criteria to viable manifold structures that can be easily introduced into existing modelling, rendering, and 3D printing design processes. While the input and output of the proposed method are bound within the unit cube to ensure hypernetwork training stability, the output manifold geometry can be scaled based on either the extracted user-defined design parameters or the initial normalisation parameter used to constrain the domain of the training dataset. Additionally, as structures are generated by querying the primary network, the resource footprint of the proposed method is only defined by the chosen size of the transformer module and hypernetwork architecture. In this sense, a trained hypernetwork can be significantly more efficient in terms of both storage and memory requirements when compared against the accumulative model database that it is trained from. A

single hypernetwork model has the capacity to learn from thousands of different structural models, while requiring only a fraction of the memory footprint. The trained hypernetwork architecture implemented across each of the three-dimensional design tasks was only 28.7 KB in size. This facilitates the use of the hypernetwork method as a light-weight tool that may be integrated within cloud-based platforms and enterprise design software.

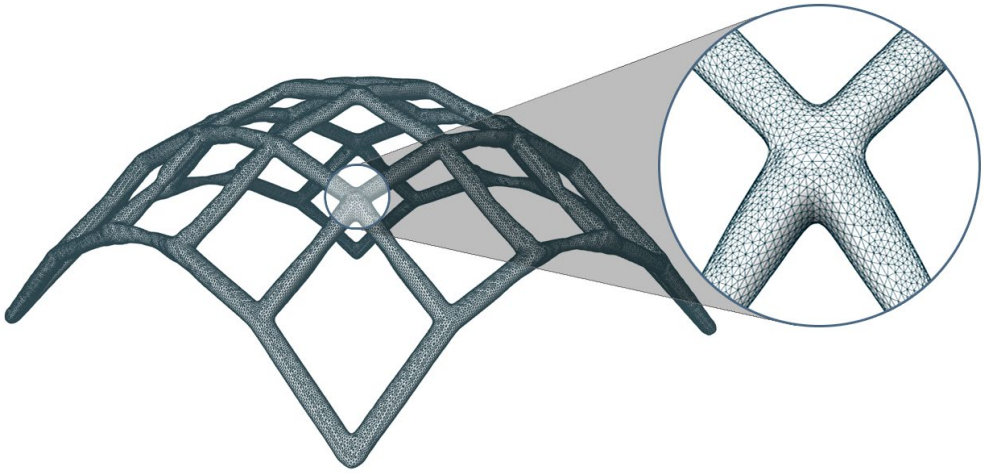


Figure 5.15 – Hypernetwork mesh reconstruction using the Marching Cubes algorithm

Importantly, the proposed model pipeline that has been introduced within this chapter is framed within the context of preliminary design iteration and benchmarking. Typically, algorithmic benchmark tasks incorporate statically determinate structures under linear-elastic conditions, observing isolated response factors such as deflection and stress using lightweight analysis strategies such as the direct stiffness method [248-250]. As such, the effects of buckling have not been included in the benchmark experiments that were developed within the scope of this research. Future research may seek to develop structural model datasets that encompass the broad array of structural standards that would be necessary to extend the proposed method within an applied design context beyond preliminary iterative phases. The information provided to the proposed hypernetwork model pipeline includes the normalised user design criteria such as structural depth and span, node boundary conditions, loading positions, and topological form. This collection of design parameters was chosen based on their relevance to the preliminary phases of the iterative design process. In principle, the output geometry aims to provide a geometric solution to these specific user requirements. Future research may seek to investigate targeted parameter sets for highly specific applications beyond preliminary iterative design and experimentation, such as the inclusion of material properties and internal forces as additional input parameters.

Furthermore, the proposed hypernetwork method is not designed to provide detailed specifications for each individual element. For example, ground truth structures generated using the Solid Isotropic Material with Penalisation (SIMP) method do not provide conventional design specifications such as the controlled definition

of individual element sizing. While hypernetworks offer a unique method to bridge optimisation disciplines, the resulting lack of element-level specification can be seen as a limitation within the context of detailed design. As a result, the application of these models leans towards conceptual and iterative design, which is particularly suited for preliminary project phases. Future research may seek to adapt the proposed hypernetwork pipeline by developing models that generate detailed design schematics that extend beyond the implicit neural field representations discussed within this section.

Finally, the structures reconstructed by the proposed hypernetwork method are directly limited by the underlying distribution of the training dataset and the selected optimisation strategy that was used to develop the dataset itself. Alternative models such as Generative Adversarial Networks and Variational Autoencoders may be substituted into the proposed model pipeline to provide more explorative variation that may be desirable in an applied design context. GANs are explicitly generative models, which use randomly sampled points from a Gaussian distribution as inputs to a generator to predict novel outputs. This stochastic variability is however often undesirable for reconstruction tasks that require close correspondence between the input data and the output geometry. Additionally, feed-forward networks are a supervised approach that leverage labelled datasets for highly accurate geometry reconstruction, while GANs are advantageous for unsupervised or semi-supervised learning tasks. The hypernetworks presented within this research have been developed to specifically approximate the underlying optimisation strategy within each respective design task under a supervised training methodology.

5.2.8. Research limitations

Despite the successful implementation of the proposed ‘text-to-structure’ method across each of the selected design tasks, there are several limiting factors that must be considered when training hypernetworks for structural reconstruction. First, the trained neural networks require high-quality structural models to achieve compelling results. The curation of these training datasets must also be carefully conducted, ensuring consistency in the ground truth structures. Additionally, the relationship between the primary and secondary networks is a leading contributor towards the representation capacity of the entire model, which can make network design a challenging task without further experimentation. Furthermore, the Named Entity Recognition task executed by the transformer region of the overall text-to-structure model assumes that the user has provided all of the necessary design criteria within their text input and that the specified units are identical to those used to develop the ground truth datasets. Future iterations on the ‘text-to-structure’ model pipeline may seek to extend the functionality of the transformer region beyond the scope of this research task to account for variations in the user input beyond the structured responses used within this research.

Notably, despite its widespread popularity, IOU suffers from several limitations that impact its application for certain highly specific design applications. Namely, IOU is insensitive to fine-grained variations in model output such as surface texture or smoothness, instead prioritising macro model accuracy. This limitation of IOU is a contributing factor for the popularity of alternative performance metrics such as Chamfer Distance (CD) or Hausdorff Distance (HD) within fields such as medical imaging [251] and surgery planning [252]. For design applications which require finer geometric detail such as additive manufacturing, these alternative performance metrics may be preferable. Inversely however, both Chamfer and Hausdorff distances suffer from several well-established limitations that make them ill-suited for the macro benchmark structures outlined in this research. Traditional forms of CD suffer from critical sensitivity to outliers, while being simultaneously insensitive to the varying density distributions that are often required to efficiently capture larger structural designs [253]. The overemphasis of point-to-point and edge-to-edge surface level details also hinders the capacity of CD and HD metrics for evaluating the global accuracy of large output geometries [254], with research demonstrating that CD-based optimisation can lead models to ignore critical elements and details [255]. The lack of inherent normalisation across HD and CD metrics furthermore limits their application within the benchmarking tasks outlined within this research, as both metrics are directly sensitive to the scale of the underlying ground truth data which varies across design tasks and individual structural components. The macro structures introduced within this section required a standardised metric to evaluate the global output structure, provided by IOU as a holistic benchmarking metric commonly used for generalised model datasets [256].

5.2.9. Summary of key findings from Section 5.2

This research introduced a novel transformer-based hypernetwork pipeline, tasked with mapping design criteria to manifold design geometry in a process termed ‘text-to-structure’ design. The collective output from this method is an implicit neural representation that can adapt to any representation provided within the training data, allowing the method to be applied to a diverse range of tasks relating to structural shape, size, schematic design, and topological form. This interactive framework, coupled with the representation capabilities of implicit neural fields, allows for the potential application of hypernetworks throughout the structural engineering and architectural disciplines. The analysis of structures constructed through the proposed model pipeline demonstrates that transformer-hypernetworks are capable of mapping text-based design criteria into a tensor of weights that, when transferred to a defined primary network, construct viable design solutions for both three-dimensional and two-dimensional structures.

5.3. TRANSFORMER HYPERNETWORKS FOR COMPLEX STRUCTURAL REPRESENTATION

T. Hielscher[1*] , S.A. Hadigheh[1]

[1]School of Civil Engineering, Faculty of Engineering, The University of Sydney, Sydney, New South Wales 2006, Australia

*Corresponding Author: Tommy Hielscher

Journal of Computing in Civil Engineering (ASCE), https://doi.org/10.1061/JCCEE5.CPENG-6388

5.3.1. Abstract

This paper introduces a novel method for generating large-scale structural systems from text-based user prompts. The proposed reconstruction approach consists of a fine-tuned named entity recognition model and an ensemble of deep neural hypernetworks, which are respectively tasked with extracting the design criteria from the user's input and predicting each of the base structural elements for the given design task. These geometries are replicated and manipulated within the design space using a series of translation vectors that are generated by an additional transformer encoder, allowing for the construction of complex, interconnected structural systems with unprecedented representation capacity. These models achieved an average criteria extraction test accuracy of 96.89% and a weighted average intersection over union of 0.7868 across the test reconstruction datasets. The complexity of designs produced by the proposed 'text-to-structure' models significantly exceeds the scope of previous research within the field of 3D shape reconstruction, advancing the hypernetwork method and similar reconstruction techniques to meet modern engineering and architectural applications.

Supplementary material is available at: https://github.com/thie7042/Scaling-Hypernetworks

5.3.2. Introduction

Hypernetworks were introduced in 2016 [109], providing a unique learning methodology that overcomes significant inference limitations impacting many modern deep learning models. Traditional machine learning models utilise fixed parameter sets that are learnt directly through backpropagation, using set architectures that cannot be modified without restarting the training process. In contrast, hypernetworks can parametrise a diverse range of models by predicting weights that are conditioned by targeted inputs. Within the context of hypernetwork research, there are two central models that are conventionally applied: the primary network and the hypernetwork. These interdependent models are responsible for distinct tasks within the overall training process. The primary network conventionally performs the required functionality of the given deep learning task, such as classification, 3D reconstruction, or regression. The hypernetwork is trained to predict the weights that the primary network is initialised with, providing an abstracted learning process with dynamic, context-aware model parameters. The given task is directly learnt through this relationship between the primary network and the hypernetwork, with gradient propagation through the hypernetwork's parameters during training and loss minimisation measured against the primary network outputs. In this sense, the hypernetwork is tasked with

generalising across a range of inputs, predicting network weights to maximise performance for a given task. This training methodology offers a range of compelling advantages over traditional methods, including soft weight sharing, parameter efficiency, adaptive architectures, and input abstraction [257].

Hypernetworks have been applied across a range of deep learning architectures since their inception, including multi-layer perceptrons (MLP) [258], recurrent neural networks (RNN) [259], and convolutional neural networks (CNN) [260]. Hypernetworks have also demonstrated state-of-the-art performance across a range of challenging deep learning tasks, including adversarial defence [261], neural architecture search [110], hyperparameter optimisation [262], pose estimation [263], neural style transfer [264], continual reinforcement learning [265], audio reconstruction [266] and shape representation [119]. Recent advances within the field of 3D geometry reconstruction have illustrated the promising versatility of the hypernetwork method, however cutting-edge applications have been predominantly applied within the context of individual object datasets such as the models provided in the open-source ShapeNet [267] and PartNet [268] sets. While this precedent is incredibly valuable for benchmarking the performance of foundational models and novel variations, there exists a significant gap between academic research and applied industry designs in terms of task complexity [269]. To the authors' knowledge, no solutions have been proposed within the available literature to address this vast gap. This research paper aims to reconcile the disconnect between promising hypernetwork results published across academic benchmarks and the complex structural systems that exist in practical design applications.

Within the field of 3D reconstruction, a broad variety of innovative techniques have been introduced with compelling results. Leading solutions include implicit neural representations from hypernetworks [119], cubic voxel reconstruction [122], template mesh deformation [270], transformers [271], and diffusion models [272]. Each of these distinct methodologies inherits a range of challenges and limitations that must be considered prior to their integration with any given reconstruction task. Transformer-based reconstruction models utilise self-attention to capture long-range relationships within the data, mapping spatial dependencies and patterns across 3D objects and environments [273]. These architectures are also highly scalable and are capable of processing unstructured representations such as point clouds and meshes [274]. However, transformer-based 3D reconstruction models are computationally expensive compared to competing approaches, requiring significant memory resource allocation for inference which limits their application within the context of consumer-grade computer aided design [275]. Diffusion-based models have also recently gained traction as powerful and robust generative approaches, however their iterative noise refinement process and significant memory requirements have impacted their broader application for enterprise design software [276]. High resolution voxel-based models also incur significant computational costs and high memory requirements due to their sampling requirements, limiting their application across lightweight applications and tasks requiring efficient inference speeds. Researchers have attempted to overcome this challenge of voxel-based approaches by significantly

reducing the resolution of the cubic grid, thereby limiting the representation capacity of the model itself [277, 278].

Implicit neural fields provide a highly flexible approach, free from the confined structure of a cubic grid. As a result, these methods also depart from traditional 3D convolutional models, which require a positional regularity that is well-suited for voxel-based reconstruction. Implicit neural field representations have demonstrated compelling versatility within this context, allowing for a theoretically unlimited representation capacity and complete flexibility in how the design space is sampled to construct the predicted geometry [119, 279]. In contrast, the representation capacity of mesh deformation is a product of the mesh template's vertex density and its similarity to the underlying target geometry [280], limiting its application to reconstructing simple geometries such as furniture [281], household objects [282], and basic vehicle shells [283].

This research posits that the academic field of 2D and 3D reconstruction has been limited by the scope it has inherited from traditional approaches. Leading papers have focused on producing novel variations to foundational methods, using generalised geometric datasets that are not representative of applied design tasks [284-286]. There are some notable exceptions to this assertion within the context of architectural and engineering design, such as wire-frame modelling [287], generative adversarial floor plan design [288], and diffusion-based shear wall modelling [289]. However, these novel deep learning solutions predominantly focus on high-level output representations and preliminary massing—ignoring the modelling demands of complex systems with hundreds or even thousands of interconnected structural elements. As a result, there is a vast gap within the field that demands research into methods of scaling these foundational approaches for complex applications. Throughout the available literature, there are currently no proposed solutions that enable researchers to scale these foundational 3D reconstruction techniques beyond their well understood limitations to meet the needs of industry. This research is centered around the limitations of foundational approaches, providing a holistic method to build complex designs from relatively simple predicted geometries.

5.3.2.1 Research Significance

The integration of predictive machine learning tools into existing architectural and engineering design workflows has significant potential to transform decision making processes, optimise performance, and enhance efficiency. As a leading model architecture that has achieved competitive performance across 3D reconstruction, neural architecture search, meta-learning, and neural representation tasks, hypernetworks offer a unique opportunity to contribute towards this transformation. Despite widespread adoption within the broader field of deep learning and a theoretically unlimited capacity for representing underlying target functions [290], the application of hypernetworks within the field of structural and architectural design is also yet to be fully realised. This is largely due to the complexity of designs present within these fields and the challenges inherent to training

hypernetworks. Within the context of shape reconstruction, the complexity of geometries that hypernetworks are capable of reconstructing is significantly impacted by the size of the hypernetwork itself, the number of predicted output weights, and the architecture of the primary network [119]. This sensitivity has limited the field of 3D reconstruction to focus on the generation of individual objects, such as the models contained within the widely used ShapeNet research dataset [267, 269, 291]. Objects within these benchmark datasets are self-contained objects such as bookshelves, mugs, and furniture, rather than components of a larger, interconnected geometric environment.

The transformer encoders introduced within this research allow foundational models such as hypernetworks to scale beyond these limitations by predicting translation vectors that transform and replicate geometries throughout the design space, offering new opportunities to reconstruct designs with hundreds or even thousands of structural elements. These models aim to advance the application of leading reconstruction methods towards the generation of complex, interconnected structures that are similar in complexity to designs drafted in preliminary architectural and engineering tasks. Furthermore, the implemented ensemble hypernetwork models allow for a diverse range of different structural elements to be constructed as separate manifold objects, which may be rendered with different material properties and manipulated independently for preliminary design tasks.

The driving motivation for this research is to empower researchers and industry practitioners to tackle challenging reconstruction design tasks. This has been addressed through the development of novel 'text-to-structure' models—extracting design criteria from the user's input commands, predicting basic structural elements using ensemble hypernetwork models, and manipulating the generated structural geometry within the design space to scale to complex applications. By constructing structural elements from ensemble-based neural fields and manipulating them within the design space using translation vectors generated from transformer encoder models, the proposed models have been applied to construct a variety of 3D models with unprecedented representation capacity. Through these contributions, this research aims to empirically demonstrate a compelling strategy for researchers to scale the predictive performance of existing deep learning models to meet the complexities of large-scale designs, which have been largely ignored throughout the available literature.

5.3.3. Methods and Techniques

5.3.3.1 Structural representations

Implicit neural representations such as signed distance functions (SDF) offer a unique opportunity to capture complex topological forms in a coordinate-based manner, rather than directly relying on explicit geometric representations such as polygonal meshes, point clouds, or cubic grids. The implicit neural field predicted by the proposed hypernetwork models aims to reconstruct the SDF of an underlying target geometry, which can be sampled at any desired resolution. An example of this structural representation format is illustrated within Figure

5.16, depicting (a) a diagrammatic 2D point map of a local region of the design space that has been sampled, (b) the corresponding 2D cross-sectional SDF heatmap, and (c) a 3D render of the mesh geometry. Within the context of this research paper, the SDF value represents the shortest distance between a given point and the surface of the structural element being generated. The decision boundary represents the manifold surface of the structural mesh at $SDF = 0$. Points existing outside of the target geometry record a positive SDF value ($SDF > 0$), while points existing within the shape are assigned a negative SDF value ($SDF < 0$). To capture a holistic representation of the design space and the underlying target geometry, a combination of random sampling and boundary-proximate sampling was used to generate the target training, validation, and testing SDF datasets [215]. Points sampled close to the boundary surface of the target geometry provide useful information regarding the detailed shape of the element, while randomly sampled points allow for a generalised representation of the design space to be learnt during training [292]. To convert the predicted implicit neural field to a manifold geometry for visualisation and evaluation, the marching cubes algorithm was implemented. For this postprocessing task, a fine grid of coordinates may be used according to the desired resolution of the final design. Unlike traditional voxel-based models which are trained to a fixed grid resolution, implicit representations can also be sampled by adaptive grids that significantly improve reconstruction resolution and reduce computational cost [293].

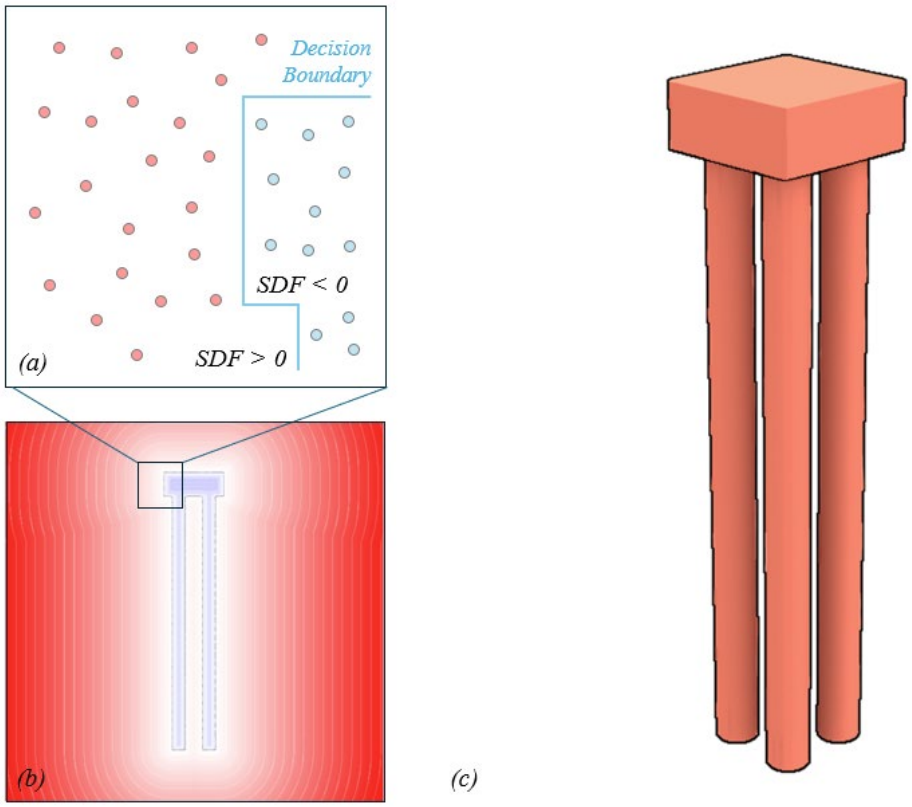


Figure 5.16 – Signed distance representation of an example structural foundation geometry

5.3.3.2 Translation Vectors

Previous applications of the hypernetwork method have been limited to simple objects that lack the complexity required for modern engineering and architectural design tasks. This paper introduces a novel method for manipulating implicit neural field representations within the design space, scaling the hypernetwork method to construct holistic and interconnected structural systems. The proposed translation vector method utilises a trained transformer model to map the extracted user design criteria into a single one-dimensional tensor. This output is then subdivided into a series of three-dimensional vectors {x,y,z}, which each represent a unique displacement applied to the base geometries generated by the ensemble hypernetworks. Basic class specific post-processing rules may also be introduced, such as symmetrical element manipulation if they meet the needs of the given design task. An example of this element displacement for a high-rise design task is outlined in Figure 5.17, demonstrating the positioning of hypernetwork geometric outputs according to transformer-based translation vectors, effectively scaling the combined representation capacity of the underlying hypernetwork models. The unique translation vector method introduced within this paper is partially inspired by existing sphere-based deformation techniques, which generate 3D shapes by progressively deforming a spherical point cloud template [271, 294]. By predicting vectors that displace individual points within the design space, sphere-based deformation models reconstruct simple geometries from point-clouds that are similar in complexity to previous hypernetwork approaches [295]. This research paper extends this concept of predicting displacement vectors to build complex designs from simple structural elements such as columns, beams, foundations, and floor slabs.

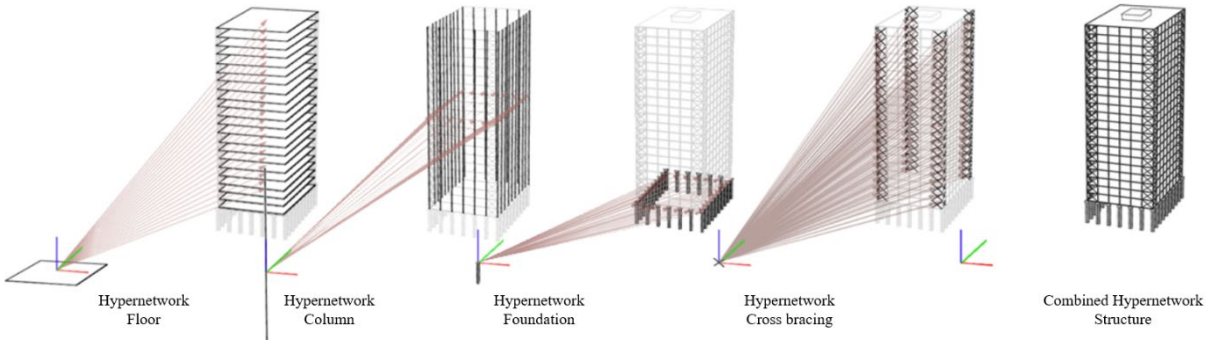


Figure 5.17 – Manipulation of hypernetwork meshes using transformer translation vectors

5.3.3.3 Model Architecture

The proposed 'text-to-structure' model consists of three distinct processes: (i) extraction of the necessary design criteria from the user's input command using a fine-tuned named entity recognition (NER) model, (ii) reconstruction of structural elements using ensemble hypernetwork models, and (iii) the prediction of transformer-based translation vectors to manipulate the generated structural geometry within the design space. This research paper employed bidirectional encoder representations from transformers (BERT) for the NER

model, as a cutting-edge pretrained transformer architecture that captures semantic relationships between the words within the user's input command [222]. Since its publication in 2019, BERT has demonstrated superior performance across a wide range of challenging natural language processing tasks, including text classification [296], information extraction [297], question answering [298], and text similarity prediction [299]. The fine-tuned NER model takes a tokenised representation of the user's input command as its input, before being passed through 12 BERT-based transfer layers to capture semantic relationships within the input sequence. Each of these layers contains multi-head self-attention modules, fully connected (FC) layers, normalisation functions, and residual skip connections [119]. The output embedding vectors are then passed to a softmax classification function, predicting a probability distribution for each token in the input sequence [300]. The design criteria are extracted from the input command according to these probability distributions, before being normalised and passed to the hypernetwork model. These criteria include the key design parameters that were necessary to build the parametric rules-based structural dataset for each design task, such as the structural span, structural width, or the number of stories within the overall design. The extracted design criteria are highlighted in bold for each of the example output designs visualised in Figures 5.23, 5.24, and 5.25.

For each of the design tasks evaluated within this research, the first four layers of the trained hypernetwork contain 1,024 neurons and a rectified linear unit (ReLU) activation function. This hypernetwork model architecture was chosen based on the precedence set within the available literature, which predominantly uses ReLU or leaky-ReLU variations for the hidden layers of the FC hypernetwork, typically with a total of 2 to 6 layers [257, 301-304]. The fifth layer of the proposed hypernetwork outputs a tensor of weights with a length of 3,329, mapping to each of the parameters within the primary network. These weights are processed by a final hyperbolic tangent activation function (Tanh) to zero and scale the output weights [305]. The five-layer primary network is initialised using these predicted output weights, before taking a series of 3D coordinates sampled from the design space as its input. The first four FC layers of the primary network contain 32 neurons and a ReLU activation function. The final layer of the primary network outputs a single value, representing the predicted signed distance value for each of the given input points. The predicted SDF values are evaluated against the ground truth SDF dataset using the mean absolute error loss function [306].

The transformer model responsible for generating the translation vectors introduced within this research takes the extracted design criteria as its input, which is then transformed into embeddings with a size of 512. Positional encodings are added to the input sequence, allowing the transformer model to learn the positional information of each design specification. The model itself consists of 6 encoder blocks, each containing multi-head attention modules with 8 heads and a FC feedforward network consisting of two layers of 2,048 neurons, a ReLU activation function, and a regularisation module with a dropout probability of 0.1. Multi-head attention has been implemented within the proposed model to allow each head to learn unique semantic and syntactic relationships within the input tokens. Each attention head learns different weight matrices, improving the efficiency of the

training process and enabling parallel processing for improved computational performance [307]. Two residual skip connections are included within this region: one skipping the multi-head attention module and one bypassing the FC layers. These residual connections reduce the effects of vanishing and exploding gradients by allowing gradients to propagate throughout the network, retaining the original input signal and improving stability during training [308]. The output tensor is subdivided into individual vectors, each defining the x,y,z displacements for the base geometry generated by the hypernetwork method. The hypernetwork manifold geometry is then manipulated within the design space according to these displacement translation vectors, allowing structural elements to be repeated and moved to assemble a final consolidated structural form.

The full model pipeline is summarised in Figure 5.18, including the BERT-based NER model that extracts the user's design criteria (D), the transformer encoder that generates the translation vectors, FC layers, and an example hypernetwork used to generate the base structural element for the given design task. The ensemble arrangement of transformer encoders and hypernetworks is illustrated in Figure 5.19, detailing how individual structural forms are predicted and manipulated within the design space to construct a holistic structural model. This figure further details the repetition of the hypernetwork and transformer encoder architecture for each structural class within the ensemble arrangement, allowing each of the fundamental base structural geometries to be predicted, replicated, and manipulated within the design space according to the criteria extracted from the user's input prompt.

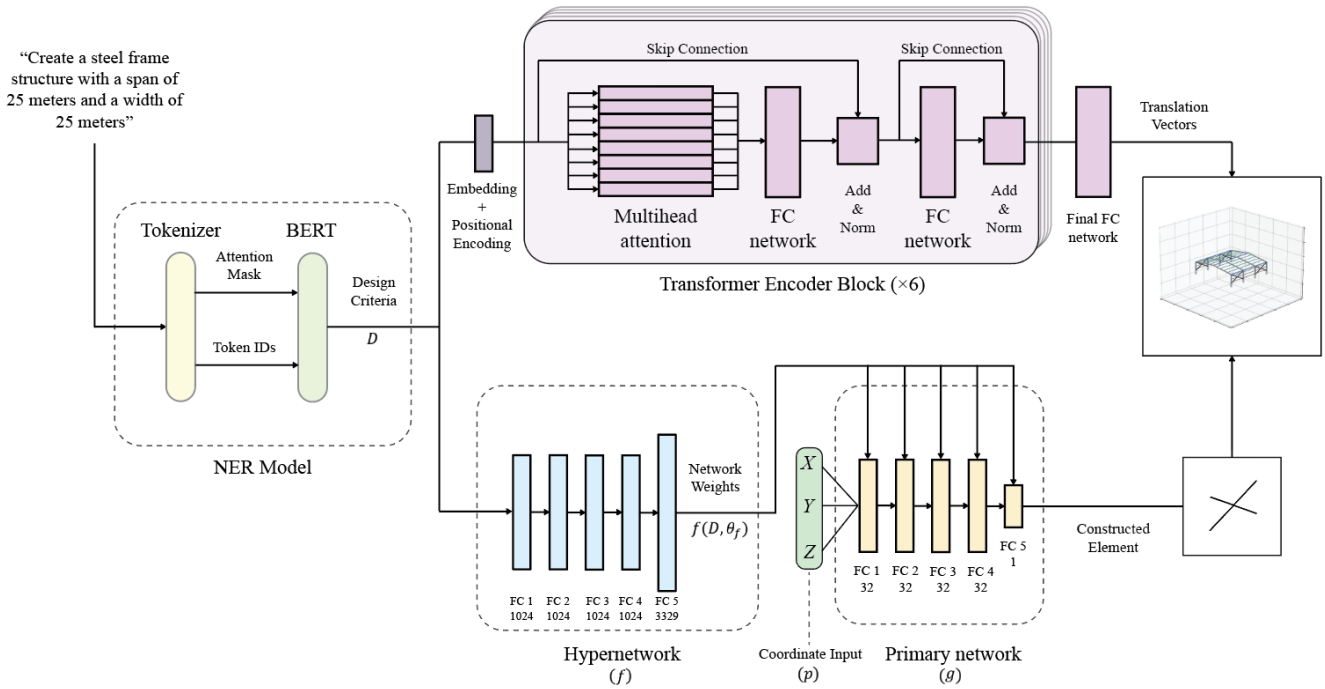


Figure 5.18 – Cross bracing text-to-structure example for steel frame structures

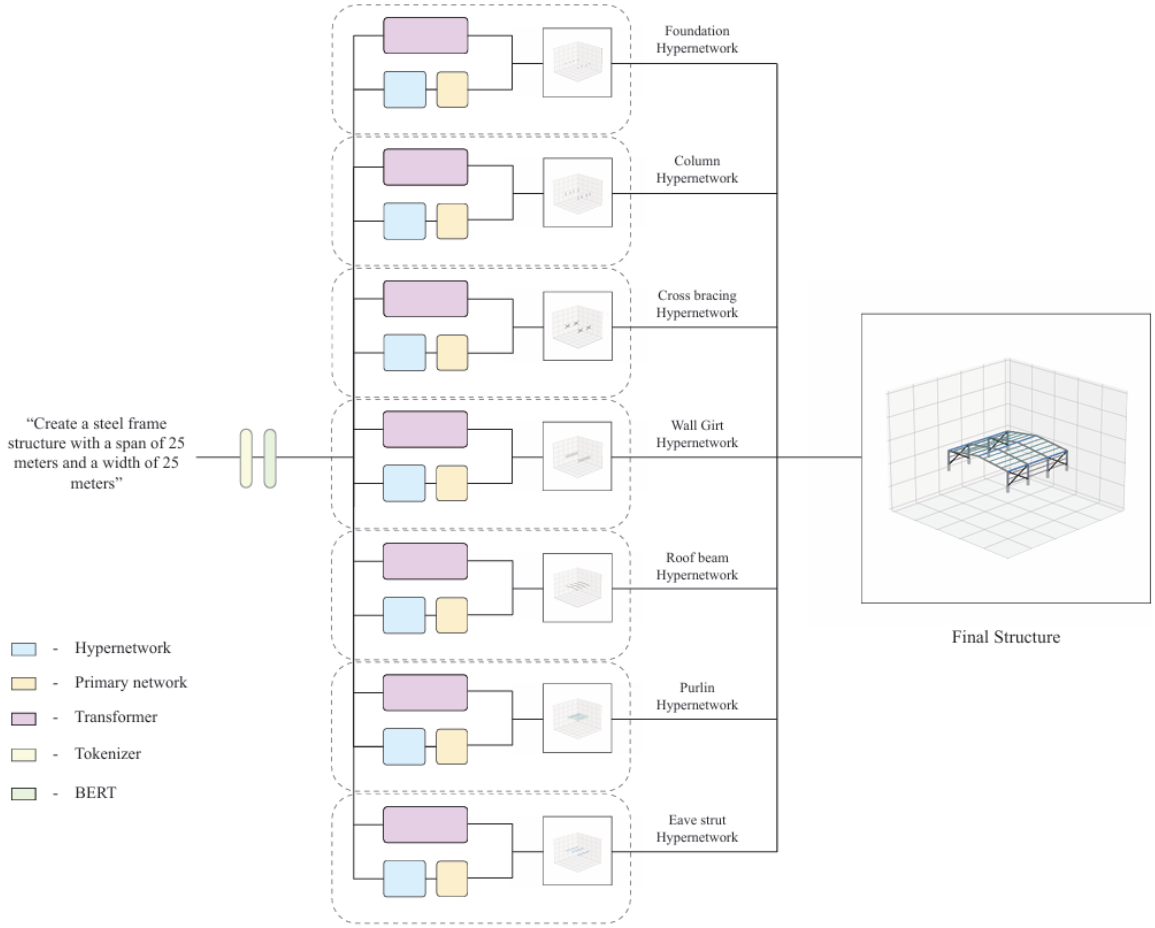


Figure 5.19 – Ensemble hypernetwork design for steel frame structures

5.3.3.4 Output Evaluation

Within the proposed hypernetwork models, a network f takes the normalised design parameters D that have been extracted from the user's input commands and predicts the weights θ_f of a primary network g. The primary network is then initialised using these weights, before taking a series of 3D coordinates as inputs and predicting the corresponding signed distance value for the target geometry at each point [309]. The mean absolute error (MAE) loss function was used to train the proposed hypernetworks based on the difference between the predicted signed distance values and the ground truth dataset, calculated as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - g\left(p, f\left(D, \theta_f\right)\right)\right| \tag{5.4}$$

where y_i is taken as the target signed distance value of each point p for sample i within a batch of size N. The MAE loss function provides an average of the absolute error between the predicted and target signed distance values. The proposed hypernetworks were trained using a learning rate of 2×10^{-5} and a batch size of 20. All proposed models that are discussed within the context of this research were trained or fine-tuned using the Adam optimiser. The generated manifold geometries have been evaluated according to the Intersection over Union

metric (IoU). The IOU value is calculated from the predicted implicit neural field of a given structural component generated by the hypernetwork method, measured against the target geometry as follows:

$$IoU = \frac{|V_{pred} \cap V_t|}{|V_{pred} \cup V_t|} \tag{5.5}$$

where V_{pred} is taken as the volume of the predicted structural element, V_t is the ground-truth target geometry, $\cap$ denotes the intersection between these volumes within the design space, and $\cup$ denotes the union of these volumes [310]. The average, standard deviation, maximum, and minimum IOU of each hypernetwork across the training, validation, and testing model datasets have been recorded in Table 5.5, documenting the performance of the proposed hypernetworks across a range of complex structural design tasks. The NER model used to extract the necessary design criteria from the user's input command was fine-tuned using the categorical cross-entropy loss function, calculating the difference between the ground truth distribution and the predicted output distribution. The target design parameters are each assigned a unique class label (y), which are used to evaluate the output predictions ($\hat{y}$) as follows:

$$CCE = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log \hat{y}_{i,c} \tag{5.6}$$

where the total number of classes used during fine-tuning is taken as C. The NER model was fine-tuned using a learning rate of 2×10^{-5} and a batch size of 20. The losses and accuracies of the fine-tuned NER model across the training, validation, and testing datasets are summarised in Table 5.8. Finally, the transformer encoder models were trained using a learning rate of 1×10^{-4} and a batch size of 64. The translation vectors generated by the transformer encoder models were evaluated according to the Mean Squared Error loss function (MSE), measuring the difference between the ground truth vectors and the model's predictions as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.7}$$

5.3.3.5 Mesh Rendering

The final implicit neural field generated for each structural component was transformed into a manifold polygonal mesh for visualisation and further evaluation using the marching cubes algorithm (Lorensen and Cline 1998). The design space containing the input 3D scalar field was first divided into a grid of cubes, with each cube evaluated to determine whether it contains a vertex. The intersection between the iso-surface and the cube provides a unique configuration, forming polygon triangles that belong to a precomputed set of 256 configurations [311]. The vertices of each cube are interpolated and connected, forming a collection of triangular polygons that are combined into a single continuous mesh [234]. This transformation of the hypernetwork output allows the proposed element construction process to be directly applied to traditional design modelling tasks [312], rendering [313], 3D printing [314], structural simulation [315], and architectural visualisation [316].

5.3.4. Experiments

Developing large scale and complex design datasets with hundreds or even thousands of interconnected structural meshes remains a challenging task for researchers aiming to bridge the gap between academia and industry. A limited availability of existing design datasets [317], industry restrictions on sharing sensitive project data [318], costs associated with labour-intensive manual model dataset curation [319], and feasibility constraints surrounding the collection of real-world high-resolution LIDAR datasets [320] have collectively hindered the practical application of industry-based research. Within this context, dataset creation methods that have received widespread application and demonstrated compelling performance typically utilise some form of rules-based programming, such as parametric computer aided design (CAD) scripts [321, 322], metaheuristic algorithms [2, 323-325], or generation through simulation [326-328].

Given the inherent challenge of dataset curation and the demonstrated application of rules-based methods, a parametric design generation process was incorporated for each of the design datasets created for this research. Models were parametrically generated for training, validation and testing datasets, including structural meshes, text-based descriptions of each structure, and signed distance values for sampled coordinates. To evaluate the performance of the proposed models for reconstructing and translating structural geometries within a set design space, three complex design tasks were selected for this research paper. These include the generation of high-rise, tied arch bridge, and steel frame structures that satisfy a broad range of design criteria. A total of twenty structural elements were selected as the base components that were manipulated within the design space to construct these designs. A total of 142,428 structural elements were produced across all three of these experiments, 80% of which were reserved for training. The remaining structures were equally divided into validation and testing datasets. Sample structural elements from this mesh dataset are provided in Figure 5.20, visually summarising the simple building block components. Notably, three of the twenty structural base geometries that have been produced for these design tasks were defined as fixed geometries, as they were not directly relevant to the selected design criteria that drove the parametric rules-based approach. These include the steel frame foundation, steel frame column, and tied-arch substructure column components. While these standardised geometries are less useful from an individual reconstruction evaluation perspective, they have been included as standardised elements that are necessary to compose each of the final designs and holistically evaluate the vector performance of the transformer models. Each of the final designs is described by a text-based description containing each of the design criteria that were used to generate the structures themselves. These parameters are extracted at inference through a trained NER transformer model, before being passed to the translation transformers and hypernetwork models, generating the translation vectors and manifold structural geometries, respectively.

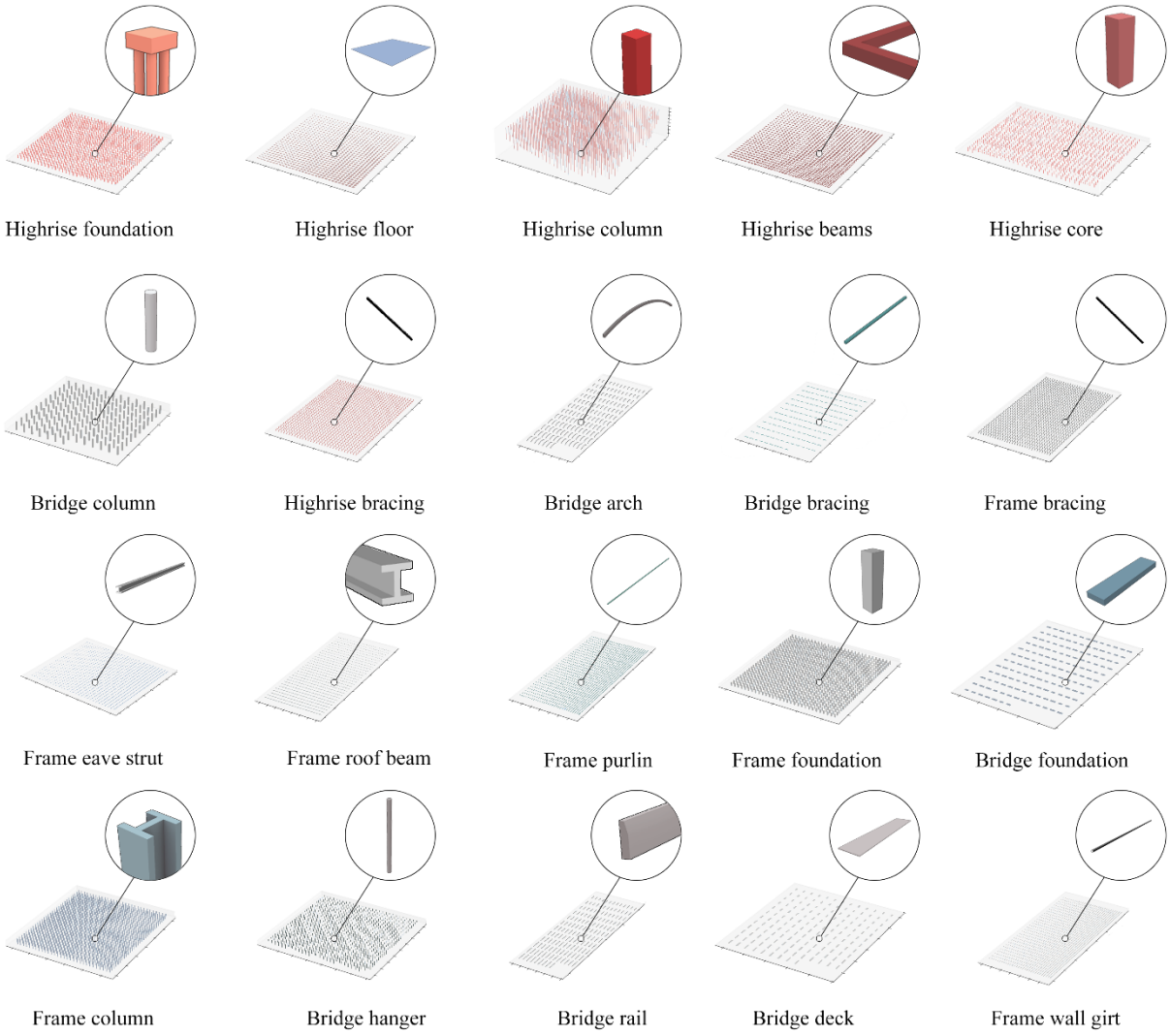


Figure 5.20 – Example samples from the base structural element dataset

5.3.4.1 Highrise Design

The Highrise design task was selected due to the complexity and range of its interdependent substructures, including foundations, columns, beams, cross bracing, floor systems and a central core. These individual geometries are generated through an ensemble of hypernetworks, before being translated within the design space using vectors generated by trained transformer models. The user commands used to train these models include the necessary plan dimensions and the number of stories required in the final structure. The training dataset that was parametrically generated for this research included structures with plan widths ranging from 10 to 40 meters in each axis and up to 30 stories in height, with a combined total of 22,500 high-rise models, each containing six distinct structural topologies.

5.3.4.2 Tied Arch Bridge Design

The chosen tied arch bridge preliminary design task required the generation of 7 different structural components, including columns, decks, arches, hangers, bracings, rails, and foundation elements. These elements constitute a diverse range of challenging geometric topologies that were learnt by the trained ensemble hypernetworks. Generated structural models were broken down into bay sections which vary according to bay span and width, conditioned by the user's input commands. These sections range in span from 30 to 60 meters, with a width ranging from 8 to 12 meters. The number of bays repeated across the total span of the bridge is also conditioned by the user's input, ranging from 1 to 6 structural bays.

5.3.4.3 Steel Frame Design

The final design task that was evaluated for this research included the generation of steel frame structural models from user commands. These structures include 7 different structural forms, including foundations, columns, cross bracing, purlins, wall girts, eave struts, and roof beams. The frames themselves vary by frame width and span, with each model assigned a corresponding text-based description of the structure containing these two necessary design parameters. The structural span and width explored within this research dataset each range between 10 and 40 meters. An example comparison between the ground truth high-rise design and the reconstructed hypernetwork for each design task is provided in Figure 5.21 and Figure 5.22, illustrating the representation capacity of the proposed models and the scalability of the novel translation vectors introduced within this research. Additional examples of design variations conditioned by text-based commands are provided in Figure 5.23, Figure 5.24, and Figure 5.25, demonstrating the diversity of structural systems that the proposed method is capable of reconstructing.

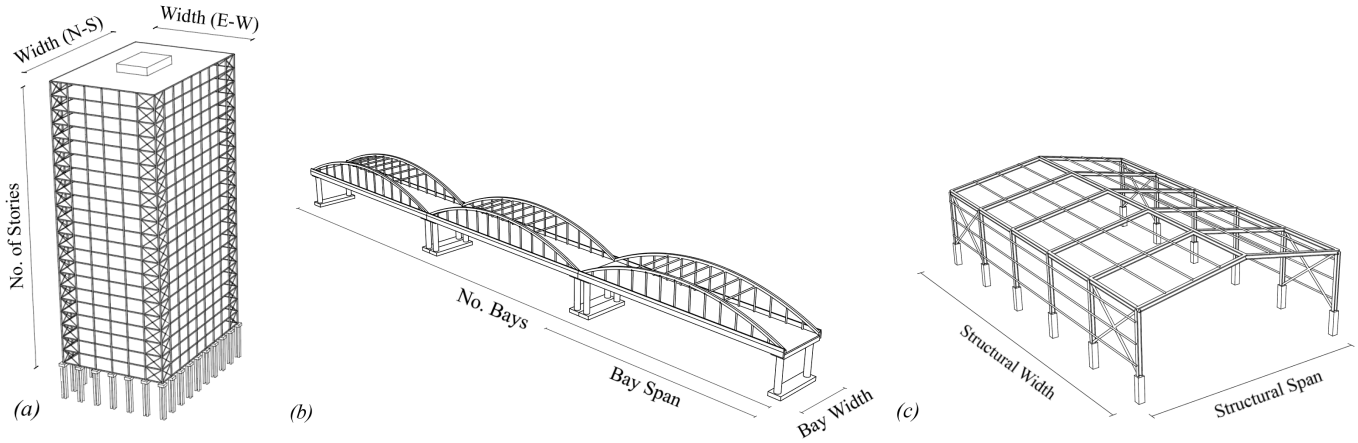


Figure 5.21 – Target (a) high-rise structure, (b) tied-arch bridge, and (c) steel frame structure

5.3.4.4 Representation benchmarking

To demonstrate the representation limitations of grid-based methods and benchmark the proposed hypernetworks for the developed structural element dataset, an additional voxel mesh dataset was created. Voxel models are conventionally limited to fixed sampling grid resolutions due to the memory requirements of high-resolution grid datasets and the popularity of convolutional architectures, typically implementing grid sizes of 32^3 [120-122], 64^3 [123, 124], or 128^3 [125]. While some examples of high-resolution voxel models have been examined [329, 330], these methods exhibit reduced scaling capacities for large environments, computationally expensive training, and limited inference applications due to the cubic relationship between grid resolution and memory usage. For this benchmarking analysis, each of the base target structural geometries was downscaled to a grid size of 128^3, matching the resolution capacity of established large convolutional voxel models. The mean IOU of these voxel-based meshes relative to the ground truth mesh dataset was then calculated to examine the limitations of a voxel approach within the context of this research and benchmark the proposed hypernetworks, the results from which are provided in Table 5.5.

5.3.5. Results

To demonstrate the performance of the proposed method, each of the proposed models has been evaluated independently. These include the fine-tuned NER model for extracting the necessary design parameters for each task, the hypernetwork models used to generate the structural geometries within an ensemble architecture, and the transformer models that generate the translation vectors used to move these elements within the design space. The reconstruction performance of the ensemble hypernetworks trained for each design task is provided in Table 5.5. The weighted average testing IOU of these ensemble arrangements across all 20 unique structural elements examined within this research was 0.7868, demonstrating the capacity of the hypernetwork method to learn a diverse range of manifold topologies. Analogously, the average standard deviation of the testing IOU was only 0.0306, 0.0112, and 0.0089 across structural elements for the high-rise, tied arch bridge, and steel frame structures, respectively. Low variance IOU metrics indicate that for all design tasks, the hypernetworks within each ensemble performed consistently, which is particularly compelling given the diversity of tested structural forms and the overall complexity of the final designs themselves.

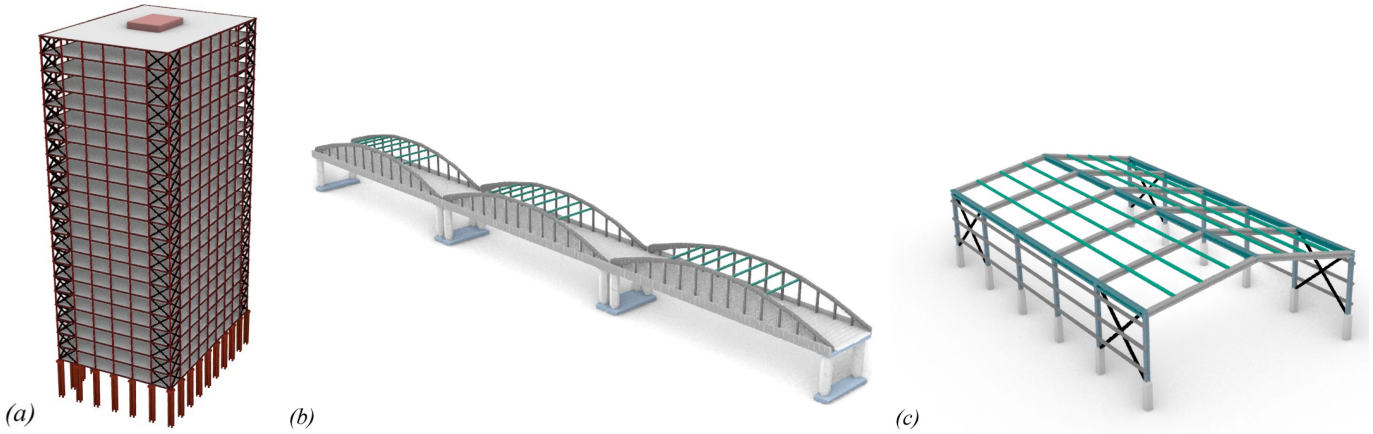


Figure 5.22 – Rendered hypernetwork (a) high-rise, (b) tied-arch bridge, and (c) steel frame structure

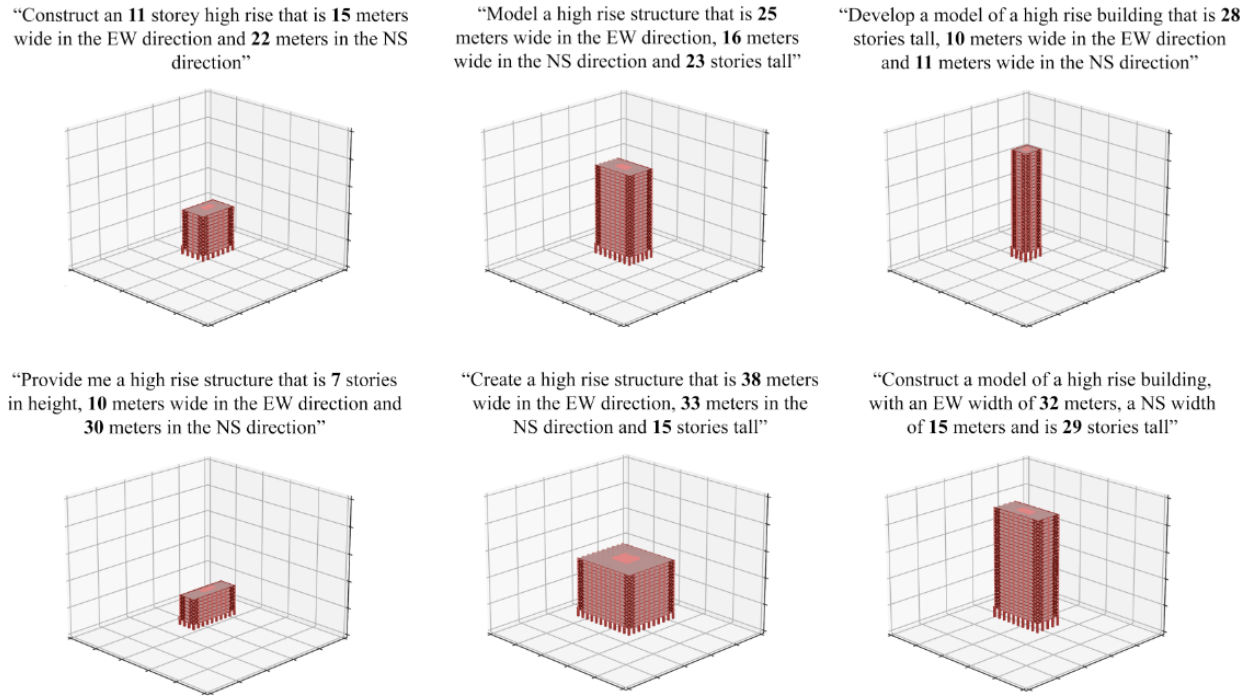


Figure 5.23 – Examples of reconstructed high-rise designs from text-based user commands

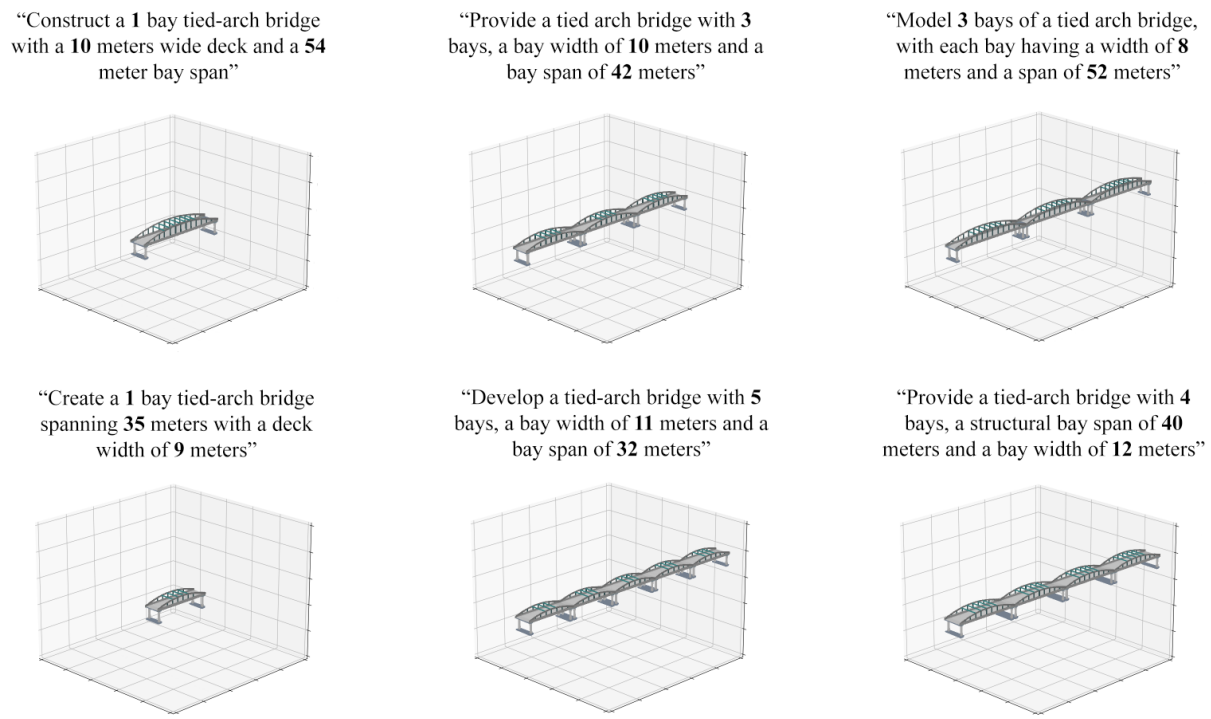


Figure 5.24 – Examples of reconstructed tied arch bridge designs from text-based user commands

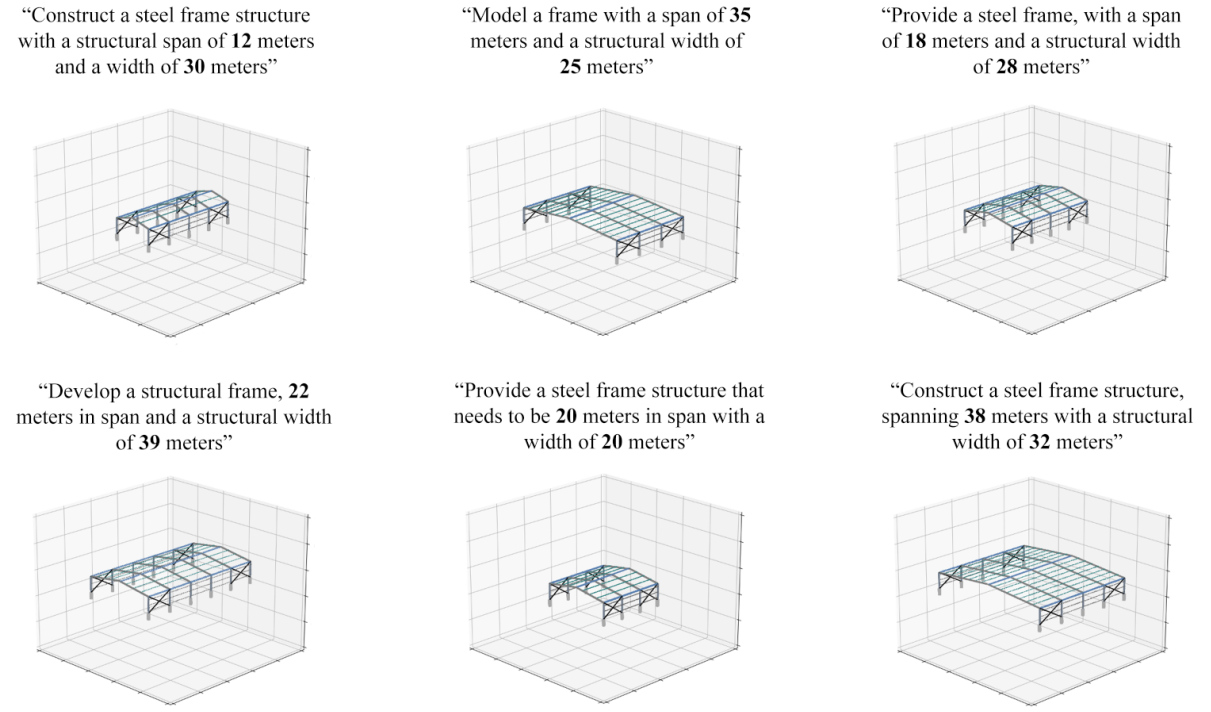


Figure 5.25 – Examples of reconstructed steel frame designs from text-based user commands

Table 5.5 – Performance of hypernetwork models for element reconstruction

Design Task	Element	IOU_{Train}				IOU_{Val}				IOU_{Test}			
		Max	**Mean**	**Min**	**STD**	**Max**	**Mean**	**Min**	**STD**	**Max**	**Mean**	**Min**	**STD**
Highrise	Column	0.6731	0.6209	0.5668	0.0236	0.6678	0.6220	0.5687	0.0235	0.6691	0.6212	0.5676	0.0234
	Floor	0.8973	0.8378	0.7645	0.0217	0.8988	0.8382	0.7717	0.0218	0.8976	0.8379	0.7664	0.0221
	Beam	0.9221	0.5874	0.3260	0.0953	0.8515	0.5849	0.3367	0.0988	0.8510	0.5911	0.3390	0.0965
	Core	0.9632	0.9497	0.9277	0.0059	0.9703	0.9498	0.9279	0.0060	0.9632	0.9496	0.9281	0.0060
	Foundation	0.9286	0.9055	0.8745	0.0098	0.9252	0.9050	0.8739	0.0100	0.9270	0.9055	0.8774	0.0098
	Cross bracing	0.8951	0.8102	0.7471	0.0253	0.8920	0.8114	0.7539	0.0258	0.8893	0.8104	0.7505	0.0255
Tied Arch Bridge	Deck	0.8302	0.7556	0.7176	0.0163	0.8033	0.7598	0.7350	0.0167	0.7597	0.7468	0.7329	0.0099
	Arch	0.7206	0.6766	0.6243	0.0317	0.7323	0.6743	0.6194	0.0318	0.7206	0.6726	0.6206	0.0305
	Bracing	0.7434	0.7427	0.7417	0.0003	0.7429	0.7426	0.7421	0.0002	0.7432	0.7426	0.7422	0.0002
	Rail	0.8745	0.8569	0.8371	0.0092	0.8748	0.8581	0.8422	0.0089	0.8746	0.8574	0.8369	0.0092
	Foundation	0.9211	0.9020	0.8795	0.0115	0.9209	0.9063	0.8861	0.0088	0.9210	0.9033	0.8854	0.0092
	Column	0.7425	0.7367	0.7306	0.0027	0.7404	0.7369	0.7315	0.0027	0.7422	0.7377	0.7316	0.0033
	Hanger	0.9194	0.9019	0.8657	0.0170	0.9191	0.9009	0.8636	0.0164	0.9195	0.9017	0.8652	0.0164
Steel Frame	Cross bracing	0.8942	0.8125	0.7190	0.0318	0.8920	0.8126	0.7220	0.0390	0.8794	0.8173	0.7247	0.0335
	Foundation	0.9610	0.9606	0.9601	0.0001	0.9612	0.9606	0.9601	0.0002	0.9612	0.9606	0.9600	0.0002
	Purlins	0.8533	0.8497	0.8464	0.0012	0.8549	0.8498	0.8459	0.0016	0.8546	0.8497	0.8468	0.0014
	Column	0.7322	0.7311	0.7300	0.0005	0.7322	0.7310	0.7300	0.0005	0.7319	0.7311	0.7300	0.0005
	Wall Girt	0.8817	0.8711	0.8619	0.0056	0.8798	0.8702	0.8616	0.0052	0.8792	0.8706	0.8613	0.0052
	Eave strut	0.8205	0.8046	0.7872	0.0113	0.8211	0.8053	0.7901	0.0096	0.8208	0.8065	0.7887	0.0099
	Roof beam	0.7789	0.7563	0.7335	0.0116	0.7767	0.7580	0.7346	0.0113	0.7767	0.7545	0.7349	0.0113

The performance implications of the proposed ensemble hypernetworks are further reaffirmed when compared against the benchmark metrics recorded for the voxel dataset, provided in Table 5.6. These results reflect the highest possible representation capacity of a voxel-based model relative to the original target meshes, assuming the voxel model is capable of perfectly reconstructing the voxel meshes at a grid resolution of 128^3. Remarkably, these benchmark IOU values indicate that across all 20 base structural element types, the trained hypernetwork models outperformed the maximum possible representation capacity of voxel-based approaches at the upper bound of conventional grid resolutions. On average, the mean testing IOU results recorded from the hypernetwork ensembles outperformed the target voxel dataset by a factor of 9.42, 2.70, and 6.06 for the highrise, tried arch bridge, and steel frame designs, respectively. This is largely due to the size of the design space relative to the base structural geometries that are being predicted. Research within the field of 3D reconstruction typically focuses on the generation of individual geometries of a similar normalised size, allowing the significant resolution limitations of voxel-based methods to be overcome by defining the reconstruction grid proximate to the bounding box of the target mesh. In contrast, this research implements a design space that is often orders of magnitude larger than the individual predicted structural geometries. As a result, the loss of representation capacity that is inherited by grid-based voxel methods is exacerbated within the context of the tested highrise, tied arch bridge, and steel frame design tasks. The scale of the design space is

also shared across the task-specific structural geometries present within the database developed for this research, such that fine-detailed geometries that occupy a smaller relative volume reflect a substantial reduction in reconstruction performance when relying on a voxel-based fixed grid approach.

Table 5.6 – Voxel representation capacity limitations

Design Task	**Element**	**Voxel Mean** IOU_{Target}
Highrise	Column	0.0807
	Floor	0.1592
	Beam	0.0577
	Core	0.7260
	Foundation	0.2220
	Cross bracing	0.0290
Tied Arch Bridge	Deck	0.4502
	Arch	0.3847
	Bracing	0.1609
	Rail	0.5035
	Foundation	0.5584
	Column	0.4324
	Hanger	0.1547
Steel Frame	Cross bracing	0.2163
	Foundation	0.6950
	Purlins	0.0794
	Column	0.1862
	Wall Girt	0.0794
	Eave strut	0.1859
	Roof beam	0.1025

In addition to reconstruction performance, the accuracy of the predicted translation vectors used to manipulate these geometries has also been evaluated in Table 5.7, documenting the MSE loss between the ground truth displacements and the predictions provided by the trained transformer models. Remarkably, the transformer models tasked with translation vector prediction performed consistently for training, validation, and testing datasets across all 20 trained instances. The standard deviation of the testing MSE was 5.8745×10^{-4}, 2.6046×10^{-3}, and 1.0097×10^{-4} across the structural components of the high-rise, tied arch bridge, and steel frame task. Furthermore, the average testing MSE for these respective experiments was 2.8713×10^{-4}, 2.0157×10^{-3}, and 7.6480×10^{-5}, highlighting the capacity of the trained transformer models to accurately predict translation vectors to construct a direct range of structures. Notably, vectors produced to build the steel frame designs achieved the lowest error when evaluated against the ground truth testing dataset. The steel frames produced for this research paper represented the smallest structural profiles within the overall structural dataset,

requiring smaller displacements on average to manipulate the hypernetwork geometries during the reconstruction process.

Table 5.7 – Performance of transformer encoders for translation vector creation

Design Task	Element	MSE_{Train}	MSE_{Val}	MSE_{Test}
Highrise	Column	3.4426×10^{-5}	3.5509×10^{-5}	3.5755×10^{-5}
	Floor	1.2026×10^{-5}	4.1204×10^{-5}	3.0000×10^{-4}
	Beam	4.8678×10^{-7}	4.8789×10^{-7}	4.9635×10^{-7}
	Core	8.0339×10^{-7}	8.3342×10^{-7}	8.7536×10^{-7}
	Foundation	5.5542×10^{-5}	5.5811×10^{-5}	5.5658×10^{-5}
	Cross bracing	1.1000×10^{-3}	1.1000×10^{-3}	1.6000×10^{-3}
Tied Arch Bridge	Deck	1.0000×10^{-4}	1.0000×10^{-4}	2.0000×10^{-4}
	Rail	9.0031×10^{-5}	4.5426×10^{-5}	1.0000×10^{-4}
	Arch	2.0000×10^{-4}	1.0000×10^{-4}	2.0000×10^{-4}
	Hanger	1.0000×10^{-3}	2.5000×10^{-3}	1.3000×10^{-3}
	Bracing	2.0000×10^{-4}	2.0000×10^{-4}	4.0000×10^{-4}
	Column	9.0000×10^{-4}	1.0000×10^{-3}	7.0000×10^{-4}
	Foundation	6.0000×10^{-4}	1.0000×10^{-3}	5.0000×10^{-4}
Steel Frame	Roof beam	8.3971×10^{-5}	1.0000×10^{-4}	1.0000×10^{-4}
	Foundation	3.0000×10^{-4}	3.0000×10^{-4}	3.0000×10^{-4}
	Purlins	7.7623×10^{-5}	8.0118×10^{-5}	1.0000×10^{-4}
	Column	2.0000×10^{-4}	2.0000×10^{-4}	1.1000×10^{-3}
	Cross bracing	6.8526×10^{-6}	7.6168×10^{-6}	8.6965×10^{-6}
	Wall Girt	1.2248×10^{-5}	8.7886×10^{-6}	1.1353×10^{-5}
	Eave strut	6.8458×10^{-6}	6.1723×10^{-6}	5.5285×10^{-6}

Finally, the performance of the fine-tuned NER transformer model is provided in Table 5.8, documenting the accuracy of the models when extracting the relevant design criteria from the user commands across all three of the chosen experiments. The average extraction accuracy for the training, validation, and testing datasets was 96.54%, 96.56%, and 96.89%, respectively. These results indicate that the proposed BERT-based NER model adequately extracted the necessary design parameters for each of the design tasks, before being passed to the ensemble hypernetwork and transformer models to reconstruct and manipulate structural elements within the design space. Inherent to these results is the assumption that the user has not provided incomplete information within their input prompt and that the extracted design criteria does not require further postprocessing beyond normalisation, the implications of which are discussed further within the limitations section.

Table 5.8 – Performance of transformer-based design parameter extraction

Design Task	Extracted parameters	Training		Validation		Testing	
		Loss	Accuracy (%)	Loss	Accuracy (%)	Loss	Accuracy (%)
Highrise	Width (N-S), Width (E-W), No. Stories	0.0002	94.17	0.0001	94.55	0.0001	94.60
Tied Arch Bridge	No. Bays, Bay Span, Bay Width	0.0004	98.52	0.0002	98.52	0.0002	99.01
Steel Frame	Structural Width, Frame Span	0.0003	96.92	0.0002	96.61	0.0002	97.06

5.3.6. Discussion

The performance of the proposed hypernetwork method may be further examined by breaking down reconstruction accuracy according to topology. Notably, the proposed hypernetwork models excelled at constructing thick structural elements that occupy larger volumes within the design space when compared against slender structural components. For example, the weighted average testing IOU value for column elements across each design task was 0.6261, while the reconstructed foundations recorded a weighted testing IOU of 0.9076, constituting a 44.95% improvement in predictive performance. These results indicate that the SDF generated from slender structural elements are more challenging to learn when compared to larger elements within the model dataset. This trend is significantly more expressed when comparing the relative difference in the target IOU performance of these classes for the voxel-based mesh dataset. For example, bulkier foundation voxel meshes provided a significantly better representation of the original geometries than column voxel meshes, recording an average increase in IOU performance of 259% across the design tasks. This further serves to highlight the consistency of the proposed hypernetwork method across a range of topological forms, exhibiting significantly less variance in representation capacity relative to the voxel dataset.

In addition to providing output structural geometries at a relatively consistent and reliable reconstruction accuracy, the proposed ensemble hypernetworks also significantly outperformed the representation capacity of the voxel-based mesh dataset. The average mean IOU values within the testing datasets were 0.7860, 0.7946, and 0.8272 for the high-rise, tied arch bridge and steel frame design tasks, respectively. In contrast, the downscaling process required to build the voxel-based dataset yielded an average target IOU of 0.2124, 0.3778, and 0.2207 for the same design tasks. This is particularly compelling evidence to support the proposed hypernetwork implementation, as a voxel-based model at this resolution would also need to exhibit perfect prediction accuracy to achieve these results.

Further comparison may also be drawn between the ensemble hypernetworks and established voxel-based methods to investigate the memory efficiency of the proposed method. For context, the voxel-based models 3D-RVP [331], Pix2Vox-A [332], and JSTO network [333] use a total of 181 million, 114.24 million, and 143 million parameters, respectively. Smaller voxel 3D reconstruction models have also been examined, such as the

15.37 million parameter SAR 3D model [334] and the 7.41 million parameter Pix2Vox-F [332]. In contrast, each of the individual hypernetwork models evaluated within this research only required 6.57 million parameters, which can be combined in an ensemble arrangement as necessary to scale the overall approach according to the computational requirements of the task or the complexity of the target designs.

Furthermore, implicit neural field representations are typically significantly more memory efficient than the ground-truth model datasets that they are trained to reconstruct. The structural element dataset for the highrise design task alone was 12.5 GB in size, while the corresponding ensemble translation vector transformers and hypernetworks have a memory footprint of only 1.7 GB. This lightweight adaptable architecture allows for the proposed models to be hosted in existing CAD workflows, offering unique memory-efficient structural reconstruction tools for enterprise software and cloud-based platforms. For context, the widely adopted enterprise CAD software SolidWorks requires at least 20 GB of disk space, 16 GB of RAM, and a dedicated GPU for optimal performance within a virtualised environment. Additionally, individual 3D BIM files that are typically being developed within applied multidisciplinary design environments can often grow to several gigabytes in size [335], illustrating the growing hardware expectations of modern design applications and the relative efficiency of the proposed method.

Finally, by training the hypernetwork, NER transformer, and vector encoder transformer as separate models, the individual components of the proposed method may be substituted, retrained, and modified according to the needs of a specific design task without impacting the remaining models. While hypernetworks were selected to generate the structural components for each design task, the proposed translation vectors are interchangeable with alternative foundational approaches such as voxel-based models or mesh deformation templates. While the representative performance of the hypernetwork method is currently well-established within the available literature [119, 240, 336-338], the proposed translation method is non-discriminatory for alternative reconstruction models that are developed in future research. By substituting the type of model used to construct the base structural elements, future research may seek to apply the same ‘text-to-structure’ models with a vastly different representation form, such as parametrised building information modelling (BIM) elements, deformation meshes, voxels, or point clouds.

5.3.7. Limitations

The cumulative modularity and memory efficiency of the proposed method provide a compelling solution that can be applied across a range of industries, including 3D printing, additive manufacturing, architectural rendering, structural visualisation, and engineering modelling. However, there are several limitations that have been inherited by the documented implementation that must be addressed in future research. Importantly, the proposed ‘text-to-structure’ models utilise simple structural geometries which are scaled to complex designs

using novel translation vectors. Central to this implementation is the assumption that a target design can be reproduced through the repetition and manipulation of simple structural elements. This assumption furthermore contributed to the parametric approach to dataset development for each of the base structural classes, which each have a relatively simple and uniform design language. In this sense, design complexity is induced through the predicted translation vectors, rather than the hypernetwork models themselves. Additional forms of predictive element manipulation may need to be introduced to extend this concept to designs that are comprised of unique or complex individual structural forms. In this sense, while the proposed method demonstrates a significant improvement in its capacity to build complex 3D designs, there remains a representation gap between the proposed training dataset and real-world structures.

This is a specific limitation of the underlying datasets generation process, which was selected based on rules-based methods provided within the available literature [321, 322] and the challenges associated with capturing, segmenting, and labelling comprehensive real-world design datasets. While parametric methods of dataset generation are prominently used across academic deep learning tasks to extend design applications beyond generalised benchmark datasets [339-342], future research may seek to develop alternative methods of creating large, complex, multi-class, real-world structural datasets. Compelling methods that are being investigated to tackle these dataset generation challenges within parallel fields of research include graph-based structural labelling [343], segmentation from 3D environment scans [344], predictive structural massing from satellite imagery [345], point-cloud informed parametric models [346], algorithmic optimisation for indoor scenes [347], and generative 3D meshing from photos and videos [121, 348].

There are also several unique modifications that should be investigated to deploy the proposed 'text-to-structure' models within alternative research domains, such as building information modelling or the design of interactive computer-aided design software. For example, the proposed models rely on the exact translation vectors provided by the trained transformer models to connect structural components, which does not guarantee perfect element connectivity throughout the overall design. This may be overcome by rounding the translation vectors to exact grid-based intervals or using object snapping and alignment to enforce global and local vector constraints. This is a particularly relevant change to apply the proposed 3D reconstruction model within the context of conventional computer aided design software, which fundamentally relies on these tools to ensure design connectivity [349].

An additional challenge that must be investigated to apply the proposed method within the context of traditional CAD software involves the potential variability of user inputs at inference. The proposed method assumes that the user has defined all of the necessary design parameters within their input prompt, which are unique for each given design task. Ambiguous or incomplete user inputs are a known limitation of conventional NER models,

which struggle to accurately extract target entities when processing out-of-distribution vocabulary, misleading phrasing, or insufficient contextual information [350]. BERT was selected as the base model for this task due to its bidirectional attention mechanism, which extracts meaning from ambiguous and missing words by evaluating semantic relationships within the overall input sentence [351], typically outperforming word-level embedding models such as Word2Vec and GloVe [352]. Despite the compelling performance of BERT for challenging NER tasks, additional input preprocessing techniques may also be explored to deploy the proposed 'text-to-structure' model within a robust real-world design environment. Recent research has demonstrated the versatility of methods such as text normalisation [353], entity mapping [354], and coreference resolution [355] to translate potentially challenging user inputs into a standardised format for downstream predictive analysis. To handle sparse or catastrophically incomplete user inputs, future research may also investigate methods of generative prompt completion [356], probabilistic Bayesian parameter estimation [357], or sampling missing parameters from a predefined distribution [358].

5.3.8. Summary of key findings from Section 5.3

The novel reconstruction techniques introduced within this research present a significant improvement in representation capacity for foundational approaches—bridging the gap between academic geometry reconstruction research and practical engineering applications. The introduced translation vectors allow predicted base structural elements to be replicated and manipulated throughout the design space, building complex structural systems that each host large quantities of interconnected elements. These encoder predicted vectors are also broadly applicable to models beyond the scope of this research, providing an entirely new method to scale foundational reconstruction methods to new representation frontiers. The ensemble hypernetwork approach is lightweight and segmented, allowing individual models to be substituted according to the memory capacity of the design environment or the complexity of the given design task. Finally, the results from the investigated design tasks indicate that the 'text-to-structure' models are collectively capable of accurately mapping unique user design prompts to vast structural systems, demonstrating compelling performance in terms of memory efficiency, reconstruction accuracy, and output scalability.

5.4. Conclusion

This chapter documents the development of an entirely novel 'text-to-structure' hypernetwork framework, which is targeted towards improving existing architectural and engineering 3D modelling design workflows. To achieve this, the proposed model aims to demonstrate a significant improvement in output scene complexity, extending possible future research applications to encompass the prediction of designs with hundreds or even thousands of interconnected structural elements. The transformer-based translation vectors that were introduced to manipulate these structural elements throughout the design space are also broadly applicable to other foundational models beyond reconstruction hypernetworks. Finally, optimal structures generated in section 5.2 via an algorithmic approach have demonstrated the compelling capacity of metaheuristics to build high-quality

design datasets, which be used to train machine learning models such as hypernetworks for predictive 3D modelling tasks.

CHAPTER 6: ENHANCING STRUCTURAL HEALTH MONITORING SYSTEMS WITH FIBRE BRAGG GRATING SENSORS AND NEURAL NETWORKS

6.1. Introduction

Remarkably, the solutions proposed throughout this thesis which bridge targeted gaps between academia and industry are also directly applicable throughout many phases of the asset lifecycle. The compelling machine learning models that have been specifically developed for design tasks are directly translatable to the field of structural health monitoring. Section 6.2 introduces a neural network approach to predicting strain baseline datasets, which are interpolated by downstream NURBS-based mapping to produce high-resolution structural response heatmaps. This framework is radically improved in section 6.3 through the adaptation of previously developed hypernetwork architectures within an applied SHM context. The novel hypernetworks designed within this chapter are capable of converting discrete sensor readings into implicit neural field representations with theoretically unlimited resolution, allowing the strain response at any point within the structure to be predicted and visualised with rapid inference. By replacing the signed distance training format discussed in chapter 5 with sensor-driven strain profiles, this chapter enables newfound predictive capabilities throughout the field of structural health monitoring — demonstrating real-time predictive monitoring with unprecedented output resolution

6.2. A NEURAL NETWORK BASED DIGITAL TWIN MODEL FOR THE STRUCTURAL HEALTH MONITORING OF REINFORCED CONCRETE BRIDGES

T. Hielscher[1*] , S. Khalil[1] , N. Virgona[1] , S.A. Hadigheh[1]

[1]School of Civil Engineering, Faculty of Engineering, The University of Sydney, Sydney, New South Wales 2006, Australia

*Corresponding Author: Tommy Hielscher

Structures, https://doi.org/10.1016/j.istruc.2023.105248, Published in November 2023

6.2.1. Abstract

Developments in Structural Health Monitoring (SHM) research over the past few decades have demonstrated potential in optimising maintenance solutions for degrading infrastructure. The scale of structural deterioration worldwide and the inadequacy of current non-destructive evaluation techniques necessitate the adoption of accessible, quantitative, continuous SHM technology into mainstream asset management practices. This paper seeks to address this significant demand by proposing a robust, end-to-end, fibre-optic sensor (FOS) monitoring prototype which utilises deep neural networks to convert FOS strain output into an interactive digital twin (DT) visualisation. Finite-element validation demonstrated that the prototype was capable of capturing reliable

structural analytics, recording an average error of < 2 kNm and an absolute error of < 0.15 mm for bending moment and deflection respectively. Furthermore, the predictive mean absolute error of the integrated artificial neural network was less than 1με during testing, demonstrating the accuracy of the digital twin when generating baseline strain data for structural analysis.

6.2.2. Introduction

Public and private authorities that manage large infrastructure projects have a responsibility and vested interest in maintaining the longevity of their assets. The structural health monitoring (SHM) methodologies which are applied to these projects conventionally prioritise minimal service disruption, efficient resource allocation, and a reduction in the degree of operational risk. The value provided by effective SHM systems cannot be overstated. In the United States, there are 625,000 bridges listed by the Department of Transportation, 30% of which have been officially classified as either structurally deficient or functionally obsolete as of 2015 [21]. The ASCE's reporting has directly reflected this, collectively grading US bridge conditions a 'C+' in 1998, 'D' in 2017 and 'C' in 2021 [359]. In 2021, the bipartisan US Infrastructure Investment and Job Act (IIJA) was passed to address the aging state of the built environment, allocating $110 billion USD towards the repair of roads and bridges out of an estimated total of $1.2 trillion in funding [360]. Globally, the cost of corroding infrastructure alone is expected to exceed $1.8 trillion, as a single facet that contributes towards the degradation of assets over time [361].

Furthermore, a BRIME report concluded that across three different European countries – France, Germany and the UK – deficiencies within highway bridges were present at rates of 39%, 30%, and 37% respectively [22]. Traditional techniques of non-destructive evaluation (NDE) for the monitoring of these assets have also been severely limited to routine visual examinations which are unable to provide continuous information into the long-term behaviour of these structures. A technical report published by the U.S Federal Highway administration concluded that 95% of element condition ratings provided by state department inspectors vary between two rating points of the average, demonstrating the significant variability of subjective visual inspections [38]. The inaccurate evaluation techniques of current maintenance practices result in economically impractical maintenance schedules and uninformed resource allocation for asset management at an industry scale [362]. The cumulative effects of these stagnant practices and mass structural deterioration pose a significant risk to the built environment, with the potential to produce a marked decline in serviceability over time.

While the term SHM is used to describe a broad domain of data-collection and maintenance practices, in the context of this research SHM is the process of using technology to obtain real-time information to identify and predict possible damages of a building structure over its service life [363]. As outlined by Ko and Ni [364], the main criteria for bridge SHM systems are the:

- validation of design assumptions and assessment of structural conditions to provide design precedents for similar structures;
- identification of anomalies in structural response for early damage detection, especially after disasters or extreme events;
- comparison and assessment of structural response over time to allow for optimal maintenance strategies; and
- collection of substantial amounts of real-time, onsite data to investigate the effects of design actions on structures for the purposes of research.

6.2.2.1 Digital Twins for Structural Health Monitoring

Throughout the published literature within the field of Digital Engineering, the term 'Digital Twin' (DT) has been subject to a broad range of definitions. Given the encompassing reach of the field, it is critical for researchers to provide a clear and consistent definition of what a digital twin refers to, both within the context of the proposed innovation and the existing literature. The UK governments Digital Built Britain program defines DT models as *"a realistic digital representation of assets, processes or systems in the built or natural environment"* [365]. More granular definitions of successful DT models emphasise the importance in an ongoing symbiotic connection between digital and physical assets, described by Dang et al. as *"a high-fidelity digital mirror of the physical entity; the former evolves synchronously with the latter throughout their entire life cycle"* [366]. The ongoing development and adaption of DT models with respect to the life cycle of physical assets is specifically relevant to the field of structural health monitoring. In recent years, SHM digital twin models have enabled this feedback between digital and physical assets through the introduction of sensor and signal interrogation systems to measure the current state of the asset. These measurements typically account for some structural metric that industry practitioners conventionally leverage to evaluate structural health, including strain [367], displacement [368], and vibration sensors [369].

In order to clearly define the criteria for a successful digital twin and evaluate the alignment of the proposed DT model to the established literature, a total of eight necessary capabilities have been outlined in Table 6.1. These key features of conventional DT models were summarised by Ye et al. through consultation with nine experts within the field of structural health monitoring, building information modelling, asset management, and digital engineering [370], providing a framework to evaluate the validity of the digital twin solution proposed within this research project.

Table 6.1 – Key criteria for a successful Digital Twin

Key Digital Twin features	Proposed Digital Twin solution
The model is a *digital replica* of a physical asset.	The model used in the visualisation of the post-tensioned footbridge was directly modelled from engineering drawings and specifications, providing an accurate digital representation of the physical asset.
The model is *connected* to the physical bridge, updating based on feedback from the assets current state in near real time.	The proposed solution captures real-time strain data from Fibre Bragg Grating (FBG) sensors that have been embedded within the footbridge during construction.
Data is leveraged as a key tool by the DT model to provide insights into maintenance, monitoring and inspections.	FBG strain data is post-processed by an interrogation system, a range of structural calculations and a deep neural network to generate further structural insights that are visualised within the DT model. These insights provide asset managers additional tools which facilitate informed operational decisions.
The model spans the *entire lifecycle* of the physical asset	As the sensors are installed during construction, data is collected throughout all phases of the asset's lifecycle. Feedback can be recorded from the concrete curing process to the structural response of the asset towards the end of its service life.
There is a *common data environment* which stores all relevant data for the model.	The software developed for this project contains a local database of post-processed data which is used to visualise the structural response of the asset. This repository enables the ongoing collection of data within a single, consolidated location for model use.
The digital twin may be used as a *visualisation* tool to retrieve data, assisting in communication to stakeholders.	The DT model that has been built within this research project is presented through a custom interactive digital environment, allowing for the retrieval and visualisation of historic loading events along with necessary structural metrics.
The digital twin provides *simulation* capabilities which can be physics based or data driven, allowing for the prediction of asset performance	The baseline strain data generated by the deep neural network provides a simulation of the unloaded state of the footbridge. This dataset is used to calculate the delta strain due to loading, capturing the assets structural performance.
The digital twin can *learn* from data to improve future performance	By integrating a deep neural network within the model workflow, the proposed DT model has the capacity to be retrained as the collected datasets continue to grow, allowing for historic data to improve future predictions.

6.2.3. Structural Health Monitoring of a post-tensioned bridge

The novel monitoring strategy that is introduced within this paper aims to provide a holistic tool to record, visualise, and analyse the structural response for a post-tensioned bridge subject to loading events. This solution required the development of a digital twin and deep neural network model, which have been integrated within a combined SHM methodology. Data collected by Fibre Bragg Grating (FBG) sensors are post-processed to visualise the continuous structural behaviour of a post-tensioned footbridge under load. This model-based monitoring strategy allows users to observe abnormalities in the relative strain baseline over time, in addition to the reporting and visualisation of critical structural metrics including bending moment and deflection. As a combined monitoring solution, the digital twin prototype and machine learning models introduced within this paper demonstrate the application of fibre optic sensors within an encompassing structural monitoring strategy. The value provided by this integrated digital twin model may be realised through its ability to facilitate informed operational decisions and improve long-term structural integrity, serviceability, and resourcefulness over the lifetime of a monitored asset. The broader objective of this research is to improve maintenance and operational services as a precedence of an integrated structural monitoring strategy.

The footbridge bridge itself consists of a 17,900 mm long and 2,600 mm wide reinforced concrete deck superstructure at a 1:20 grade, connected by a movement joint to an adjacent walkway and a construction joint to a mezzanine slab. A rectangular-section concrete column provided a pin support near the midspan of the sloped deck, while the landing slab acts as a fixed-end support. The sensor design consists of three sets of two strain sensor arrays, each containing 16 discrete FBG sensors cast in situ at 1.0 m spacing. Documentation of sensor installation during the construction of the footbridge is provided in Figure 6.1. The sensors leveraged for this analysis are attached to the side of the longitudinal reinforcement, with 58mm of cover between the steel rebar and the top surface of the concrete deck. The top and bottom FOS arrays are installed with a 280 mm difference in depth to capture a holistic strain profile of the footbridge. Additionally, there are two sets of temperature arrays with the sensors attached to the rebars at 5.0 m spacing. The individual FOS arrays attached to the longitudinal rebar are spaced 1080 mm apart. The positioning of the temperature and FBG sensors relative to the longitudinal rebar is further outlined in Figure 6.2.

Figure 6.1 – Installation of FBG sensors during construction of the footbridge

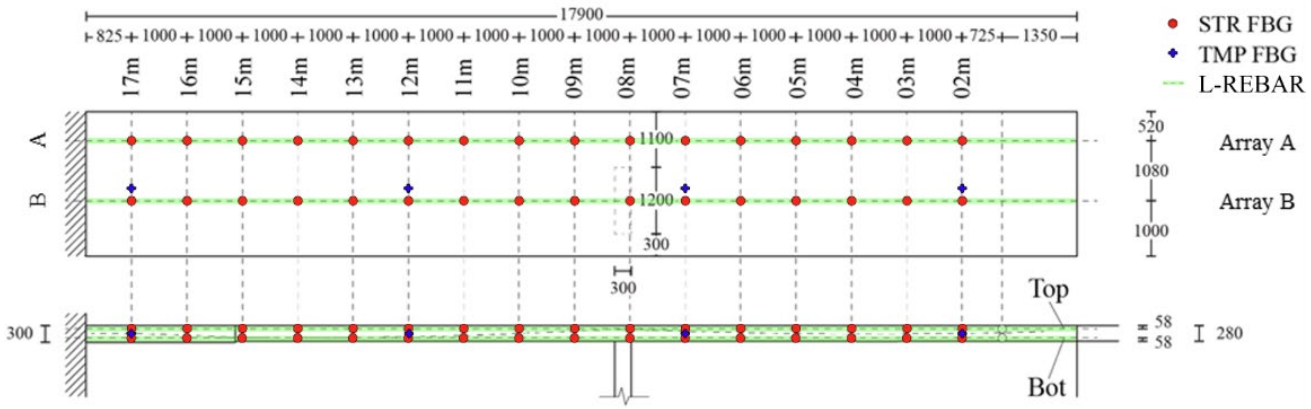


Figure 6.2 – Plan and section of installed FBG sensor layout

A static Luna Hyperion si255 Optical Sensing Instrument connecting each of these arrays is located in a storeroom underneath the bridge. The output is interpreted by the interrogator which measures 16 FBG wavelengths at 160 nm wide channels to 1 picometre precision, equating to ±1 με of total strain or equivalent to a ±0.1℃ change in temperature. The data is processed via a remote computer connected to an on-site interrogator unit. Concurrently, Python scripts were developed to compile the strain and temperature information for each of the sensors as .csv files which are then stored locally on the computer. The data columns pertaining to each of the strain sensors are labelled with the STR_array_Xm nomenclature where 'array' refers to either A or B and X represents the distance in metres from the rightmost column support. The derivation of the delta strain datasets, deflection and bending response are conducted autonomously, including the use of the Artificial Neural Network (ANN) to generate the relevant baseline data. Derived structural responses are then imported into the Digital Twin environment as a parametric user interface for analysis, as presented in Figure 6.7. Finally, finite element modelling was used to confirm the validity of the downstream structural analytics. The methodology designed over the course of this research project is illustrated graphically in Figure 6.3.

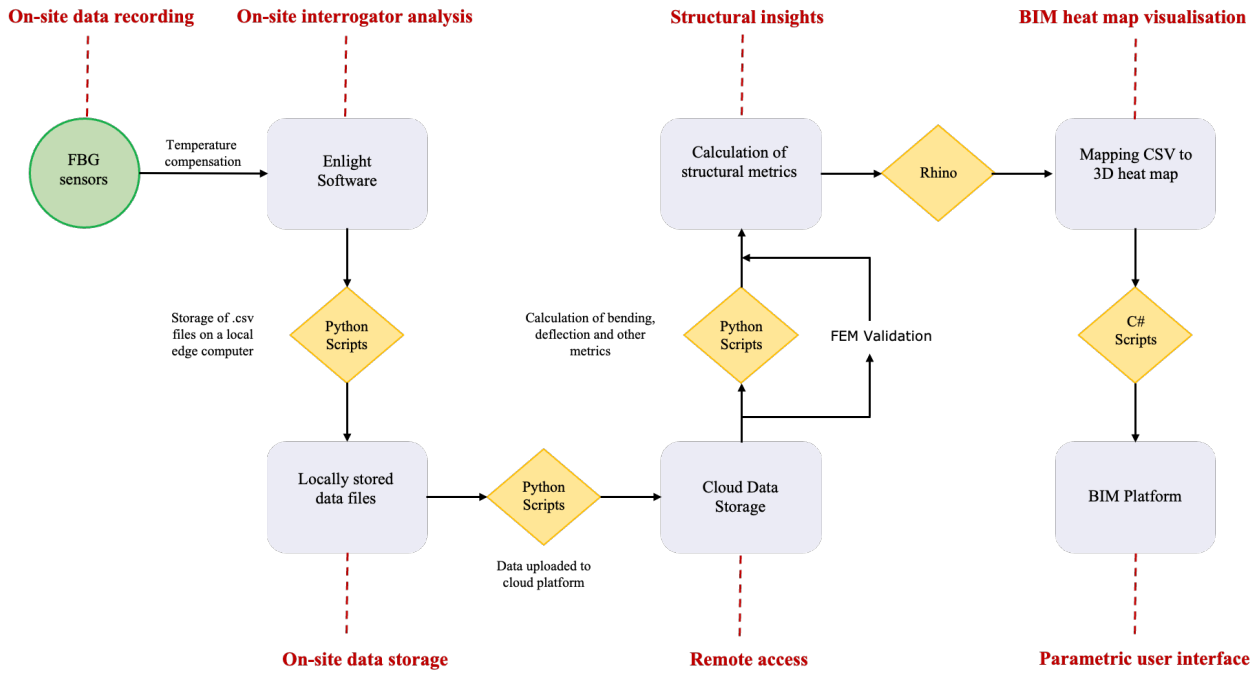


Figure 6.3 – Data architecture of the proposed digital twin

6.2.3.1 Fibre Bragg Grating (FBG) Sensors

Fibre Bragg Grating (FBG) sensors measure the strain at discrete locations along the bridge based on changes in light intensity within an optical fibre. These FOS are 5-7 mm in size and are comprised of the optical fibre, fibre core, and a grating period typically around 500 nm in width. They operate under Bragg's law whereby the spectral properties of the incident light are transformed when it passes through the grating at a particular wavelength [371]. The central Bragg wavelength (λ_b) of the incident light is based on changes in the grating period (Λ) and effective index of refraction (n_{eff}).

$$\lambda_b = 2n_{\text{eff}}\Lambda \tag{6.1}$$

FBG sensors have immense utility in SHM due to their profound ability in accurately measuring strain and temperature. While most of the sensors on the market are electric-based, FBG sensors are preferred for long-term monitoring due to their suitability in highly stressed composite structures and their superiority in measuring thermal and mechanical effects [372]. FBGs are manufactured as small, compact, and lightweight sensors using silica, which is highly immune to electromagnetic interferences and can therefore function well in electrically noisy environments. They are durable and can withstand tensile strains >10,000 με and have the additional capacity for multiple sensors to be multiplexed onto a single optical fibre cable.

The acquisition of FBG strain and temperature information is based on the relative shift in Bragg wavelength signal caused by changes in the effective refractive index and the periodicity of the fibre optic cable grating. FBG sensors are highly sensitive to thermal and chemical effects which is problematic as this can create unpredictability in the observed data stream. The major hurdle is in extracting a true strain measurement after

isolating the data fluctuations induced by thermal contraction and chemical resin shrinkage [373]. Therefore, the temperature on the FBG sensors needs to be compensated by considering the refractive Bragg wavelength shift ($\Delta\lambda_b$) as a linear perturbation of the total strain ($\Delta\varepsilon$) and temperature change (ΔT) as follows:

$$\frac{\Delta\lambda_b}{\lambda_b} = K\Delta\varepsilon + \beta\Delta T \tag{6.2}$$

where K is the strain-optic coefficient and β is the temperature sensitivity factor of the optical fibre.

6.2.3.2 Structural Analysis

The post-tensioned bridge monitored within this research project can be simplified as a beam structure. The application of dynamic point loads, such as from pedestrian traffic, will cause the beam to deform in a manner that is dependent on the boundary conditions specific to this structure. As the beam bends, the longitudinal strain varies linearly with the distance from the beam's neutral axis. However, the neutral axis may be shifted in response to temperature variations or if any physical damage such as cracking was present. These structural complexities can be resolved by defining the curvature, κ, as the difference in strain measurements at two arbitrary positions in the cross-section divided by their distance of separation [374].

Two equidistantly placed strain sensors parallel with the neutral axis can be used in this instance to extract the curvature generated from a mechanically induced load:

$$\kappa = \frac{\varepsilon_{\mathrm{bot}} - \varepsilon_{\mathrm{top}}}{h} \tag{6.3}$$

where $\varepsilon_{\mathrm{top}}$ and $\varepsilon_{\mathrm{bot}}$ represent the top and bottom FBG strain sensor readings and h is the vertical distance of separation between their centre of masses. Assuming that the structure is uncracked during most of its operational life, the bridge behaviour can be represented simply as beam elements exhibiting linear elastic deformations. Here the induced bending moments are a component of the flexural stiffness and the deflections can be estimated through the double integral of curvature. In real time, these parameters can be autonomously retrieved through an exact curvature function obtained using regression methods and with respect to the longitudinal bridge span as follows:

$$\kappa = a_n x_n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \tag{6.4}$$

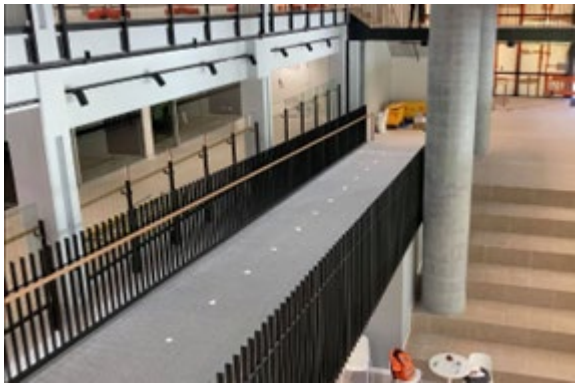
where x is the discrete longitudinal position of a strain sensor along the bridge span and a_0, a_1, …, a_n are the coefficients of a degree n curvature polynomial. For a generic span with n boundary constraints, a maximum $n + 1$ degree for the curvature function shall be used for the entire span to avoid any issues with overfitting of the dataset that may arise with the use of higher degree polynomials. The accuracy of the approximated deflection result however is dependent on the closeness of the sensor spacing and improves as more data is collected [375]. To account for this, the optimal degree of the curvature polynomial may be selected according to the optimal R^2 regression coefficient value which reflects the correlation of the function with the dataset [376]. For structural calculations presented within this paper, the modulus of elasticity of the concrete, E_c, was

taken as 38 GPa as obtained from concrete testing and for steel, E_s, it was taken as 200 GPa. The reinforcement bars were taken as having a yield strength 500 MPa and the post-tensioned slab was taken to have a compressive strength of 50 MPa.

6.2.3.3 Experimental Load Testing

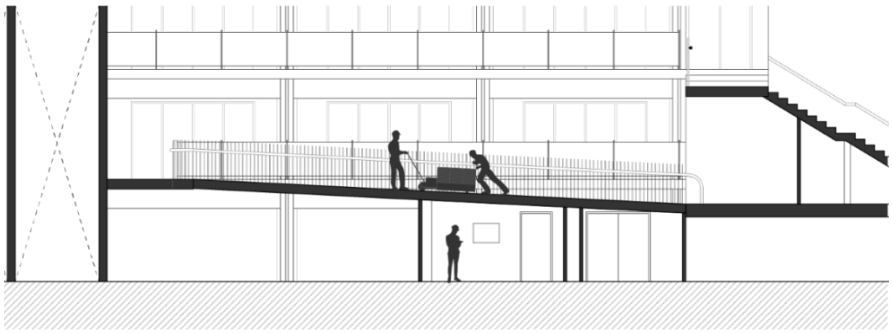
The site experiment was conducted nine months after the concrete deck was poured. The primary objective of the experiment was to analyse the behaviour of the FBG system under a known loading condition and to calibrate the embedded sensors. The experimental data would serve to verify the data architecture, whereby a DT would be able to visualise the structural effects of the experimental load consistent with a finite element analysis.

The applied load consisted of two operators guiding a pallet jack carrying 32 weights across the bridge, with a combined mass of 1.1 tonnes. The loading of the footbridge across marked sensor locations is documented in Figure 6.4, including the selected testing apparatus. For the purpose of analysis, the load of the crate and operators was considered as a uniform pressure acting over a 2.0 m long × 1.0 m wide area. The locations of the strain-FBG sensors in Array A were marked on the bridge based on the construction drawings. The calibration of the data was performed by moving the load from mezzanine slab end to the walkway landing, with the load centred over each marked sensor at one-minute intervals. The duration of the test was 960 seconds in total, excluding set-up and movement off the bridge.



(a)

(b)



(c)

Figure 6.4 – (a) Footbridge with marked FBG locations, (b) pallet crate load, and (c) section of experiment

6.2.3.4 Initial design considerations

The site experiment was designed to ensure that the derived structural calculations were reliable and that they can be used for subsequent structural analysis. Several design considerations, however, needed to be accounted for in the calibration of these sensors. Firstly, when an FBG sensor is embedded into a structure and is subjected to structural deformation, a strain transfer mechanism is initiated between the optical fibre and host structure. Thermal variations create a prominent structural effect, leading to volumetric changes in the optical fibre impacting the accuracy of the recorded strain measurements [377]. Therefore, any individual set of strain readings cannot be interpreted directly as the fibre optic cables will be subjected to varying degrees of initial bending due to the construction process and temperature [378]. Instead, the relative change in the strain reading from an original reference level (referred to as the baseline) needs to be considered to accurately represent the structural deformations [379]. Figure 6.5 highlights this 'delta strain' ($\Delta\varepsilon$) shift on a sample FBG dataset as initiated by a mechanical loading event.

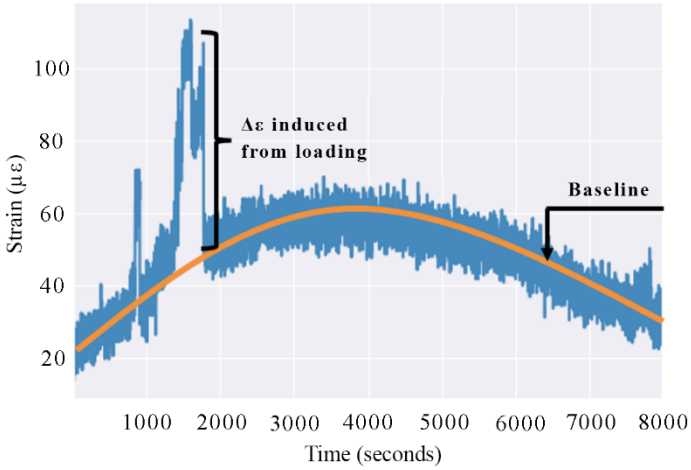


Figure 6.5 – Relative change in strain measurement induced by a loading event

6.2.3.5 Mechanical and Thermal Strain

Temperature-compensated FBG strain sensors generate total strain measurements which is the summation of thermal and mechanical strain. These total strain readings fluctuate sinusoidally as the structure expands and contracts due to temperature changes. Separation of the mechanical strain is therefore necessary to accurately capture the structural response. This is achieved by taking the coefficient of thermal expansion (CTE), α, of concrete as 10 με/°C, or a result determined from test data as specified in AS5100.5:2017 (Clause 3.1.6(a)) [380] and subtracting it from the total strain:

$$\varepsilon_{\text{mech}} = \varepsilon_{\text{total}} - \alpha\Delta T \tag{6.5}$$

where $\varepsilon_{\text{mech}}$ and $\varepsilon_{\text{total}}$ represent the mechanical and total strains respectively and ΔT is the change in temperature from a reference value. In this format, the delta strain computations do not require complex modelling algorithms as the general strain behaviour is steady and any load-induced deviations can easily be

subtracted from a horizontal baseline. It should be noted, however, that this approach is imperfect as the CTE of Portland cement concrete ranges from about 7.2 to 14.4 με/°C depending on the variation of the concrete's component materials [381].

6.2.4. Proposed machine learning framework for structural monitoring

In order to generate baseline datasets, an artificial neural network was created to model strain behaviour as influenced by temperature variance. Within the context of the load testing experimentation, the proposed neural network was introduced to generate the baseline strain datasets necessary to derive further structural analytics. The neural network model selection was driven through a comparative study from a broad range of machine learning (ML) models. Strain baseline monitoring is a key element of FBG structural analysis, as it provides a relative framework from which deviations and abnormalities may be observed. The strain output from this analysis has been visualised through a digital twin model, which displays historic structural analytics. A comparative analysis was conducted to evaluate the performance of the proposed neural network solution against results obtained through Linear, Ridge, and Lasso regression.

6.2.4.1 Linear regression

As FBG data collection requires the monitoring of temperature proximate to the installed sensor locations, ML models have been successfully implemented in the past to investigate the relationship between the recorded Bragg wavelength and local temperature. This was directly demonstrated by [382], in which thermal variance was identified from strain sensor measurements. For modelling cases in which the dependent and independent variables have an established causal link, simplified linear regression models are a common machine learning solution. This is due in part to their computational efficiency, simplicity, and interoperability of results. Complexity may be introduced to these models if multiple independent variables are used, in which the multivariate linear equation of the model weights may be taken as follows.

$$f(X) = \beta_0 + \sum_{i=1}^{M} X_i \beta_i \tag{6.6}$$

This equation assumes multiple inputs (X), where M is the number of model input variables, β_i is the coefficient of regression, and β_0 is taken as the regression intercept. The coefficient of regression represents the mean change of the dependant variable, strain, and is derived by reducing the sum of squared errors between the predicted strain and the recorded strain data.

6.2.4.2 Ridge and Lasso regression

Notably, the regression coefficients generated by the least squared method have a high degree of variability which may compromise the validity of the model output. Ridge and Lasso regression techniques comparatively produce biased results with a much lower degree of variance by integrating a penalising regularisation

coefficient into the cost function. Specifically, ridge regression utilises a regularisation coefficient to penalise the squared coefficients, illustrated in the following cost function.

$$Ridge = \sum_{i=1}^{N} \left(y_i - \sum_{j=0}^{M} (\beta_j X_{ij}) \right)^2 + \lambda \sum_{j=0}^{M} {\beta_j}^2 \tag{6.7}$$

The penalisation term, λ, effectively regulates the model coefficient values β, while $\hat{y}_i$ and y_i represent the predicted and target values respectively. Within this context, X_{ij} represents the value of the *j-th* feature for the *i-th* data point, across a total of N points. As the regularisation coefficient increases, the regression coefficients shrink towards a minimum value. Comparatively, Least Absolute Selection Operator (Lasso) regression utilises a penalisation term to optimise the absolute weight rather than the square of the weights. For Lasso regression, the cost function is defined as follows.

$$Lasso = \sum_{i=1}^{N} \left(y_i - \sum_{j=0}^{M} (\beta_j X_{ij}) \right)^2 + \lambda \sum_{j=0}^{M} |\beta_j| \tag{6.8}$$

The regularisation techniques within both Ridge and Lasso models serve to prevent overfitting and control the weighted coefficients. Lasso deviates from Ridge regression through its mediation of the absolute penalty function. As the weights are increasingly penalised, Lasso regression shrinks the estimate towards absolute zero, allowing for optimised feature selection and filtering [383].

6.2.4.3 Artificial Neural Network

In order to quantify the performance of the proposed Deep Learning framework, a comparative analysis was conducted across all of the tested model. This included the calculation of the mean squared error (MSE) for each of the model's output, where y is the sampled data and $\hat{y}$ is the model output from N data points. In the evaluation of model performance after the validation phase, it is desirable for the MSE and mean absolute error (MAE) to be as close to zero as possible. Furthermore, in order to test the model's robustness and validity, the distribution of the errors produced must be Gaussian according to the Central limit theorem. Finally, the model performance may also be quantified by calculating the root mean squared error (RMSE) and the coefficient of determination (R^2).

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{6.9}$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{6.10}$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{6.11}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \binom{n}{k}}{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2 \binom{n}{k}} \tag{6.12}$$

6.2.5. Proposed Digital Twin

The heatmap-style meshing process that was designed for this structural monitoring system utilises 50 mm square elements created using the parametric visual programming software Grasshopper. Within this Grasshopper script, several number slider control tools provide the ability to tune the parameters that define the generated geometries and analysis, including the element sizing, critical bounds, timestamp selection and gradient colour scheme. An example visualisation isolated from this Grasshopper script is illustrated in Figure 6.6, including the meshing, interpolation, and visualisation of the desired structural analysis. These control tools are stored with the exported data in a standardised format, allowing the DT tool to import the heatmap-style visualisation and its corresponding defined user tools, outlined in Figure 6.7. This structured transfer of information between the developed scripts and programs allows for the continuous generation and interpolation of structural data within Grasshopper, which can be stored for viewing within the DT interface without requiring the recalculation of the parametric visualisations at each timestamp.

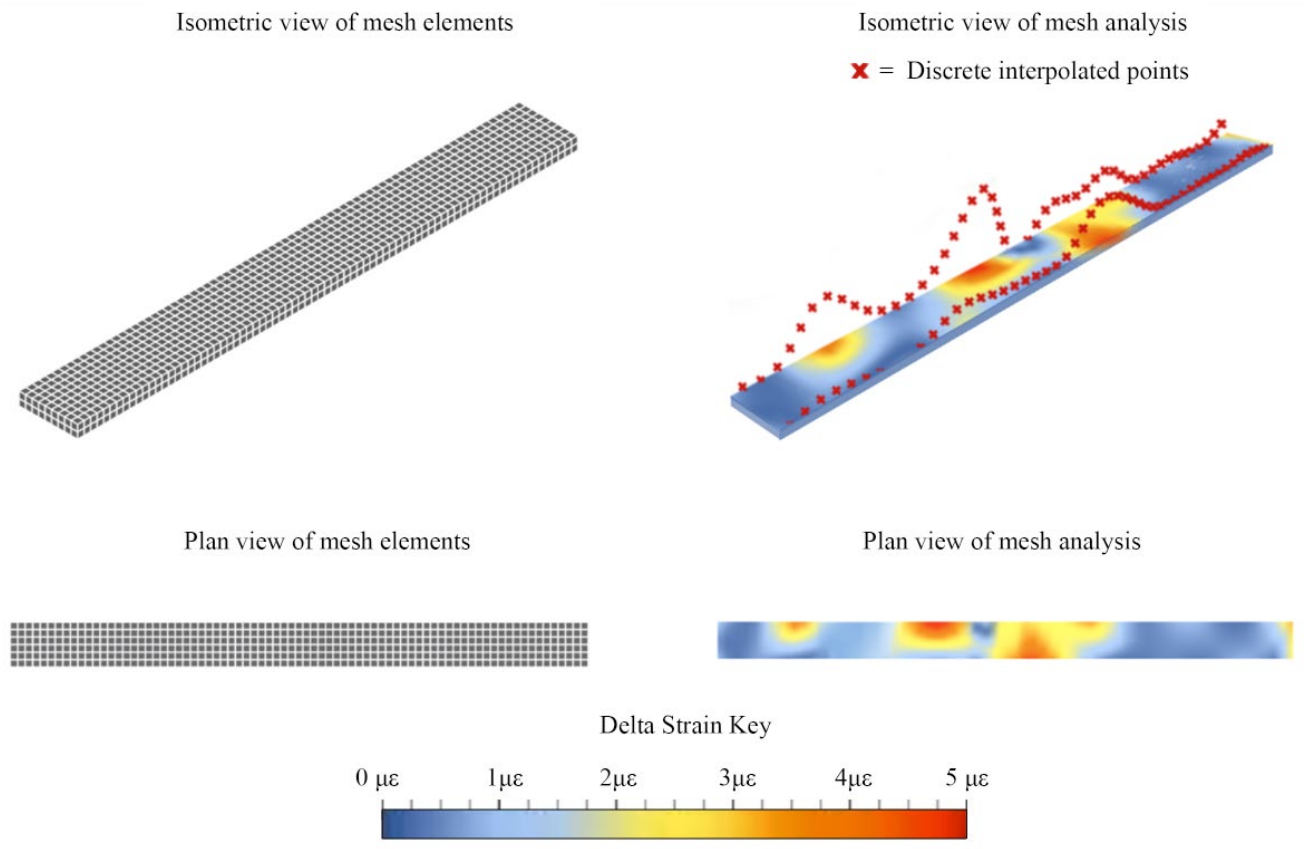


Figure 6.6 – Strain heatmap generation example

6.2.5.1 Digital Twin Prototype

The DT prototype was programmed with the objective of communicating significant quantities of data to provide actionable insights into the structural response of an asset under load. The node structure created within the Grasshopper script allows the interpolated data to be exported as a .csv file, in which the location coordinates

and value of each shell element are categorised and recorded. The generated .csv file is then imported into the DT environment, allowing a user to interact with the generate structural data. This external tool was developed to create a SHM workflow in which the user can directly interact with the analysis in an environment that is directly focused on capturing and visualising structural response. The user interface provided within the prototype controls the mode of analysis, the selected timestamp, the critical structural bounds, and the animated time series. A plotted graph is provided within the UI to directly compare conventional analysis against the heatmap-style visualisation, as a holistic overview of the structural response during the loading event (Figure 6.7). Critical and maximum values are also summarised within the user tools section, such that the user can identify and quantify the induced strain, bending moment and deflection within the structure.

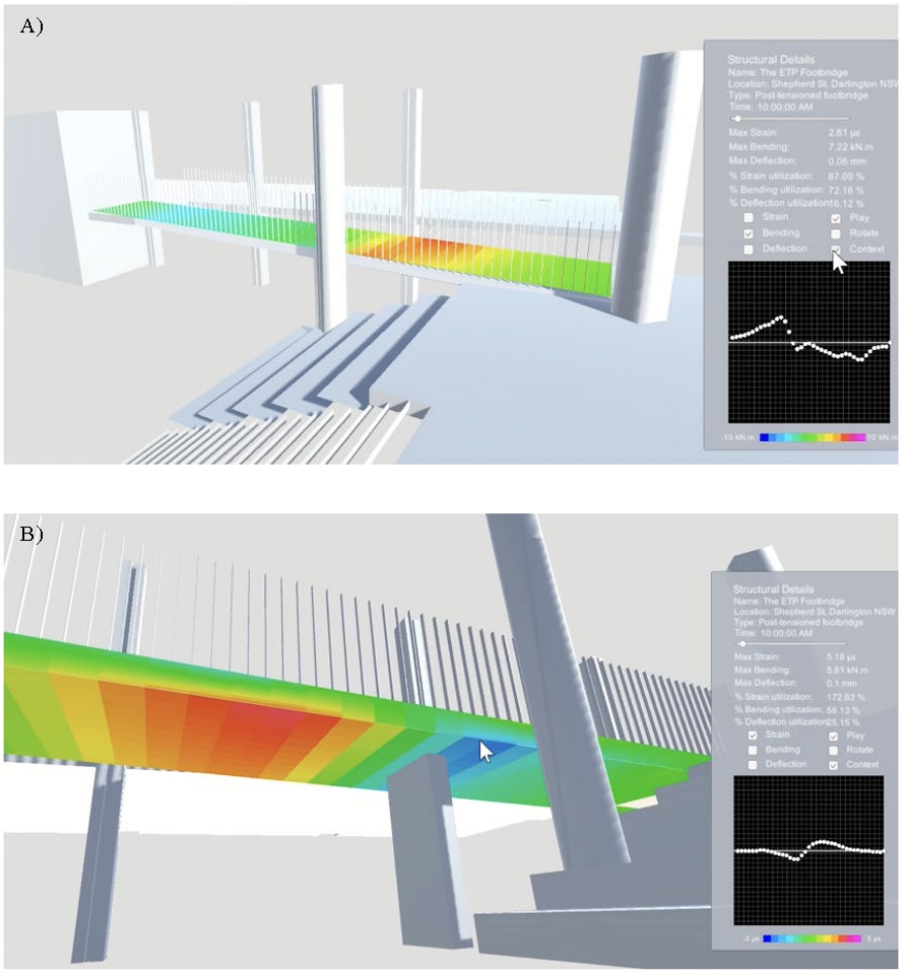


Figure 6.7 – Digital twin visualisation (a) mesh plane for bending – (b) mesh shell for delta strain

6.2.6. Numerical analysis

To validate the observed structural response, the results of the experimental loading condition were compared to a finite element method (FE) model. Model properties and assembly were assigned in accordance with the onsite concrete testing and construction drawings respectively and are given in Table 6.2 and Figure 6.8. The construction joint at the mezzanine slab was modelled as a pinned connection. The concrete deck was modelled

as a solid three-dimensional extrusion of a concrete damage plasticity (CDP) material section in Abaqus. The tensile and compressive reinforcement layers were modelled as wire beam elements with solid circular sections and assembled as per the construction drawings, accounting for the changing section geometry. The post-tensioning ducts were modelled as thin flat rectangular sections of area equivalent to a ⌀25 mm solid circular bar. Geometrically, flat bars provide axial stiffness while minimising the *I*-value [384]. All reinforcement was modelled as embedded in the concrete with a 'rough' no-slip condition, reflecting the effects of the threaded reinforcement. The pallet crate was applied as a point load acting over a 2.0 m long × 1.0 m wide rigid body, corresponding to the assumed application area in the site experiment. A pin boundary condition was applied to the load surface by a reference point. For this FE study, a pin (translation fixed, rotation free) and a rigid (translation and rotation fixed) boundary condition was applied over the entire surfaces of the mezzanine and walkway landing ends respectively. A pin boundary condition was also applied over a 1200 mm × 300 mm area at the bottom surface of the bridge to model the effects of the central column support.

Table 6.2 – Abaqus model parameters

Part	Concrete deck	Longitudinal reinforcement	Stirrups	Post-tensioning duct
Model	3D solid brick	Wire	Wire	Wire
Dimensions (mm)	2600×280×15200 2600×300×2700	⌀10, ⌀12, ⌀16	⌀16	49.1×10
Approx. mesh sizing (mm)	70	70	70	70
Mesh method	C3D8R	B31	B31	B31

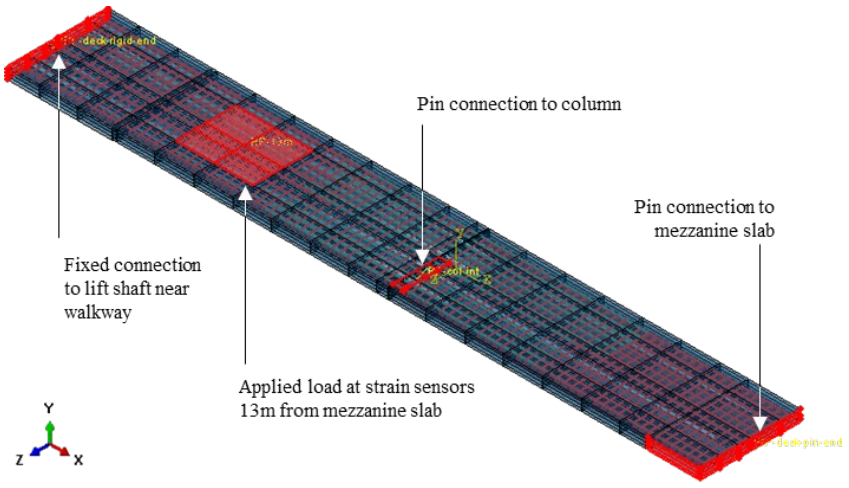


Figure 6.8 – FE model assembly, partitioning, and boundary conditions

A convergence study was undertaken using an *h*-refinement framework to test the validity of the meshing method, the results of which are outlined in Table 6.3. A given three-dimensional brick element is divided into

eight elements, as divisions are made in each axis. For coarse meshes where partitioning would dramatically distort the aspect ratio (AR) of the brick elements (beyond AR = 2), the partitions at the reinforcement were removed, however the column placement and shape was preserved to maintain a consistent boundary condition. The mesh was subdivided equally across all elements along the length of the bridge. A local refinement methodology was considered for the mesh convergence study, as refinement of elements about the area of stress concentration improves accuracy of the results without dramatically increasing the processing time. Refinement was conducted using a coarse mesh equal to the depth of the section with refined meshes constructed until the failure from the processing system at $h = 0.0175$. By calculating and plotting the natural logarithms of average mesh size, h, against relative change in deflection, Δu_y, the mesh convergence rate was derived. From this analysis, the convergence rate (CR) was determined to be 4.36, which implies the mesh was made approximately four times as accurate with every further refined mesh.

Table 6.3 – h-refinement calculations

$\boldsymbol{h}$ (m)	Elements	$\boldsymbol{u_y}$ (m)	$\boldsymbol{\Delta}$ (m)	$\ln \boldsymbol{h}$	$\ln \Delta$
0.280	4388	-0.00402		-1.27	
0.140	11998	-0.00028	0.00374	-1.97	-5.58
0.070	50780	-0.00023	0.00005	-2.66	-9.90
0.035	341520	-0.00022	0.00001	-3.35	-11.64

6.2.7. Results

6.2.7.1 Model performance comparison

In order to evaluate the effectiveness of this end-to-end structural monitoring solution, the results comparison has been divided into two distinct categories: the validation of the various machine learning (ML) models for baseline generation, and the validation of the digital twin prototype which leverages this baseline information. The initial theoretical observation of a linear relationship between the Bragg wavelength and local temperature was useful to identify potentially effective machine learning models for testing. However, due to the stochastic nature of ML prediction, the variance induced by hyperparameter tuning and the disparate functionality of the chosen models, it is not possible to extensively evaluate these models a priori. A critical review of the tested machine learning techniques is therefore necessary in order to identify the most robust and accurate predictive solution. The performance metrics are summarised in Table 6.4 for the testing and training of the selected machine learning models.

Table 6.4 – Statistical metrics of machine learning models

	Training Data				Test Data			
Model	MAE	MSE	RMSE	R^2	MAE	MSE	RMSE	R^2
ANN	1.096	1.923	1.387	0.9960	0.732	0.845	0.919	0.9959
Linear	1.144	1.957	1.399	0.9907	1.143	1.949	1.396	0.9908
Lasso	1.147	1.961	1.401	0.9907	1.133	1.931	1.390	0.9908
Ridge	1.199	2.152	1.467	0.9898	1.203	2.155	1.468	0.9897

Further residual analysis was conducted to extend the performance analysis for the tested models, outlined in Figure 6.9. Models which are systematically biased towards under- or over-prediction will typically exhibit a degree of skewness in the distribution of their errors, such that the model inadequately describes a particular trend that is present within the data. The density distribution histogram provided in Figure 6.9 plots the model prediction errors for each of the machine learning models.

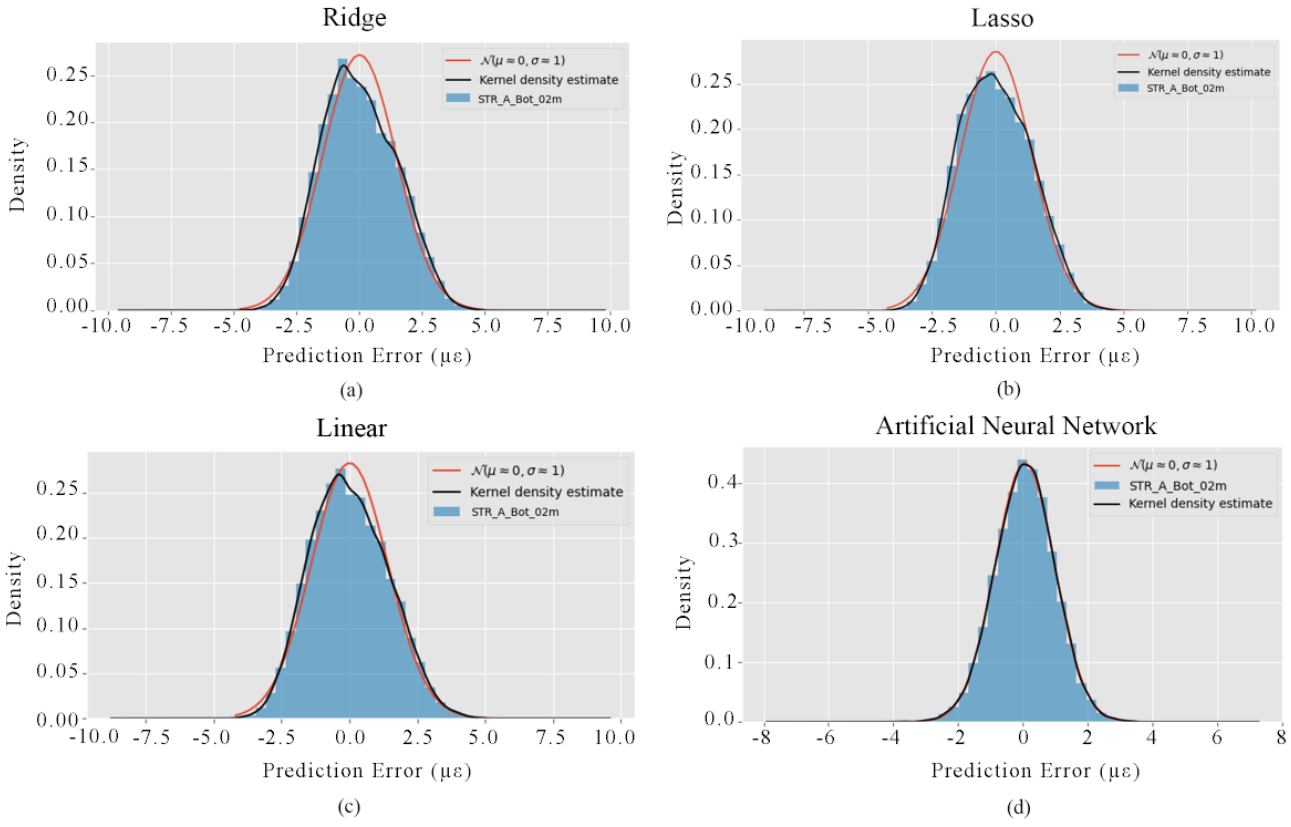


Figure 6.9 – Prediction error density plot for each model

The error density plots provided in Figure 6.9 allow for a direct comparison of the various model residual distributions, in which the idealised Gaussian fit is superimposed as a red curve centred around a mean of zero. The kernel density estimate curve (KDE) represents a non-parametric probability density function that estimates the smoothed shape of the model's residual distribution, generated using the following equation:

$$\hat{p}_n(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) \tag{6.13}$$

where h is defined as a smoothing bandwidth that controls the degree of smoothing that the curve is subject to, X_i is the smoothed data point and $K(x)$ is the Gaussian kernel function. It follows that for each of the tested machine learning models plotted in Figure 6.9, the central limit theorem is satisfied if the black KDE curve is proximate to the red theoretical Gaussian distribution curve. Given this performance criteria, the statistical analysis graphically demonstrates that the residuals produced by the ANN are more normally distributed and homoscedastic relative to the other machine learning techniques that were tested. In contrast, the Ridge, Lasso, and linear techniques exhibit a degree of skewness and asymmetry relative to their target values. It is therefore necessary to conduct further statistical analysis to identify the degree to which the model errors satisfy the central limit theorem for each of the selected ML techniques. This observation aligns with the previous statistical analysis and the conclusions drawn from the preliminary performance metric comparison, demonstrating that the ANN is an appropriate candidate to be used in conjunction with the DT prototype. In order to validate this conclusion further, quantile-quantile (Q-Q) plots of the various modelling techniques are provided in Figure 6.10. For this analysis, the theoretical quantile samples are defined by a Gaussian distribution. A 45-degree red line is superimposed as a guide to demonstrate the case in which the theoretical and sampled quantiles come from identical distributions. Deviation from this theoretical target line signify a departure from the theoretical normal distribution, which is undesirable when selecting an appropriate machine learning model.

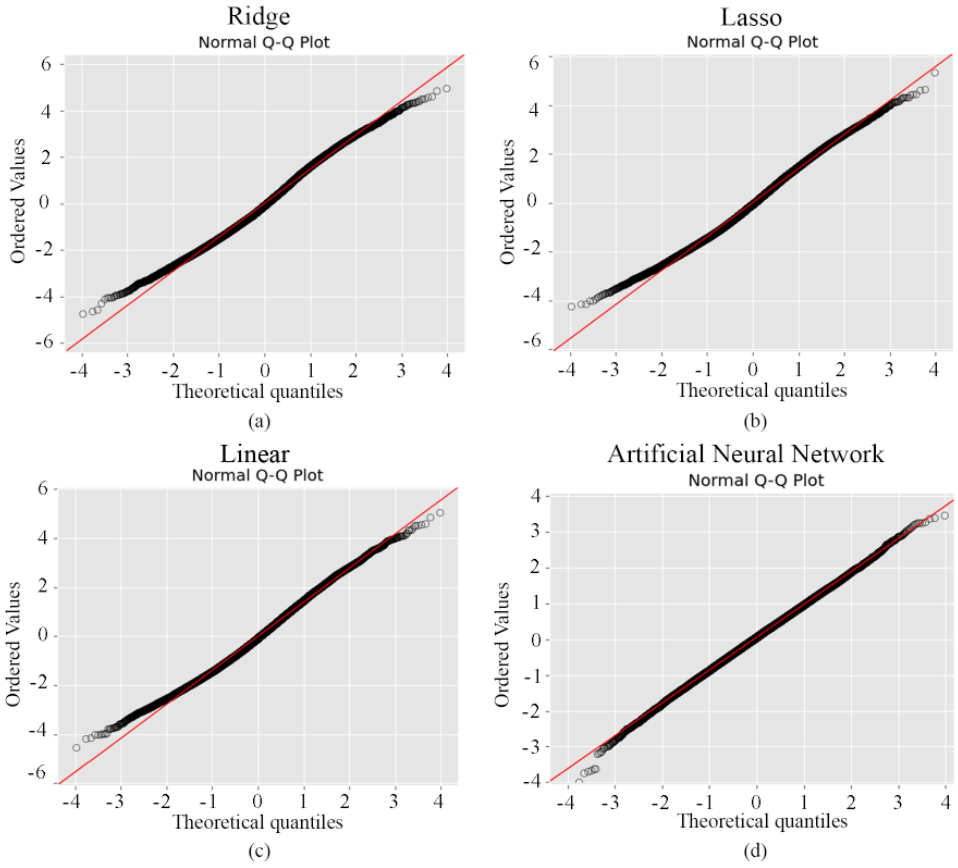


Figure 6.10 – Normal Q-Q plot of residuals for each model

The Q-Q plot provided in Figure 6.10 demonstrates that the ANN residuals are drawn from a symmetric normal distribution due to the linear alignment of the sampled quantiles to the theoretical target line. Deviations from this line are minimal, indicating that the residuals of the ANN are acceptably Gaussian in nature. In comparison, the tails of the Ridge, Lasso and linear Q-Q plots are lightly curved away from the perfect fit, demonstrating that the distribution of their respective prediction errors is less satisfactory. While a degree of deviation is common when visualising ML Q-Q plots, graphs which are excessively "S" shaped indicate that the distribution of the model errors is influenced by skewed prediction data. The extent to which the tails of the Q-Q plot deviate is indicative of the degree to which the model residuals differ from the expected gaussian distribution.

6.2.7.2 Validation of Digital Twin Prototype

The predictive output from the ANN generated the baseline datasets necessary to compute the structural metrics visualised by the DT. Through an analytical study of the induced delta strain measurements caused by the load testing experiment, validation of the key structural metrics in the format of deflections and bending moments was therefore possible. The critical loading case was chosen when the footbridge experienced the most prominent structural deformations, specifically the instance whereby the pallet jack was equidistant between the midspan supporting column and fixed end support (t =13 minutes). With reference to the defined sensor layout, this was located approximately at the strain sensors embedded 13 m from the mezzanine slab. Results obtained from experimental loading were averaged per minute interval to account for the influence of signal noise. In addition to bending moment, the real-time vertical deflections of the bridge were also analysed and, at most, a degree-4 regression polynomial was considered to avoid curve overfitting. The optimum polynomial was selected based on the highest coefficient of determination R^2 after cycling through each of the degrees up to and including 4. The absolute error was used as an indicator of the DT performance used and to compare the outputs generated between the DT model and FE analysis. These results have been summarised in Table 6.5 and 6.6.

Table 6.5 – Comparison of Digital Twin and Finite Element bending moment results

	Bending moment (kNm)															
Array	Str_ 02m	Str_ 03m	Str_ 04m	Str_ 05m	Str_ 06m	Str_ 07m	Str_ 08m	Str_ 09m	Str_ 10m	Str_ 11m	Str_ 12m	Str_ 13m	Str_ 14m	Str_ 15m	Str_ 16m	Str_ 17m
DT-A	0.06	0.19	0.03	-0.71	-1.52	-2.86	-5.02	-3.96	-0.94	2.24	5.11	6.69	7.56	3.94	0.82	-1.69
FE-A	0.80	0.53	0.26	0.25	0.25	-0.54	-0.94	-4.89	0.42	4.05	7.78	7.78	6.71	1.79	-3.81	-9.27
Error	0.74	0.34	0.23	0.96	1.77	2.32	4.08	0.93	1.36	1.81	2.67	1.09	0.85	2.15	4.63	7.58
DT-B	0.17	0.32	0.28	-0.52	-1.52	-2.51	-2.93	-3.83	-0.89	1.95	5.02	7.03	6.47	2.95	0.75	-1.81
FE-B	0.80	0.53	0.26	0.25	0.25	-0.54	-0.94	-4.89	0.42	4.05	7.78	7.78	6.71	1.79	-3.81	-9.27
Error	0.63	0.21	0.02	0.77	1.77	1.97	1.99	1.06	1.31	2.1	2.76	0.75	0.24	1.16	4.56	7.46

Note: DT-A and FE-A denotes Digital Twin and Finite Element respectively for strain Array A.
Note: Absolute error taken between DT and FE results.

Table 6.6 – Comparison of Digital Twin and Finite Element deflection results

	Deflection (mm)															
Array	Str_ 02m	Str_ 03m	Str_ 04m	Str_ 05m	Str_ 06m	Str_ 07m	Str_ 08m	Str_ 09m	Str_ 10m	Str_ 11m	Str_ 12m	Str_ 13m	Str_ 14m	Str_ 15m	Str_ 16m	Str_ 17m
DT-A	0.04	0.06	0.07	0.08	0.08	0.07	0.05	0.03	0.00	-0.05	-0.09	-0.01	-0.11	-0.09	-0.07	-0.03
FE-A	0.01	0.03	0.03	0.03	0.02	0.01	-0.03	-0.08	-0.14	-0.20	-0.22	-0.21	-0.20	-0.15	-0.08	-0.02
Error	0.03	0.03	0.04	0.05	0.06	0.06	0.08	0.11	0.14	0.15	0.13	0.2	0.09	0.06	0.01	0.01
DT-B	0.03	0.04	0.06	0.06	0.06	0.05	0.04	0.02	0	-0.04	-0.07	-0.09	-0.08	-0.07	-0.05	-0.03
FE-B	0.04	0.06	0.06	0.06	0.05	0.04	0.00	-0.05	-0.12	-0.18	-0.21	-0.19	-0.18	-0.14	-0.07	-0.02
Error	0.01	0.02	0	0	0.01	0.01	0.04	0.07	0.12	0.14	0.14	0.1	0.1	0.07	0.02	0.01

Note: DT-A and FE-A denotes Digital Twin and Finite Element respectively for strain Array A.
Note: Absolute error taken between DT and FE results.

6.2.8. Discussion

From the statistical results summarised in Table 6.4, the ANN performed the best across all measured evaluation metrics for both training and testing datasets. Notably, the ANN had the smallest MAE, MSE, and RMSE, indicating that the ANN model was capable of generating baseline strain datasets more accurately and consistently relative to the other tested modelling techniques. Furthermore, the coefficient of determination for the ANN was measurably greater than that of the other machine learning models. In contrast, the Ridge regression model performed the worst across all measured statistical metrics, indicating that the Ridge modelling technique was less effective in capturing and generalising the relationship between strain and temperature. The accuracy of the neural network's predictions is further demonstrated in Figure 6.11, in which the model's output is consistently within the threshold of sensor noise present within the test data, which is approximately ±1 με for the installed fibre optic sensors.

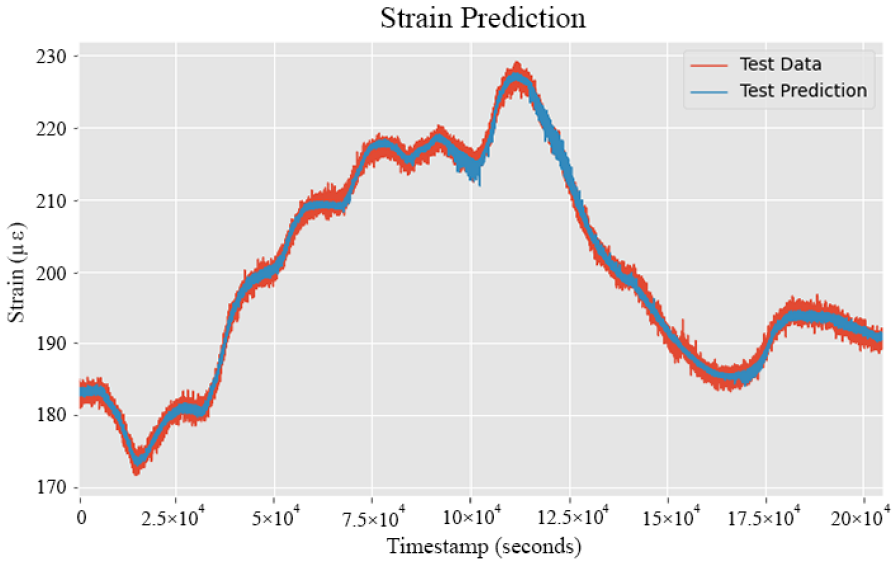


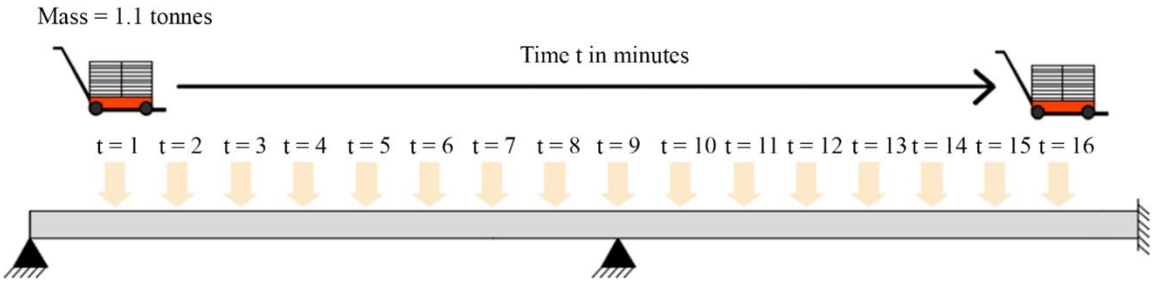
Figure 6.11 – Artificial neural network baseline strain prediction

Notably, the performance metrics of the linear and Lasso regression models are extremely proximate to one another. Conceptually, this is due to the fact that the Lasso modelling technique is a modification of the linear method, in which the model weights experience a penalisation controlled by the regularisation coefficient (λ).

That is, if the hyperparameter λ is taken to be zero, the Lasso model would produce the same coefficients as the linear model for the same given dataset. During the hyperparameter tuning phase, the optimised regularisation coefficient was tuned down to a value of 0.001, significantly reducing the penalisation to which the model weights are subjected. Hence, within the Lasso model environment, the performance was iteratively improved as the model more closely resembled the linear regression method. It follows that the performance variance between the linear and Lasso models is primarily due to the minimal impact of the tuned regularisation coefficient and the stochastic nature of the tested ML models. Further research would be required to demonstrate the feasibility of alternative Lasso hyperparameter tuning techniques in achieving measurable performance improvements over linear regression modelling.

The efficacy of the footbridge DT was assessed on its ability to replicate the structural deformations induced by loading to a high level of precision. FE modelling was used in this validation to determine the expected numerical values for deflection and bending moment for the loading test. The post-processed longitudinal bending moments were mostly consistent, with FEM recording an average absolute error of less than 2 kNm, excluding strain sensors in close proximity to the boundary constraints. This is also true for the deflection calculations measuring a negligible highest absolute error of only 0.15mm. While the overall alignment of the analytical computations with FE method results is promising and demonstrates reliability in the proposed processing algorithms, the system's inability to account for the discrepancies near to the fixed end remains a challenge (>7 kNm absolute error) and warrants further review in future investigations. It is hypothesised that the cause of this error is due to the decision not to model the complex three-dimensional effects of the adjoining platform, which would likely induce additional spring stiffness. This was outside the scope of this project, which focused primarily on two-dimensional structural analysis within the digital twin.

These results show a general agreement between FE analysis and the DT, with outputs represented in a heatmap-style visualisation. The bending moments were mapped to a 2D geometric plane using 16 × 2 cell values (each corresponding with a distinct sensor location) to visualise the real-time structural response. Figure 6.12 illustrates this dynamic map for bending moment which clearly demonstrates the movement and corresponding deformations of the bridge during the loading experiment.



t = 1	0.90	0.25	-0.20	-0.18	0.05	0.81	1.94	1.12	-0.08	-0.53	-1.53	-1.66	-1.42	-0.85	-0.30	0.10	Array A
	0.67	-0.44	-0.16	-0.21	0.12	0.60	1.36	1.12	0.02	-0.46	-1.12	-1.29	-1.28	-0.98	-0.65	0.07	Array B
t = 2	1.57	0.90	0.23	-0.36	-0.74	0.14	1.85	1.02	-0.27	-1.29	-1.55	-1.93	-1.96	-1.40	-0.46	-0.50	Array A
	1.16	0.22	0.18	-0.43	-0.38	0.24	0.98	0.71	-0.64	-0.97	-1.64	-1.52	-1.60	-1.27	-0.90	0.09	Array B
t = 3	0.89	1.30	1.64	0.33	0.01	0.34	1.39	0.70	-0.17	-1.01	-1.51	-1.69	-1.62	-0.54	-0.32	0.70	Array A
	0.62	0.86	0.97	0.54	0.22	0.33	0.86	0.72	-0.43	-1.00	-1.40	-1.64	-1.45	-0.92	-0.52	0.53	Array B
t = 4	-0.29	0.45	2.14	1.85	0.98	0.49	0.84	0.47	-0.74	-1.28	-1.75	-1.91	-1.66	-1.20	-0.37	0.52	Array A
	-0.15	0.20	1.36	1.70	1.30	0.63	0.71	0.45	-0.56	-1.17	-1.42	-1.70	-1.36	-0.72	-0.43	0.88	Array B
t = 5	-0.41	-0.56	1.24	2.26	2.51	1.35	0.59	0.22	-0.51	-1.28	-1.67	-1.59	-1.89	-1.13	-0.72	0.43	Array A
	-0.38	-0.24	0.35	2.23	2.43	0.94	0.21	0.33	-0.48	-1.49	-1.69	-1.72	-1.39	-1.29	-0.60	0.35	Array B
t = 6	-0.64	-1.03	-0.06	1.27	2.66	1.99	0.42	0.35	-0.97	-1.40	-1.99	-2.31	-1.85	-1.05	-1.01	-0.07	Array A
	-0.63	-0.68	-0.42	1.36	2.59	2.12	-0.05	0.09	-0.66	-1.45	-2.01	-1.95	-1.85	-0.97	-0.60	0.13	Array B
t = 7	-0.48	-0.53	-0.67	0.38	1.11	2.11	-0.25	0.84	-0.49	-1.22	-1.82	-1.16	-2.01	-1.30	-1.20	-0.24	Array A
	-0.35	-0.38	-0.33	0.09	1.42	1.56	0.00	0.67	-0.30	-0.81	-1.67	-1.79	-1.44	-1.42	-0.99	-0.22	Array B
t = 8	0.05	-0.15	-0.35	-0.21	0.35	0.98	-0.46	1.35	0.79	-0.31	-1.02	-1.29	-1.53	-1.35	-0.81	-0.25	Array A
	0.18	-0.35	-0.12	-0.11	0.50	0.75	0.17	1.42	0.57	-0.16	-0.90	-1.46	-1.42	-0.92	-0.94	0.04	Array B
t = 9	0.10	-0.07	-0.13	-0.54	-0.57	-1.22	-2.05	1.52	2.75	1.79	0.49	0.76	-0.72	-0.83	-1.16	-0.59	Array A
	-0.05	0.31	-0.13	-0.51	-1.08	-1.37	-1.50	1.04	2.69	1.82	0.53	-0.44	-0.91	-0.81	-1.08	-0.51	Array B
t = 10	0.32	0.04	-0.16	-0.51	-1.48	-2.72	-4.49	-0.95	3.09	4.02	3.79	1.78	0.49	-0.31	-1.01	-1.27	Array A
	0.05	0.43	0.22	-0.40	-1.79	-2.18	-3.10	-0.86	2.81	3.73	3.27	2.12	0.38	0.16	-1.17	-1.10	Array B
t = 11	0.30	0.38	-0.05	-0.85	-1.75	-3.40	-5.91	-2.85	2.26	5.27	5.96	4.31	2.74	0.81	-0.87	-1.58	Array A
	-0.11	0.54	0.31	-0.77	-1.84	-2.70	-3.53	-2.85	1.85	4.61	5.30	4.36	2.43	0.57	-1.04	-1.43	Array B
t = 12	0.06	0.30	0.23	-0.76	-1.65	-3.50	-5.70	-4.15	0.38	3.99	6.78	7.01	5.17	2.10	-0.35	-1.71	Array A
	0.36	0.11	0.32	-0.81	-2.06	-2.79	-3.79	-3.74	0.51	3.64	6.30	6.22	4.80	2.06	-0.10	-1.65	Array B
t = 13	0.06	0.19	0.03	-0.71	-1.52	-2.86	-5.02	-3.96	-0.94	2.24	5.11	6.69	7.56	3.94	0.82	-1.69	Array A
	0.17	0.32	0.28	-0.52	-1.52	-2.51	-2.93	-3.83	-0.89	1.95	5.02	7.03	6.47	2.95	0.75	-1.81	Array B
t = 14	0.14	0.17	0.11	-0.79	-1.27	-2.41	-3.97	-3.22	-1.30	0.94	3.04	6.08	7.78	4.91	1.96	-0.94	Array A
	0.24	0.28	0.25	-0.55	-1.45	-2.05	-2.65	-3.37	-1.41	0.69	3.09	5.75	6.66	4.35	2.05	-1.27	Array B
t = 15	0.16	0.27	-0.10	-0.22	-0.69	-1.13	-1.73	-2.11	-0.83	-0.46	0.90	2.69	4.44	5.23	4.37	1.80	Array A
	0.25	0.28	-0.19	-0.33	-1.00	-0.74	-0.93	-1.96	-1.00	-0.21	1.33	2.12	4.22	4.44	4.10	0.46	Array B
t = 16	-0.19	0.37	0.07	-0.36	-0.36	-0.28	-0.09	-0.63	-0.74	-0.16	-0.12	0.45	1.38	2.36	3.92	1.42	Array A
	-0.19	0.37	0.07	-0.36	-0.36	-0.28	-0.09	-0.63	-0.74	-0.16	-0.12	0.45	1.38	2.36	3.92	1.42	Array B

Figure 6.12 - Digital twin bending moment map

6.2.9. Research limitations

While the novel data architecture has been successful in reproducing the FBG strain and temperature information into practical structural metrics for SHM, it remains limited in its design and functionality, specifically:

- The embedment of fibre optic cables in post-tensioned concrete structures is an imperfect procedure. Stretching and bending of the cables during installation affects the magnitude of the Bragg wavelength

recorded by the interrogator. The degree to which this variance influences the proposed strategy will vary between projects.

- The rate to which any specific structural element expands or contracts depends on a multitude of factors including the ambient temperature, humidity, and presence of direct solar radiation. For outdoor structures, the thermal variation can be significant and should therefore be listed as a scope for further research.
- The efficacy of the structural approach relies on curvature-based rotations. The assumptions and calculations outlined in this paper are suitable for post-tensioned simply supported bridge structures. These assumptions will need to be reassessed for other bridge geometries such as arch, suspension, or steel bridges.
- The simplification of the bridge structure that was used to improve the speed of structural calculations will not be fully applicable for alternative structures. The elimination of the 1:20 grade and generalisation of the structure's boundary conditions were contributing factors towards the discrepancy between the DT model output and finite element calculations, which were justified within the scope of this research project. These will need to be assessed on a case-by-case basis.

6.2.10. Acknowledgements

This research was conducted through a collaborative partnership between the University of Sydney and Laing O'Rourke. We would like to thank our colleagues at Laing O'Rourke: Angus McFarlane, Monica Hanus-Smith, Liam Condon, Marcello Poli, Dillian Gonzales, Benjamin Sweedman, and Georgina North for their continual involvement in this project and for providing the necessary resources and support. Without the help of this collaborative partnership, this project would not have been possible. Hadigheh would also like to acknowledge the University of Sydney Nano Institute and supports he received from New South Wales (NSW) Smart Sensing Network (NSSN) via project number CT35811.

6.2.11. Summary of key findings from Section 6.2

According to the calculated delta strain, deflection, and bending moment datasets, the DT solution provided a comprehensive summary of the structural response to experimental loading conditions. Notably, the prototype errors present within the delta strain calculations were primarily influenced by signal noise and baseline accuracy, while the bending moment and deflection calculations for the DT and FE analysis were also subject to a series of structural, material, geometric, and boundary condition assumptions. Despite this potential variability, the critical values of strain, deflection, and bending moment derived by the DT program were sufficiently proximate to the output of the FE model, indicating the validity of the solution as a supplementary SHM tool. These results indicate that the monitoring strategy outlined in this paper warrants further research into the application of FOS systems for SHM.

Furthermore, across all of the evaluated performance metrics, the ANN designed within this project produced strain baseline data sets that were more accurate, robust, and consistent than the Linear, Lasso and Ridge ML models. These results accumulatively demonstrate that the ANN was the most capable ML technique used to generalise relationships and trends present within the discrete FBG strain and account for any subtle non-linear response that was induced within the structure. With respect to all eight of the key criteria for successful digital twin models derived from the available literature, the proposed deep learning DT model provides an innovative application of digital twinning for the applied purpose of structural health monitoring. As a solution specifically designed for the monitoring of post-tensioned bridge assets, the DT prototype and ML models introduced within this paper demonstrate the application of FOS within a holistic SHM strategy. The value provided by this novel DT model may be realised through its ability to facilitate informed operational decisions and maintenance procedures, in turn improving long-term structural integrity, serviceability, and resourcefulness over the lifetime of a monitored asset.

6.3. HYPERNETWORKS FOR REAL-TIME STRUCTURAL HEALTH MONITORING OF BRIDGES USING EMBEDDED FIBRE BRAGG GRATING SENSORS

6.3.1. Summary

The current state of sensor-driven structural health monitoring research is limited by a conventional reliance on low-dimensional analysis that fails to capture the full complexity of structural behaviour. This chapter introduces a novel hypernetwork model to overcome this problem, trained to predict the strain response at any point within the monitored structure using embedded Fibre Bragg grating sensors. The inference performance of the proposed hypernetwork model allows for the prediction of high resolution three-dimensional structural analysis heatmaps in real-time with unprecedented speed, accuracy, and resolution. The proposed hypernetwork exhibited remarkable strain prediction capabilities, achieving a mean absolute error of 0.0171 $\mu\varepsilon$, 0.0285 $\mu\varepsilon$, and 0.0294 $\mu\varepsilon$ across the training, testing, and validation datasets, respectively. The coefficient of determination R^2 of the hypernetwork residual errors was approximately 0.998 for both the testing and validation datasets, indicating that only 0.2% of the variance observed in the target strain datasets was not explained by the trained model. Furthermore, the maximum absolute error of the proposed hypernetwork observed across all predicted training, testing, and validation heatmap cells was 1.6265 $\mu\varepsilon$, 1.5078 $\mu\varepsilon$, and 1.5636 $\mu\varepsilon$, respectively. The outcomes of this research demonstrate the compelling representation capacity of hypernetworks for predicting cell-based strain heatmaps, with significant potential to enrich modern structural health monitoring systems.

6.3.2. Introduction

Structural health monitoring research (SHM) has evolved significantly over the past decades, transitioning from an overreliance on inefficient forms of non-destructive evaluation to applied sensor technologies that capture enormous amounts of data. These live SHM systems have the potential to offer asset managers and engineers critical insight into the health of monitored structures, enabling early detection of potential issues, facilitating preventative maintenance, enriching structural understanding under complex loading conditions, and speeding up response time during natural disasters or sudden catastrophic structural events [385-387]. Through this transition however, the field of structural health monitoring has inherited a ‘Big-Data’ problem. Modern structural health monitoring solutions struggle to extract value from the sheer volume of data recorded by integrated sensor systems, resulting in complex methods of data parsing that are computationally inefficient for real-time applications. SHM has emerged within this context as an academic field that has a strong emphasis on advancing technological capabilities, however there remains a growing need to address the fundamental challenges that these technologies present [388]. At an industry scale, infrastructure owners and operators remain largely unconvinced that the upfront cost of these sensor systems can be reconciled by slow, low-resolution solutions that produce vast quantities of unused data [389]. Despite meaningful advances within the field itself, the current era of SHM has insufficiently captured the needs of modern infrastructure owners and operators [390-394]. This has driven researchers to investigate new breakthrough methods of analysing sensor recordings. However, leading solutions remain limited in their capacity to meaningfully generate and visualise

analysis from these vast datasets. Existing publications within this field primarily present structural analytics through two-dimensional plots and time series analysis, with limited integration of geometric context with respect to the monitored structure itself [395-404]. Within a field that has historically relied entirely on thorough visual inspection of structural elements, it is perhaps not surprising that low-dimensional visualisations have not motivated widespread investment from asset owners.

Low-dimensional analysis has become an unfortunate staple of contemporary structural health monitoring research. Often referred to as a problem of *'missing measurements'* [405], discrete sensor technologies only offer structural metrics at isolated points within a monitored asset. Without a viable method of mapping these discrete recordings, SHM research has been severely restricted to measuring structural response at these remote points—an inherited limitation of the implemented sensors that directly diminishes the viability of these systems. Comprehensive reviews of sensor technologies for SHM have concluded that future research must significantly improve the accuracy and resolution of current SHM systems, continuing to suggest a present need to integrate SHM analysis within the geometric context of 3D models [406]. Research investigating smart sensors for SHM tasks relating to load identification [407, 408], damage detection [409], strain analysis [410], and the monitoring of crack propagation [411, 412] have furthermore reflected the fact that the value of modern solutions is directly dependant on the resolution of available data. The challenge of missing measurements is compounded for research investigating damage detection, as structural deterioration is typically a localised phenomenon. Researchers have conventionally attempted to overcome this by optimising a bespoke sensor layout for each given structure, positioning sensors proximate to estimated critical points of failure [413-417]. This solution may be well suited to the isolated setting of a laboratory experiment or simulation, in which the loading conditions, environmental factors, and failure cases are easy to control; however, serviceable structures deteriorate unpredictably over their design life. By optimising the sensor layout based on structural criticality estimated '*a priori*' and focusing on low-dimensional output analysis, these SHM methods limit their accuracy to a narrow range of failure cases and applications.

This pervasive reliance on low-dimensional analysis is compounded by the discrete nature of popular technologies such as Fibre Bragg grating (FBG) sensors, which require individual engravings manufactured along the fibre to be carefully positioned at critical locations along the monitored structure. The usefulness of modern FBG monitoring systems is highly dependent on carefully designed sensor layouts, which can be compromised if any sensors are damaged during installation. Additionally, these discrete readings provide a low-dimensional representation of structural behaviour which is less intuitive relative to continuous sensor technologies, limiting the effectiveness of FBG monitoring strategies in representing the condition of a monitored asset throughout the entire structure. Researchers will often attempt to reconcile these limitations by interpolating readings or by simulating time-consuming finite element models. To the authors' knowledge, a computationally efficient end-to-end method of predicting the response of a monitored asset across its entire

structure based on real-time sensor data has not been proposed prior to the hypernetwork model introduced within this chapter.

This thesis posits that modern SHM research faces two critical challenges that appear to be largely ignored and are rarely reconciled together; the conversion of tremendous quantities of recorded discrete sensor data to high-resolution structural analysis, and an architecture that enables rapid inference speeds for real-time asset monitoring. In recent years, there has been an interest in addressing the latter challenge, driven by increased demand for real-time structural monitoring strategies and the reducing cost of precise sensor technologies [418-420]. However, despite the rising need for innovative monitoring strategies, the academic and applied fields of real-time structural health monitoring have been largely limited by low-resolution forms of structural analysis [421-423]. Existing methods of structural analysis such as Finite Element Analysis (FEA) that can be used to develop similar structural response visualisations are computationally intensive, particularly for complex structures or models that require high-resolution cell-based analysis. This limits the application of finite element methods when investigating the live response of a structure or when visualising changes in structural analysis over long periods. Similarly, previous approaches to develop comprehensive three-dimensional structural analysis from embedded FBG sensors have required complex monitoring pipelines that significantly limit their application for real-time monitoring tasks and time-series visualisations [424, 425]. This presents a significant gap within the available literature, demanding a monitoring solution that leverages modern discrete sensor technologies to generate fast, high-resolution structural analysis.

The hypernetwork architecture developed within this research project is specifically designed to address modern challenges that have been largely overlooked throughout the available literature—namely, mapping low-dimensional strain recordings to high-dimensional, geometrically informed strain heatmaps with compelling inference speed. Hypernetworks were introduced as a form of abstraction for neural network weight configuration, creating a higher-level representation that can generalise weight prediction based on selected inputs [109]. The hypernetwork method is typically characterised by the distinct interaction between two networks, the first of which is tasked with predicting a tensor of weights. These parameters are then used to initialise the second network, providing significant parameter efficiency, improved generalisation capacity, and abstraction for learning implicit functions. In recent years, hypernetworks have demonstrated competitive performance across a diverse range of challenging deep learning tasks, such as three-dimensional shape representation [119], hyperparameter optimisation [262], neural architecture search [426], and natural language processing [427]. Hypernetworks have gained notable traction within the field of 3D reconstruction due to their unique advantages over established voxel-based methods, which suffer from cubic memory resolution requirements that typically limit voxel models to grid sizes of 32^3 or 64^3 [122]. These resolutions are particularly insufficient within the context of FBG asset monitoring, which requires the analysis of strain values for large structures that can potentially span long distances. Within the context of 3D shape reconstruction,

hypernetworks typically utilise signed distance functions (SDF) to represent the underlying target geometry, overcoming the significant scaling memory limitations of voxel-based methods [428]. This research project adopts fundamental concepts from the field of 3D reconstruction, replacing traditional signed distance functions with strain-based implicit neural fields.

6.3.2.1 Research Significance

This chapter introduces a novel method of applying hypernetwork models within the context of structural health monitoring to predict the strain response of a monitored asset at any point within the structure. The inference performance of the proposed hypernetwork model marks a significant improvement in the speed and representation capacity of model structural health monitoring systems, providing a live predictive visualisation of the entire structure's response to load. The strain-based implicit neural fields introduced within this thesis present a new method of predicting structural behaviour, allowing the strain response at any point within the structure to be predicted. The proposed hypernetwork can be trained to approximate structures of any topological form, owing to its ability to generalise across a diverse range of functions. Furthermore, while a cell-based grid of 100,000 values was selected to render the predicted heatmaps, the proposed stain representation has theoretically unlimited resolution. Under the implemented inference workflow, the proposed hypernetwork generates 100,000 strain cell values every 0.0036 seconds—faster than the set sampling frequency of the interrogator unit used to process the signal from the embedded FBG sensors. In this sense, the proposed hypernetwork model is capable of predicting high-resolution strain heatmaps before the interrogator finishes processing new strain values, effectively providing a real-time predictive monitoring strategy.

This research marks the first application of hypernetworks within the field of structural health monitoring, establishing a precedence for future research to investigate hypernetworks for alternative SHM tasks. The low memory footprint, rapid inference, and prediction resolution of the proposed hypernetwork model enables its inference application for a broad range of additional SHM applications, which typically depend on remote edge devices with limited computational resources. These potential future applications include the prediction of displacement, crack propagation, vibrations, rates of corrosive agents, or stress distributions from applied loads. The value of the proposed hypernetwork monitoring may be realised through its ability to reduce operational downtime, inform optimised maintenance schedules, empower asset managers, and reduce the critical response time for sudden structural failures. Collectively, these features improve the structural integrity, serviceability, and utilisation over the lifetime of a monitored asset, furthering the application of embedded sensor technologies by overcoming critical limitations faced within the modern era of structural health monitoring.

6.3.3. Methods and Techniques

6.3.3.1 FBG Selection

Discrete Fibre Bragg grating (FBG) sensors have become an integral component of many modern structural health monitoring systems, installed across a diverse range of assets including bridges [429], roads [430], buildings [431], geotechnical structures [432], and railway infrastructure [433]. Within the context of structural health monitoring, Fibre Bragg grating sensors have several established advantages over conventional sensors such as electrical strain gauges and piezoelectric sensors. FBG sensors are lightweight, resistant to corrosion and electromagnetic interference, robust for extreme environments, and highly accurate for measuring the strain response at discrete locations [434]. The relatively non-intrusive size of FBG sensors furthermore allows them to be embedded within composite materials during construction, providing critical internal monitoring capabilities without compromising structural integrity. FBG sensors have also demonstrated advantages relative to alternative fibre optic sensors such as distributed strain sensors. The multiplexing capabilities of FBG's allow for an array of sensors to be manufactured across a single length of optical fibre without requiring individual wiring [435]. These arrays can span large distances without experiencing significant signal loss—a critical feature that enables the monitoring of assets with long spans such as bridges, roads, and railways.

Fibre Bragg grating sensors work by reflecting a specific wavelength of light that is transmitted through the fibre cable, known as the Bragg wavelength (λ_b). The specific spectrum of light that is reflected is determined by the period (Λ) of the manufactured grating and the effective refractive index of the fibre itself (n_{eff}). When mechanical deformations or thermal variations are imposed upon the FBG, the period of the grating is slightly altered, causing a change in the Bragg wavelength. Mechanical or thermal factors directly impact the strain recorded at discrete FBG locations by shifting the specific wavelength of light that the grating reflects. The Bragg wavelength of reflected light is governed by the following equation:

$$\lambda_b = 2\, n_{\text{eff}}\, \Lambda \tag{6.14}$$

6.3.3.2 Sensor Installation

The post-tensioned footbridge investigated within this research consists of a 17.9 meter long and 2.6 meter wide reinforced concrete deck superstructure at a 1:20 grade. The bridge itself may be simplified as a beam structure, connected by a movement joint to an adjacent walkway and a construction joint to a mezzanine slab. There is a pin support provided by a rectangular concrete column underneath the deck near the midspan, in addition to the fixed-end support provided by the landing slab. Photo documentation of the installation of Fibre Bragg grating sensors and the monitored footbridge structure are provided in Figure 6.13 and Figure 6.14, respectively. The datasets used to train the proposed hypernetwork model were recorded from 64 Fibre Bragg grating sensors that were attached to longitudinal rebar within the footbridge prior to the pouring of the concrete. Four fibre optic arrays split into 'A' and 'B' groupings were installed along the top and bottom longitudinal rebar of the deck,

each containing 16 FBG sensors positioned at 1-meter increments. The top and bottom arrays of each group were positioned 164 mm apart vertically, with 58 mm of cover between the top array and the surface of the concrete deck, connecting to a Luna Hyperion si255 interrogator system measuring wavelengths to a precision of 1 pm.

Figure 6.13 – Installation of Fibre Bragg grating sensors during construction

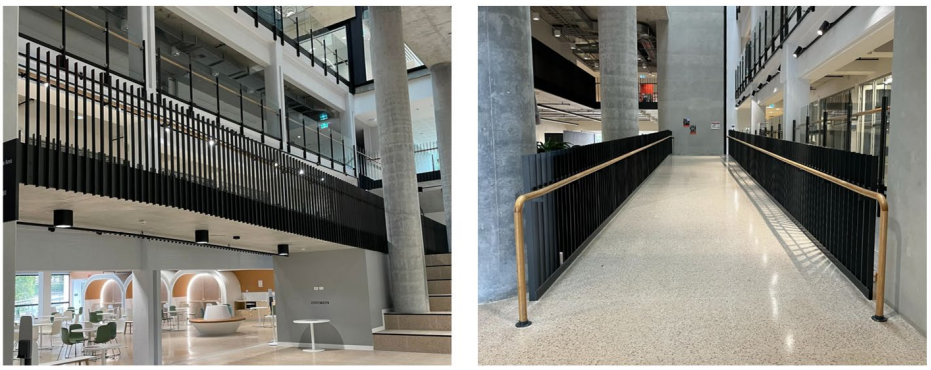


Figure 6.14 – Monitored footbridge structure

6.3.3.3 Signal Preprocessing

To account for environmental changes that the bridge is subject to and capture the expected strain response of the structure in its unloaded state, baseline strain values were generated using a Savgol-Golay filter. This baseline datasets provides critical context from which the impact of loading events can be observed. The Savgol-Golay convolution increases the precision of the strain readings by fitting successive subsets of strain data using a window of length $N = 2m + 1$, where m is the number of points on either side of the windows center [436,

437]. Within each window, a polynomial of degree n is fitted to the subset using the least-squares method, minimising the squared error between actual strain readings and the polynomial values. For each subset, the cost function of a given polynomial to be minimised is calculated as:

$$E = \sum_{i=-m}^{m}\left(\sum_{k=0}^{n} a_k i^k - \epsilon_i\right)^2 \tag{6.15}$$

where ϵ_i is taken as the original strain recording and a_k is the kth polynomial coefficient. The output of the Savgol filter for a given subset is generated by evaluating the centre point of each polynomial curve. The following point is output by shifting the subset window by a single sample and repeating the polynomial operation. The final output of this process represents the baseline strain dataset, which aims to reflect the long-term changes to the strain profile of the unloaded structure, filtering out signal noise and local loading events. This output can therefore be expressed in terms of the filter coefficients (w) as follows:

$$y(k) = \sum_{i=-m}^{m} w_{k-i}\epsilon_i \tag{6.16}$$

The signal used to track mechanical effects is referred to as the 'delta strain' within the context of this research, which is calculated by subtracting the baseline from the original sensor value for a given timestamp. Notably, FBG sensor readings are subject to various forms of signal noise, including temperature fluctuations that influence the fibre's refractive index, interference from other distrusted FBG sensors along the fibre, imperfections within the fibre itself impacting the signal intensity, and structural vibrations. As the delta strain data was calculated by subtracting the smoothed baseline from noisy strain recordings, an additional preprocessing step was required to further improve the precision of the training data. To mitigate the influence of signal noise on the predictive capacity of the hypernetwork model, the delta strain dataset was pre-processed using an adaptive exponential moving average (EMA). This step utilises a dynamic smoothing coefficient (α) to modify the degree of smoothing based on the variation in consecutive delta strain values (ε), while accounting for a threshold of signal noise ($\gamma = 4\mu\varepsilon$) beyond which external mechanical loading is expected. The degree of smoothing provided by the adaptive exponential moving average is also regulated by thresholds imposed to the maximum (α_{max}) and minimum (α_{max}) smoothing coefficient, taken as 0.01 and 0.09 for this implementation. These values were chosen to reduce the impact of random signal noise, while allowing the influence of mechanical loading to be adequately captured during the experiment. The smoothing coefficient for a given timestamp (i) is calculated as follows:

$$\alpha_i \begin{cases} \alpha_{max}, & \text{if } |\varepsilon_i - \varepsilon_{i-1}| > \gamma \\ \alpha_{min} + (\alpha_{max} - \alpha_{min}) \cdot \left(\frac{|\varepsilon_i - \varepsilon_{i-1}|}{\gamma}\right), & \text{if } |\varepsilon_i - \varepsilon_{i-1}| < \gamma \end{cases} \tag{6.17}$$

The exponential moving average of the delta strain signal (y) at the required timestamp is then updated using the previously calculated smoothing coefficient:

$$y_i = \alpha_i x_i + (1 - \alpha_i)\, y_{i-1} \tag{6.18}$$

The implementation of the Savgol-Golay filter and the Adaptive Exponential Moving Average smoothing function are visually illustrated in Figure 6.15, generating the smoothed baseline and delta strain values, respectively. As the baseline dataset aims to capture the unloaded state of the structure, seqeuntial low-order polynomials are used to represent the long-term change in the strain profile over the duration of the experiment while smoothing over localised effects. In contrast, the EMA delta strain reflects the strain behaviour under load. This is achieved by using a threshold value γ to adapt the degree of smoothing and ensure that sudden shifts in the strain data that are caused by mechanical loading are captured within the training dataset.

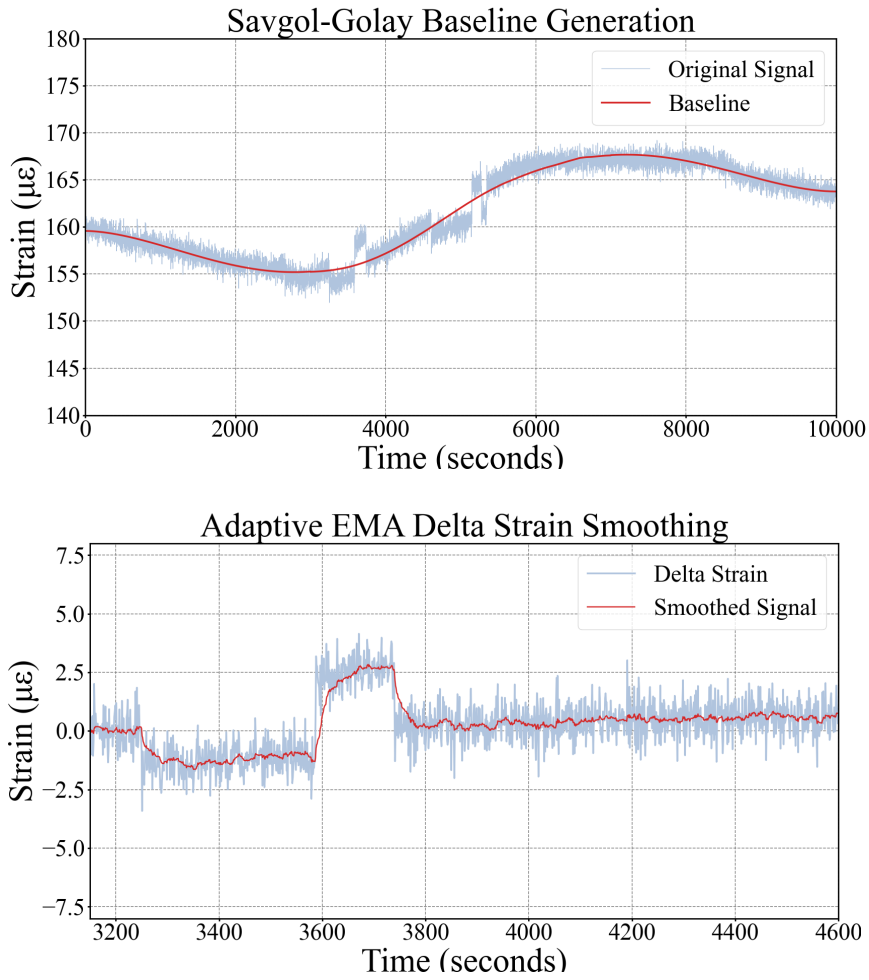


Figure 6.15 (a) – Savgol-Golay filter for baseline generation, (b) – Adaptive EMA function for delta strain smoothing

6.3.4. Proposed hypernetworks

The hypernetworks developed within this research are trained using novel function-based three-dimensional strain heatmaps. These cell-based strain values have been implemented to replace the signed distance functions that traditional hypernetworks are tasked with learning for 3D reconstruction tasks. The proposed heatmap prediction model pipeline consists of a hypernetwork (f) and a primary network (g). Readings from the embedded Fibre Bragg grating sensors are first processed using a Savgol-Golay filter and the corresponding

baseline is subtracted to calculate the delta strain. These values are passed through an Adaptive Exponential Moving Average smoothing function, before being used as input to the hypernetwork model. The hypernetwork takes a tensor of 64 delta strain readings as its input, followed by four fully connected layers each containing 1,024 neurons with a ReLU activation function. The fifth layer of the hypernetwork contains a hyperbolic tangent activation function and outputs a tensor of 3,329 weights, each mapping to a parameter within the primary network. The four fully connected layers of the primary network are then parametrised using these predicted weights. The first four layers of the primary network each use a ReLU activation function and contain 32 neurons. The primary network takes a batch of 3-dimensional coordinates as its input and predicts a single strain value at each of these points within the structure. By establishing a cell-based representation of the structure and passing these coordinates as inputs to the initialised primary network, the prediction strain heatmap is generated from discrete FBG sensor values, illustrated in Figure 6.16. The mapping from discrete sensor values to a cell-based heatmap of the structure's strain response is further demonstrated in Figure 6.17.

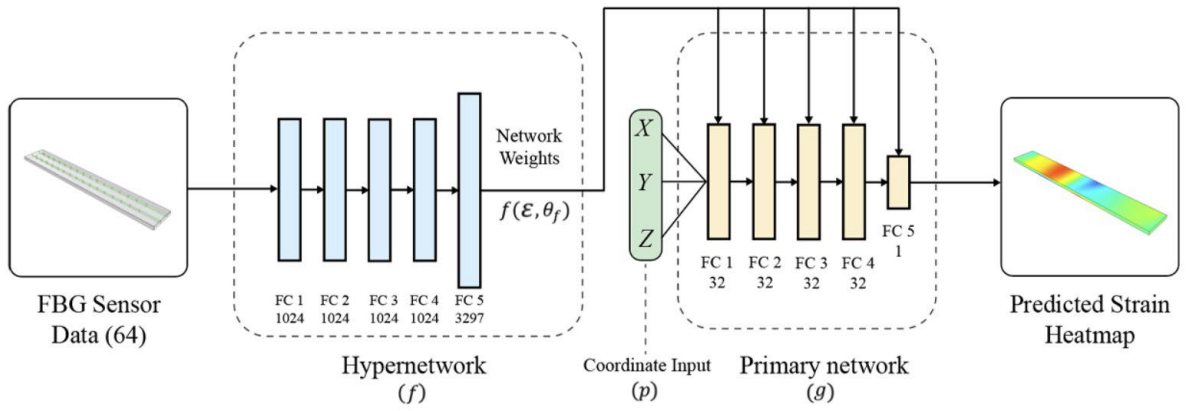


Figure 6.16 – Proposed hypernetwork architecture for strain heatmap generation

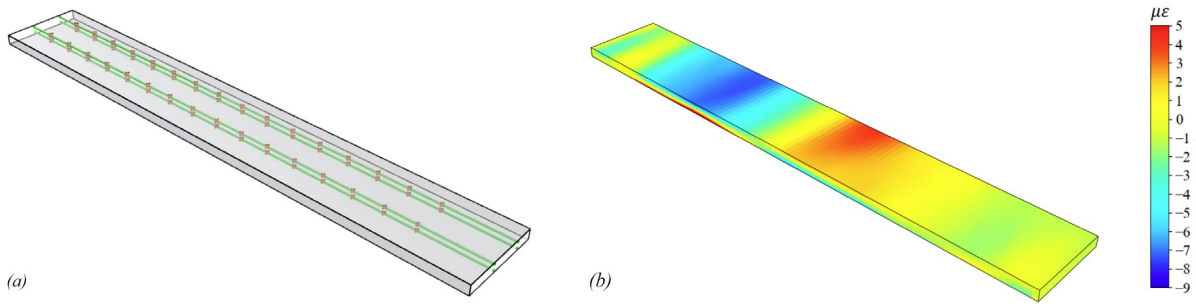


Figure 6.17 (a) – Embedded Fibre Bragg grating sensor arrays, 6.17 (b) – Hypernetwork predicted strain response of footbridge deck during loading

The loss for the proposed strain function is calculated as the mean absolute error between the predicted strain and the delta strain dataset, calculated as follows:

$$H(\theta_f, \varepsilon) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - g\left(p, f(\varepsilon, \theta_f)\right)\right| \tag{6.19}$$

The trained hypernetwork takes 64 delta strain values as its input, before outputting a tensor of predicted weights. The primary network is then initialised using these weights before it is being passed a series of normalised three-dimensional points. Each of the coordinates provided as inputs to the primary network represent a cell in the final heatmap. The corresponding output from the primary network represents the predicted strain value at each of these coordinates, which are evaluated against the ground truth delta strain for each timestamp. The hypernetwork model outlined within this chapter used the Adam optimiser with a learning rate of 2×10^{-5} and a batch size of $n = 10$ heatmaps per step, trained for 25,000 epochs over 5 days. All models trained and evaluated within this research were afforded the same computational budget, using an intel(R) Core(TM) i7-11700 with an NVIDIA GeForce RTX 4090.

6.3.4.1 Dataset Generation

The training, testing, and validation datasets used to train and evaluate the proposed hypernetwork model contain fixed sets of three-dimensional coordinates positioned throughout the structure of the footbridge and the estimated strain at these points. The width and span of the structure were divided into 100 cells along each axis, while the depth of the deck was divided into 10 cell-based layers. Bayesian inference was then utilised to generate 100,000 corresponding strain values for each of the 10,000 seconds of recorded data, totalling 1×10^9 cell values. For each timestamp, a new prior distribution over possible functions was generated and updated through Gaussian Process using all 64 post-processed delta strain values. This in turn allowed for the estimation of the strain value at each of the 100,000 heatmap cells. These complete heatmaps were then collated and divided into the respective training, testing, and validation datasets used to develop and evaluate the implemented hypernetwork model.

Bayesian inference with Gaussian Process suffers from several limitations that hinder its application within the context of live structural health monitoring. Namely, conventional Bayesian methods suffer from the "curse of dimensionality" and tend to lose efficiency for high dimensional tasks [438-440]. Additionally, the computational complexity of the method inhibits its application for tasks requiring rapid inference speeds, such as live structural health monitoring or the visualisation of structural response over an extended loading event. For these reasons, Bayesian inference has been exclusively used within the context of this research to develop the training, validation, and testing strain datasets necessary to develop and analyse the proposed hypernetwork model.

A Gaussian Process (GP) is a statistical method that assumes the underlying data can be modelled as a multivariate Gaussian distribution. That is, for any collection of input points $X = \{x_2, x_2, \dots, x_n\}$, the corresponding outputs $f(X) = \{f(x_1), f(x_2), \dots, f(x_n)\}$ are jointly Gaussian [441]. The Gaussian Process defines a prior distribution over possible functions, which is updated to a posterior distribution based on historical observations. The underlying unknown function that the GP aims to model within the context of this research project is the strain response of the footbridge structure at any point, given the 64 post-processed delta strain readings from the discrete FBG sensors. The delta strain values at a given timestamp are 'observations' used to condition the GP, which in turn models a function that approximates the strain response throughout the structure. An example Bayesian model is visualised in Figure 6.18, in which a single array of 16 discrete embedded FBG sensors was used to condition a posterior distribution through a Gaussian Process. The mean of this posterior distribution is then calculated, providing the interpolated strain across the entire span of the fibre optic cable. These strain profiles have been predicted for every second of a dynamic loading event and combined as a heatmap within the figure.

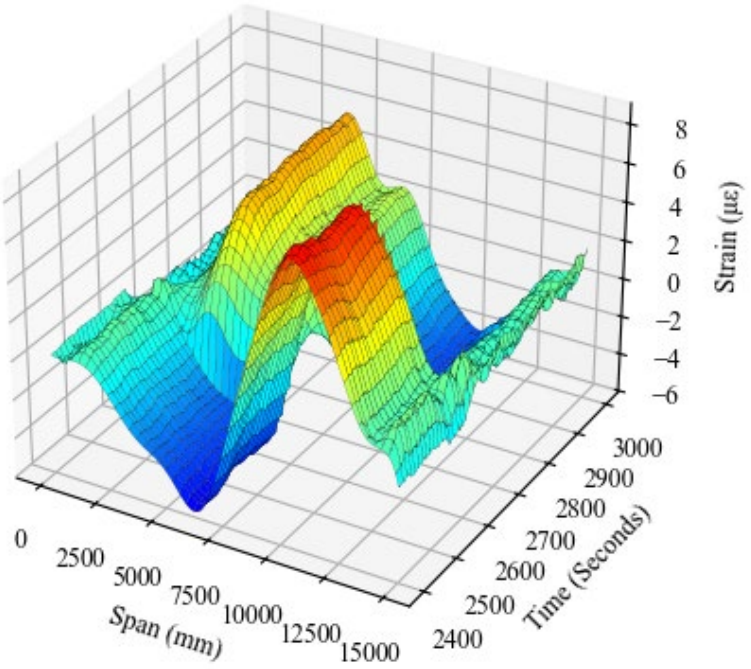


Figure 6.18 – Example Bayesian model of discrete fibre optic sensor array

Notably, this example maps strain readings from 16 one-dimensional points to a probability-based Bayesian model of the strain response along a single axis—the span of a single embedded fibre optic cable. The complete database developed within this research extends upon this example using delta strain readings from 64 embedded sensors, fitting a Bayesian model to estimate the strain at any three-dimensional point within the structure. The covariance function k, or 'kernel', represents the similarity between function values at different input points, and assumes that proximate points have similar outputs. The assumed smoothness of the target function is satisfied by the discrete delta strain dataset, as the distribution of strain along the structure is uniform

under elastic conditions. For this application, the Radial-Bias function was chosen as the kernel [442], defined as follows:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{\|x-x'\|^2}{2\lambda^2}\right) \tag{6.20}$$

where σ_f^2 represents the signal variance and λ is the length scale hyperparameter that controls the decay of the RBF kernel. The variance and length scale are scaled between the bounds [0.001,100] and [0.1,100] respectively during optimisation. The Radial-Bias kernel determines the dependence between the function values at the input points x and x'.

6.3.5. Experiment

6.3.5.1 Strain Response

The implemented strain dataset is an extension of a loading experiment carried out nine months after the embedding of the FBG sensors, which explored the viability of using neural networks to map discrete sensor data for high-resolution structural analysis. While the authors' previous work [443] represents a significant improvement in the representation capacity of modern SHM systems, the complex computational pipeline and reliance on non-uniform rational B-splines modelling (NURBS) significantly inhibits its potential application for real-time structural monitoring applications. The proposed hypernetworks offer a radically novel opportunity to bridge this gap between cutting edge machine learning models and the unaddressed challenges facing applied real-time SHM systems. The raw dataset consists of 640,000 individual sensor readings that were recorded under a live loading test, during which two operators guided a loaded pallet jack with a combined mass of approximately 1.1 tonnes over the span of the footbridge structure. The load was moved at one-minute intervals over each FBG sensor. It is assumed that the 64 sensors spaced at 1-meter increments and split over 4 embedded arrays captured an adequately representative discrete snapshot of the footbridge's structural response under load. The relative coordinates of these embedded sensors and their strain values were then used to condition a Gaussian Process for each timestamp. These were sampled to generate 10,000 heatmaps of the footbridge, each consisting of 100,000 individual cells. Following this probabilistic interpolation, these heatmaps were then divided across the training (80%), validation (10%), and testing (10%) datasets used to train and evaluate the proposed hypernetwork model.

6.3.5.2 Competing models

Within the field of 3D shape reconstruction, hypernetworks have been established as a leading deep learning architecture, with competitive benchmarked performance against competing models. One common alternative 3D reconstruction technique requires each three-dimensional point to be appended to the inputs of a single deep neural network, referred to as "conditioning by concatenation" [230]. To further evaluate the relative performance of the proposed hypernetwork method, two variants of this foundational approach have also been

trained and evaluated for the novel task of strain heatmap generation. Alternative fully connected multi-layer perceptron (MLP) models have been applied within the context of this project to predict the strain values at each cell of the footbridge structure, using the concatenated input of the three-dimensional cell coordinates and the 64 postprocessed FBG sensor readings at each timestamp. Referred to as 'DeepStrain' models, these benchmark MLP networks were developed as variations of the established DeepSDF approach for reconstructing 3D geometries [231]. The smallest of the two alternative benchmarks (DeepStrain_{1024}) consists of four fully connected layers, each containing 1024 neurons and ReLU activation functions, outputting a single predicted strain value at the cell level. This architecture was chosen to align with the parameters of the base hypernetwork for fair comparative analysis. The largest MLP variation (DeepStrain_{2048}) uses a similar foundational MLP layer architecture, implemented with 2048 neurons per fully connected layer.

6.3.6. Results and Discussion

The performance of the hypernetwork model for predicting cell-based strain heatmaps from embedded FBG sensors has been evaluated across training, testing, and validation datasets, with examples of the hypernetwork predicted responses provided in Figure 6.19. These neural field representations of the structure's strain response are mapped from 64 discrete sensor values and have theoretically infinite resolution capacity, illustrating the 3D visualisation capabilities of the proposed hypernetwork model. The final model can therefore be sampled at any desired resolution, while also allowing isolated regions of the structure to be analysed independently if required. The performance metrics of the proposed hypernetwork for generating these implicit neural structural heatmaps are summarised in Table 6.7, including the results from the benchmark DeepStrain models.

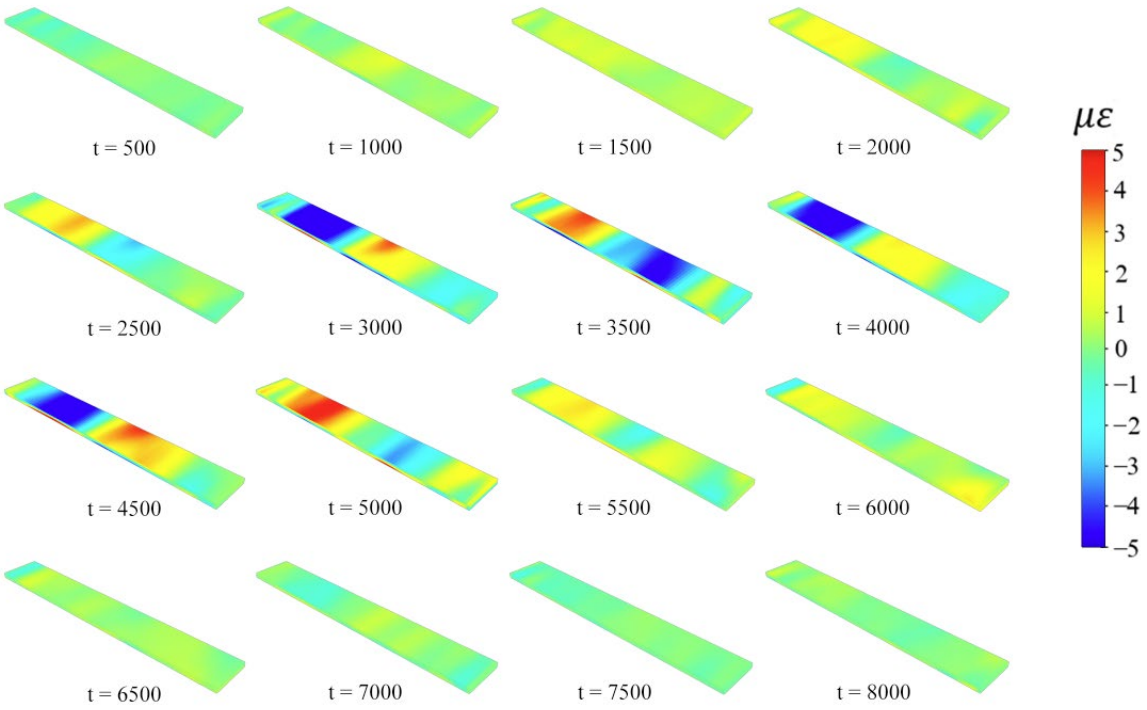


Figure 6.19 – Hypernetwork predicted strain response during the loading event (t-time)

Table 6.7 – Comparative model performance across delta strain datasets

Model	No. Parameters	Dataset	Mean Absolute Error (MAE)	Maximum Absolute Error	Mean Bias Deviation (MBD)	Standard Deviation of Errors	Interquartile Range of Errors (IQR)
Hypernetwork	6,627,585	Training	0.0171	1.6265	-0.0015	0.0286	0.0203
		Testing	0.0285	1.5078	-0.0014	0.0439	0.0403
		Validation	0.0294	1.5636	-0.0009	0.0461	0.0408
DeepStrain_{1024}	3,219,457	Training	0.3408	4.7819	0.0928	0.4832	0.3329
		Testing	0.5430	4.5821	0.2770	0.4722	0.3241
		Validation	0.2428	4.7123	0.1180	0.4889	0.3412
DeepStrain_{2048}	12,730,369	Training	0.3487	4.7277	-0.0712	0.5016	0.3595
		Testing	0.5288	4.6991	0.0078	0.4945	0.3546
		Validation	0.2439	4.7233	-0.0520	0.5071	0.3674

The Mean Absolute Error (MAE) was used as the loss function for the hypernetwork during training, borrowing from established hypernetwork conventions within the context of 3D shape reconstruction by signed distance function. Within the proposed implementation, the MAE provides the average magnitude of hypernetwork residuals when predicting individual strain cell values, rather than conventional signed distance values. Additional performance metrics have also been calculated across these datasets, as strain cells at unloaded states can disproportionately impact mean-based evaluation metrics. The training, testing, and validation mean absolute error loss was 0.0171 $\mu\varepsilon$, 0.0285 $\mu\varepsilon$, and 0.0294 $\mu\varepsilon$, respectively, demonstrating the representation capacity of the hypernetwork method for predicting the generalised strain response of the monitored structure. The maximum absolute error recorded for any cell across the training, testing, and validation datasets was 1.6265 $\mu\varepsilon$, 1.5078 $\mu\varepsilon$, and 1.5636 $\mu\varepsilon$, respectively, which is proximate to the anticipated threshold of unprocessed FBG signal noise for the installed sensors ($\pm$1με). Notably, the number of parameters required by the proposed hypernetwork model is only 52% of the total parameter count used by the largest DeepStrain model benchmarked within this research. Despite this, the hypernetwork model outperformed DeepStrain architectures across all implemented evaluation metrics. The Mean Absolute Error of the hypernetwork model was 67.91% smaller than DeepStrain_{2048} across the testing dataset and 66.90% smaller when benchmarked using the validation dataset. Furthermore, the maximum recorded absolute strain error from predictions generated by the DeepStrain_{2048} model was 312% larger than the proposed hypernetwork model for the testing dataset and 302% larger when applied to the validation dataset. The stagnation in model performance across the DeepStrain architectures suggests that conditioning by concatenation is significantly less effective for strain heatmap generation relative to the implemented hypernetwork framework.

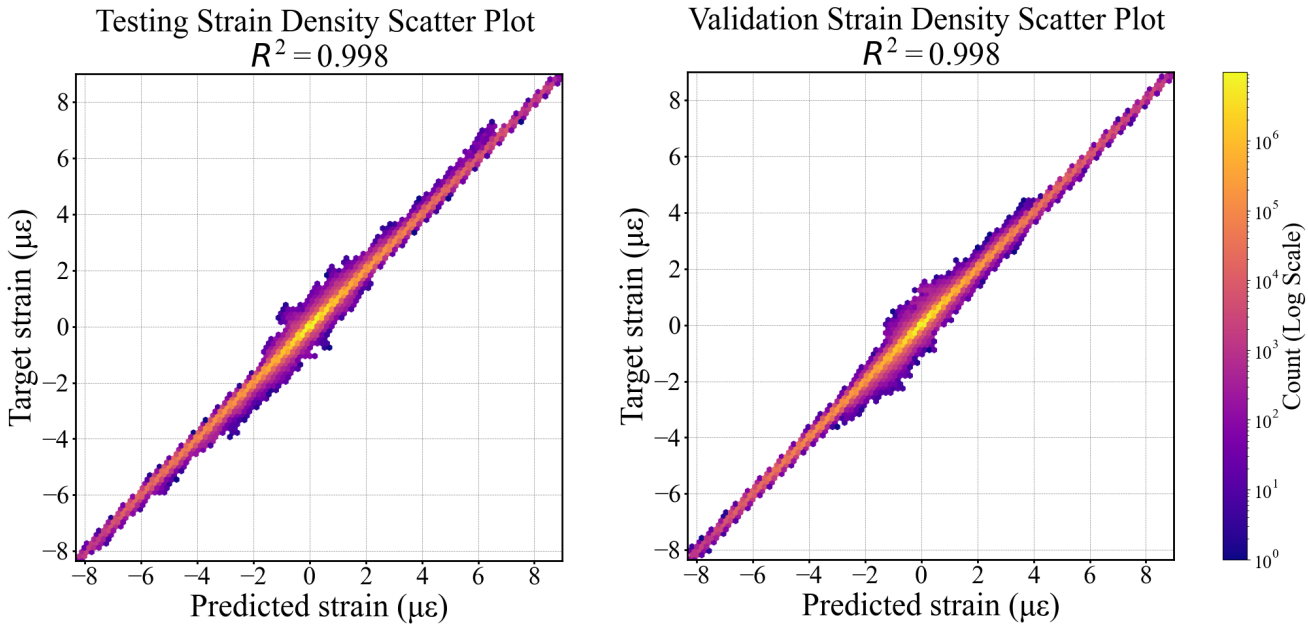


Figure 6.20 (a) – Hexbin density scatter plot of predicted vs target testing strain, 6.20 (b) – Hexbin density scatter plot of predicted vs target validation strain

A density scatter plot of the hypernetwork predictions and targets across the testing and validation dataset is provided in Figure 6.20, visually demonstrating the predictive capacity and fit of the proposed hypernetwork model. The linear relationship between predicted and target strain values for both the training and validation heatmaps indicates a lack of systematic bias and predominantly symmetric residuals proximate to the line of perfect prediction ($y = x$). The coefficient of determination R^2 for these predictions was approximately 0.998 for both the testing and validation datasets, indicating that only 0.2% of the variance observed in the target strain datasets was not explained by the trained model. The standard deviation of these errors across the testing and validation dataset was 0.0439 $\mu\varepsilon$ and 0.0461 $\mu\varepsilon$, respectively, indicating the proximity of the model's predictions to the ground truth target strain values and the stability of the trained hypernetwork to predict consistent and accurate strain heatmaps. The low magnitude of the mean absolute error and the standard deviation of prediction residuals across the testing and validation dataset collectively demonstrate the compelling calibration of the trained hypernetwork in terms of both accuracy and consistency.

To further analyse the symmetry of model prediction errors, the distribution frequency of hypernetwork residuals across the testing and validation strain datasets is provided in Figure 6.21. Bias for under or over prediction will typically produce a skewed distribution of errors, which is indicative of overfitting to the training set. In contrast, the prediction errors of the trained hypernetwork exhibit a symmetric distribution with an

approximately zero mean. This in turn indicates that the hypernetwork has sufficiently captured the strain response of the footbridge across the sampled cell-based datasets. The symmetric distribution of residuals across testing and validation set suggests that the prediction error is predominantly impacted by stochastic noise within the post-processed delta strain data rather than by insufficient underlying input information or by overfitting experienced during training. This is reaffirmed by the calculated Mean Bias Deviation of the hypernetwork provided in Table 6.7, achieving a MBD of -0.0014 $\mu\varepsilon$ and -0.0009 $\mu\varepsilon$ across the testing and validation datasets. While Mean Bias Deviation does not reflect the magnitude of errors in the same way as the Mean Absolute Error loss function, it does provide invaluable insights into the bias of the evaluated model. The near-zero Mean Bias Deviation of the hypernetwork across the testing and validation datasets indicates that there is minimal systematic bias exhibited by the hypernetwork when predicting cell-based strain heatmaps, with a negligible tendency to underestimate strain values.

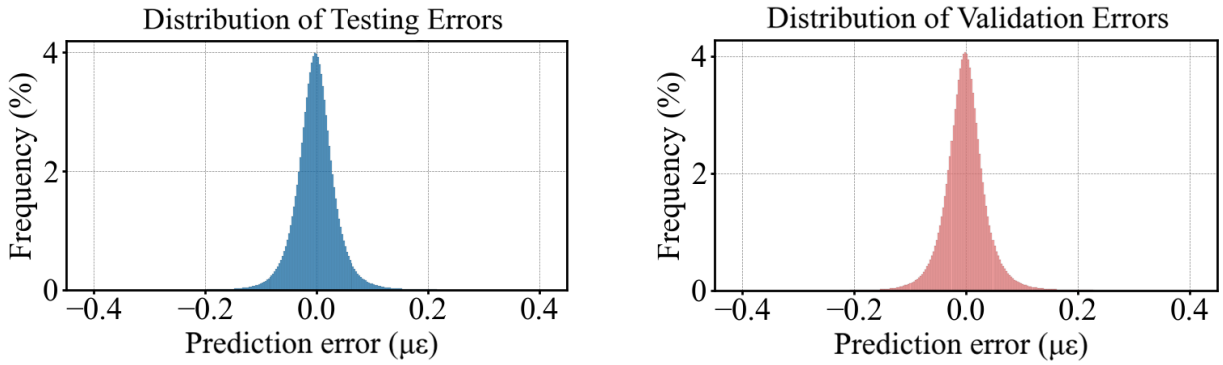


Figure 6.21 (a) – Histogram of prediction errors on testing dataset, 6.21 (b) – Histogram of prediction errors on validation dataset

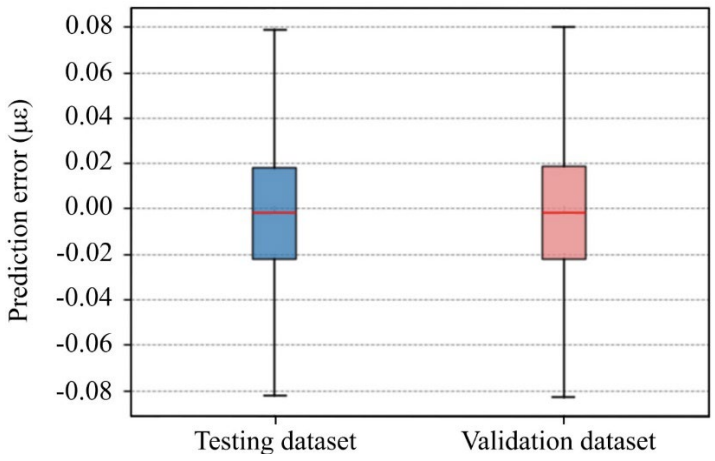


Figure 6.22 – Boxplot of prediction errors on testing and validation datasets

The boxplot of hypernetwork errors for the testing and validation datasets is provided in Figure 6.22, summarising the distribution of model residuals between upper and lower bounds. The Interquartile Range

(IQR) of model errors for the testing and validation heatmaps was 0.0403 $\mu\varepsilon$ and 0.0408 $\mu\varepsilon$ respectively, reflecting the error range for the central 50% of predicted strain residuals. The small magnitude of the IQR for both testing and validation heatmap datasets and the clustering of residuals around a mean error proximate to zero indicates a low variability of errors and a high degree of precision in the hypernetworks prediction of individual strain cells. The consistency between the testing and validation datasets furthermore indicates that the hypernetwork generalises well to unseen strain response states, mapping. Collectively, these results suggest that the trained model exhibits minimal bias and a capacity to generate high-resolution, accurate visualisations of the footbridge's strain response entirely from 64 discrete FBG sensors embedded within the structure.

The hypernetwork model introduced within this chapter exhibited remarkable strain prediction capabilities throughout the footbridge structure, achieving a mean absolute error of 0.0171 $\mu\varepsilon$, 0.0285 $\mu\varepsilon$, and 0.0294 $\mu\varepsilon$ across the training, testing, and validation datasets, respectively. This thesis posits that the competitive performance of hypernetworks for generating manifold geometries from learnt signed distance functions is directly translatable to predicting the structural response of monitored assets. The compelling accuracy of the trained model may be attributed to the established effectiveness of hypernetworks for learning complex functions within the field of 3D reconstruction [269, 336, 444], and the preprocessing noise reduction methods used to improve the precision of the input data. The unprocessed noise recorded from the installed FBG sensors was approximately $\pm 1\mu\varepsilon$, while the maximum recorded individual cell absolute error calculated across all training, testing, and validation heatmaps was 1.6265 $\mu\varepsilon$, 1.5078 $\mu\varepsilon$, and 1.5636 $\mu\varepsilon$, respectively. Furthermore, despite requiring 47.94% fewer parameters when compared against the largest competing DeepStrain architecture, the trained hypernetwork model outperformed both benchmarks across all recorded training, testing, and validation performance metrics. The low memory footprint of the proposed hypernetwork model furthermore enables its inference application across remote devices with limited computational resources, which is a critical feature for real-time processing applications that typically rely on highly efficient, low specification edge devices.

Notably, as the hypernetwork model pipeline is trained to learn a neural field representation of the structure's strain response, the resolution of the output heatmap is entirely customisable and can be altered at inference according to the needs of a given structural monitoring strategy without requiring the model to be retrained. As an implicit neural field representation, the structural form of the generated 3D output can furthermore take on alternative manifold forms beyond the simple rectilinear footbridge implemented within this research, demonstrated by the competitive performance of hypernetworks within the field of 3D geometry reconstruction [232, 445]. This includes the potential application of hypernetworks for dynamic structural simulations which may require the form of the visualised structure to deform under imposed loads, or the simulation of structural damage.

6.3.7. Limitations and Future Research

There are notable limitations to the Bayesian inference method used to generate the necessary training, validation, and testing strain datasets. Namely, Bayesian methods typically suffer from the curse of dimensionality, limiting their application to low-dimensional tasks. This was not a significant limitation within the context of this research as a fixed three-feature set was used to represent the coordinates of each FBG sensor reading. For research tasks that are investigating problems with a larger collection of features, methods such as Deep Bayesian Optimisation may be used to substitute traditional surrogate Gaussian Processes with neural networks [446]. The limitations of Bayesian inference may also be overcome by applying traditional structural engineering approaches methods such as Finite Element Analysis (FEA) to build out these preliminary datasets. As a mesh-based mathematical method, FEA is a well-suited alternative solution to generate hypernetwork training data, as the structural response may be calculated at the cell level throughout the body of the evaluated structure. While inference speed is a primary concern for real-time SHM solutions, the computational inefficiency of finite element analysis is less of a concern when purely used for developing training datasets. FEA may also be appropriate for more complex structures or when limited sensor data is available. For research tasks investigating the use of FEA for hypernetwork dataset generation, the cell resolution of the training dataset should approximate the desired precision of the output prediction heatmap. While the function-based strain generation process introduced within this research project can take on a theoretically infinite resolution, a cell-based grid of 100,000 values was selected to render the predicted heatmap.

Furthermore, the training datasets developed within this research were driven by embedded FBG sensors installed during construction. Future research may investigate the application of the proposed model pipeline for monitoring assets with retrofitted sensors, which is critical for maintaining existing infrastructure that may be deteriorating or at risk of critical failure. While this research project investigated the application of hypernetworks for predicting the strain response of an asset using embedding FBG sensors, future research may also seek to investigate the proposed monitoring strategy using alternative sensor technologies and structural health metrics. The compelling performance of the developed hypernetwork solution implies the potential application of hypernetworks throughout the field of structural health monitoring. These future hypernetwork research scopes may include the use of displacement sensors to predict structural movements [447], corrosion sensors to predict the rates of corrosive agents [448], acoustic emission sensors to predict crack propagation [449], accelerometers to predict structural vibrations [450], or humidity sensors to predict moisture levels in structural materials such as wood or concrete [451].

6.3.8. Summary of key findings from Section 6.3

This research posits that the potential of hypernetworks within the context of structural health monitoring has been unrealised across the available literature—offering a unique framework to predict an asset's entire

structural response with unprecedented inference speed and resolution capacity. The proposed novel hypernetwork model exhibits compelling performance for cell-based strain prediction, outputting strain values with a maximum absolute validation error of 1.5636 $\mu\varepsilon$ and a mean absolute validation error of 0.0294 $\mu\varepsilon$. The trained hypernetwork significantly outperforms competing models that were conditioned by concatenation, despite requiring 47.94% fewer parameters when compared against the largest benchmark model. The low memory footprint of the proposed hypernetwork and its rapid inference speeds are highly suitable for efficient inference on remote, low specification edge devices that are typically used for existing real-time monitoring solutions. This research posits that the competitive prediction capacity of the trained hypernetwork model for generating cell-based strain heatmaps demonstrates significant potential to enrich modern structural health monitoring systems and improve the structural integrity, serviceability, and utilisation of monitored assets over their operational lifespan.

6.4. Conclusion

This chapter aims to address the prominent 'Big data' problem that has emerged within the field of structural health monitoring, mapping vast quantities of discrete strain data to implicit neural representations that can be used to predict structural response at any point within the monitored asset. Prior to this thesis, a comprehensive, memory-efficient, spatially rich, high-resolution machine learning model had not been developed for sensor-driven, real-time predictive structural health monitoring. By more closely aligning the predictive output of the proposed SHM solution with the practical strategies currently utilised by industry practitioners, this thesis aims to further encourage the broader adoption of sensor technologies such as fiber Bragg gratings and reduce existing overdependencies on inefficient and subjective visual inspections. This is reflective of a broader objective within the field of SHM to mobilise the industry from a reactive asset maintenance approach to once that proactively assesses structural health.

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

7.1. Summary

To frame the discussion of this thesis appropriately, a broader question surrounding our built environment must be addressed: by breaking down existing research barriers, can interdisciplinary machine learning solutions and novel algorithms successfully bridge the vast gap between academia and industry? The models trained across each chapter of this thesis aim to answer this fundamental question, while directly building off the contributions of previous chapters. The specific gaps that each chapter has aimed to address varies significantly according to the given asset phase, while continuing a deep interdisciplinary study of metaheuristics and machine learning systems. These may be highlighted as follows:

- For algorithmic design tasks, this thesis has aimed to address the extensive experience currently required from industry practitioners. A complete knowledge of algorithm-specific hyperparameter sets and their corresponding impacts on exploration-exploitation behavior, solution diversity, and convergence should not be necessary for industry practitioners to use these tools effectively.
- Likewise, researchers that are looking to develop high-performing and efficient machine learning models for complex tasks should not be restricted to computationally inefficient methods of hyperparameter tuning and manual neural architecture search.
- Within the context of predictive design, this thesis has sought to find innovative methods to align the academic field of predictive 3D reconstruction with the complexity of real-world structural designs, which may contain hundreds or even thousands of interconnected elements.
- Finally, this thesis has aimed to increase the dimensionality, resolution, and predictive accuracy of ML-based SHM systems. While applied asset monitoring has historically relied on antiquated forms of non-destructive visual inspection, academia has diverted away from the 3D world — finding itself tangled in vast arrays of ‘Big data’ and 2D time series graphs that fail to represent the full spectrum of structural response.

Inherent to each of the distinct research scopes investigated within this thesis is the assertion that computational methods at the intersection of metaheuristics and machine learning offer profoundly unique solutions that have the capacity to reshape our approach to the built environment. With practical applications ranging from algorithmic structural design to real-time predictive structural health monitoring, the models proposed throughout this thesis demonstrate compelling performance, and in some cases have enabled new capabilities for future work across research and industry.

7.2. Conclusions

7.2.1. Design phase contributions

By introducing neural networks within the optimisation loop of a differential evolution algorithm, this thesis demonstrates the compelling capacity for deep learning models to mediate the algorithmic behavior of metaheuristics and reduce performance dependency on static, user-defined hyperparameters. This significantly reduces the degree to which algorithmic performance is contingent on user experience, allowing industry practitioners to deploy metaheuristics for real-world design tasks without requiring extensive knowledge of hyperparameters and their impact on algorithmic performance. The implications of these research results extend broadly beyond the mutation operators of the differential evolution algorithm, with potential future applications including alternative neural-adaptive metaheuristic parameters such as population size, generation count, elitism rate, and candidate selection controls.

The success of this implementation inspired the development of the MEMODE algorithm, which demonstrated the capacity of the differential evolution algorithm to optimise high-performing and memory efficient multimodal models. Despite the architectural complexity required to reconcile multiple streams of input and the memory limitations placed upon the optimisation process, the proposed algorithm effectively evolved the NLP and image-procession branches of the model and tuned the hyperparameter sets to outperform all 1,600 competing models. These included models that were optimised using several alternative approaches, including the genetic algorithm, particle swarm optimisation, and Bayesian optimisation. Additionally, each of the implemented metaheuristic approaches outperformed competing ResNet and Inception variations according to mean F1 score, indicating the viability of these algorithms for memory efficient model design.

The performance of the MEMODE algorithm further justifies the potential application of differential evolution and the broader field of metaheuristics for the automated design and tuning of multimodal deep learning systems, even while under strict computational constraints. These efficient multimodal models have enormous potential throughout the asset lifecycle, with the ability to simultaneously process engineering drawings alongside written design briefs, site construction photos alongside progress plans, and photos of asset deterioration alongside written inspection reports. Despite these opportunities, the typical memory requirements of these models have significantly limited their application across modern consumer-grade devices. The MEMODE algorithm provides a framework to further develop these models with a direct appreciation for the realistic customer-side hardware limitations at inference, with the scale of the final network directly mediated through the user-defined memory penalty function.

The development of machine learning datasets using metaheuristic algorithms provides a radically unique and effective supervised training framework that is particularly well suited to address the pervasive lack of structural design data available to researchers. Widely used academic object-based geometry datasets such as ShapeNet and PartNet are not adequately indicative of the topological forms or scene complexity that practicing designers and engineers require. The symbiotic relationships between machine learning and metaheuristics effectively allows the traditional limitations of deep learning systems — complex architectural design, inefficient manual hyperparameters tuning, and inherent data-dependencies — to be compensated for in a highly targeted manner. One of the most significant challenges that modern deep learning researchers face when applying cutting-edge models within applied contexts such as structural engineering and architectural design is the significant lack of open-source data publicly available for training. This thesis explores novel solutions to this problem by introducing novel strategies for algorithmic dataset generation, employing a parametric rules-based heuristic approach. Metaheuristics offer a structured methodology for the collection and labelling of highly specialised design datasets, overcoming a significant 'data-drought' that has inhibited the application of modern machine learning models within the context of real-world predictive design. As a computationally efficient optimisation strategy, metaheuristics offer a flexible and scalable approach to dataset generation that may be adapted to the resource limitations or dataset size requirements for a given research project. As a result, new algorithmic approaches to structural dataset generation may come to further bridge the gap between compelling academic deep learning models and applied industry design tasks.

The structural datasets generated throughout this thesis include 2D and 3D trusses, gridshells, girder bridges, tied arch bridges, high-rises, and steel frame structures. These algorithmically generated structures have been used to train novel hypernetwork models for highly specialised 3D reconstruction tasks. The capacity of these models to represent complex structural systems has been further scaled through the introduction of transformer predicted translation vectors, which replicate and manipulate predicted base structural elements throughout the design space. Importantly, the design complexity achieved through the proposed element-wise manipulation method reflects a degree of complexity that may be expected within an applied context of iterative design modelling — bridging the vast gap in representation output that exists between academic predictive models and real-world design tasks. This novel solution may also be applied to other foundational models that are commonly used within the field of predictive 3D reconstruction, such as voxels, sphere-based deformation, and point cloud models.

7.2.2. Maintenance phase contributions

Beyond these key contributions within the design phase of the asset lifecycle, this thesis has also developed an entirely novel hypernetwork approach to sensor-based structural health monitoring. Previous research applications within the field of smart sensor structural health monitoring have been severely limited to low-dimensional forms of analysis, which vastly underrepresent the broader complexity of the structural datasets

that these sensors collect. This limitation is compounded by the vast scale of data that these sensors collect, which researchers and asset managers have historically struggled to extract value from. This thesis addresses this 'Big Data' problem through the introduction of strain-based implicit neural representations and a novel hypernetwork architecture. The comprehensive structural response heatmaps predicted by these models provide theoretically unlimited resolution capacity and rapid inference, with the capacity to be adapted according to the computational requirements or limitations of a given project. The architecture of the proposed hypernetworks is also extremely efficient relative to its predictive capabilities. This computational efficiency is critical for remote monitoring applications which may have limited computational resources, further improving the cost alignment between academic predictive solutions and the realistic requirements of asset managers.

The proposed hypernetwork architecture has the capacity to predict the structural response at any point within a monitored asset, overcoming the severe limitations of discrete sensor monitoring systems for which performance is highly contingent on the effective optimisation of sensor layouts. These carefully designed sensor layouts can become severely undermined if any sensors are damaged during installation. The failure of FBG sensors due to mechanical damage occurring during installation is a particularly challenging factor that risks inhibiting the full utilisation of modern SHM systems. These issues are compounded by a distinct lack of high-resolution predictive modelling throughout the academic field of structural health monitoring, which has become increasingly dependent these highly optimised discrete sensor layouts and low-dimensional forms of analysis. Within this context, a failure of any sensor may catastrophically undermine the viability of these existing monitoring solutions, as these systems provide inherently limited contingencies for sensor failure. For embedded sensor applications, these failures are irreversible and, in some sense, inevitable at an industry scale. The risk of a single sensor failure to completely undermine a costly long-term structural monitoring system has been an ever-present fear in the minds of asset managers, undoubtably contributing in part to the broader hesitancy to adopt this technology for real-world structures. The predictive capacity of the proposed hypernetwork framework overcomes these limitations that have inhibited the broader adoption of these sensor technologies throughout the built environment. By mapping entire FOS arrays to implicit neural strain representations, this dependency on individual sensors is significantly reduced. The capacity to predict the strain response at any point within the structure based on the holistic strain response enables the monitoring of critical structural regions, even if one of these critically placed sensors has been damaged.

Given this assertion, it is possible that trained hypernetworks may also significantly reduce the number of FBG sensors required to effectively monitor structural behaviour, as the need for compact senor placement is directly addressed by an unprecedented improvement in predictive output resolution. In turn, this may improve the economic viability of modern SHM systems, particularly when considering long-spanning assets that previously could not justify the upfront cost of extensive, low-interval sensor arrays. This speculation will require further comparative hypernetwork research within the field of applied structural health monitoring.

7.3. Future work

The models introduced throughout this thesis are radical in their approach to solving existing limitations across many phases of the asset life cycle. Each research chapter has provided entirely new processes for solving some of the most challenging barriers that are faced within the fields of structural optimisation, predictive design, and structural health monitoring. These opportunities include neural-automated algorithms, evolutionary machine learning, new methods of scaling 3D design reconstruction models to unprecedented scene complexity, and new real-time visualisation capabilities for sensor-based structural health monitoring. Given the compelling results demonstrated thus far, the future exploration of these techniques will likely yield solutions that greatly overshadow the individual disruptive models trained throughout this thesis.

The performance improvements demonstrated by an adaptive exploration-exploitation DE response reveal a range of potential future research applications for neural networks to mediate algorithmic behavior. While this thesis explored this hybrid approach through the mutation function of the differential evolution algorithm, future research may seek to investigate alternative optimisation parameters such as the population size, selection criteria, crossover rate, or differential weight. Analogously, the potential for neural-based adaptive algorithms extends beyond differential evolution applications, with broader implications across the metaheuristic family. Algorithms such as particle swarm optimisation, simulated annealing, and the genetic algorithm are all viable candidates for variations on the proposed approach. Future research investigating the potential for reinforcement learning to automate algorithmic behavior must also be taken into consideration. For example, PSO offers the possibility of using neural networks to mediate the inertia weight, cognitive and social coefficients, velocity limitations, or swarm size. Similarly, trajectory-based algorithms such as simulated annealing may be adapted according to the neighborhood size or type, the cooling schedule, the step size, or the memory utilisation. The research opportunities that can be undertaken by developing bespoke algorithm-specific solutions at the intersection of metaheuristics and machine learning are vast and unexplored.

Notably, the optimisation strategies developed within this thesis did not require the investigation of multi-objective fitness functions, which would require foundational changes to the underlying algorithmic approaches. This may include introducing methods of recording Pareto dominance, preserving solution diversity, and archiving non-dominate solutions for the chosen algorithm. These changes provide fertile ground for future research within the context of neural-automated optimisation, which was restricted to single-objective neural architecture search and hyperparameter tuning for this thesis. Within the context of the MEMODE algorithm, additional metrics such as inference speed may also be included alongside memory efficiency within a multi-objective function, rather than the proposed fitness penalty. Additionally, while MEMODE has demonstrated remarkable results for evolving efficient multimodal models, future research may seek to adapt the implemented algorithm to generate architectures and hyperparameter sets for highly targeted single-input deep learning tasks

such as named entity recognition, question-answering, object localisation, autoencoding, or generative modelling.

One compelling research opportunity that has not been explored within this thesis includes the capacity for metaheuristic algorithms to optimise the design of hypernetwork architectures, which are notoriously complex due to the codependent relationship between primary and secondary networks. Results from the development and benchmarking of the MEMODE algorithm suggest that methods such as differential evolution are viable approaches for automating the design of complex neural models which are significantly larger than the hypernetworks introduced in larger research chapters. In a sense, the complexity of building hypernetwork models may be seen as a potential factor that has limited their application, despite their competitive performance. Due to the focused research scope that was defined for papers relating to 3D reconstruction and structural monitoring, algorithms such as MEMODE were not investigated within this context. The application of metaheuristics for hypernetwork design remains a compelling extension of the findings from this thesis, which has the potential to vastly improve the performance of models proposed within this thesis. As neural evolution came to inspire the hypernetwork method, these evolutionary algorithms may come to play a part in the broader adoption of hypernetworks as competitive deep learning solutions.

Beyond a hypernetwork-based approach, the proposed 'text-to-structure' model pipeline is directly compatible with alternative methods of 3D reconstruction. The agnostic training method and ensemble architecture allow for the use of voxels, deformation meshes, or point cloud-based models without needing to modify NER extraction or translation vector prediction. Future research may seek to extend the introduced transformer-based scaling method using entirely different reconstruction models to generate the foundational base geometries. The translation vector scaling method is also broadly applicable outside of the context of structural design and engineering. Fields such as geospatial engineering, environmental modeling, manufacturing, industrial design, and medical modeling may also benefit from the scene complexity that translation vectors enable. Additionally, while transformers provide a cutting-edge architecture for translation vector prediction, future research may wish to investigate the viability of alternative deep learning models such as multilayer perceptrons, recurrent neural networks, or graph neural networks. The selection of an appropriate substitute architecture will be entirely contingent on the nature of the given task, the available data, and the performance goals at inference such as accuracy, speed, efficiency, and interpretability. Regardless, the fundamental scaling concepts introduced through these translation vectors may be directly translated to a vast range of future research applications irrespective of any specific model or scope.

While this research has demonstrated compelling methods to scale foundational reconstruction models to meet the complexity of preliminary industry design tasks, future research may seek to expand on the proposed 'text-

to-structure' methodology. For example, the introduced translation vectors are restricted to three-dimensional {x,y,z} vectors, ignoring the potential for additional scaling and rotating parameters to further manipulate the foundational geometries throughout the design space. While these additional translations were not required within the chosen parametric design tasks, more complex real-world design applications will likely require additional forms of element manipulation. Furthermore, the implemented named entity recognition model assumes that the user has provided all of the necessary design variables within their input prompt for each design task. Ambiguous or incomplete user inputs are a known limitation of conventional NER models, which struggle to accurately extract target entities when processing out-of-distribution vocabulary, misleading phrasing, or insufficient contextual information. Future research may seek to amend these challenges by incorporating text preprocessing methods such as text normalisation, entity mapping, coreference, generative prompt completion, probabilistic Bayesian parameter estimation, or sampling missing parameters from a predefined distribution.

The maintenance phase scope of this thesis benefitted significantly from the research undertaken during the benchmarking and design phase research. The resulting hypernetworks provide remarkable opportunities throughout the field of sensor-based SHM. While this thesis has largely focused on the application of novel hypernetworks to predict strain response heatmaps with unprecedented accuracy, resolution, and inference speed, future research may seek to explore the potential opportunities for these deep learning models for alternative structural health monitoring challenges. In particular, the demonstrated topological freedom, resolution, and inference speed of the hypernetwork method for 3D reconstruction tasks offer significant opportunities throughout many fields of computational simulation. Within the context of applied SHM, tasks such as predicting crack propagation, rates of corrosion, vibration response, and structural deflection are particularly well suited to benefit from the radical improvements in predictive capacity offered by the proposed hypernetwork method. This future research will require alternative the methods of training dataset generation, such as cohesive zone modelling (CZM) for modelling crack propagation, diffusion-based modelling for corrosion simulation, or finite element modelling for deflection data collection. The ability of hypernetworks to predict these structural changes at any point within a monitored asset has the potential to radically improve the application of remote sensor technologies for real-world structural health monitoring tasks.

Given their theoretically infinite resolution capacity, spatially rich output, and inference efficiency, there is also a significant opportunity to further explore the predictive capabilities of hypernetworks for predictive structural simulations to inform maintenance procedures. By combining time-series information with cell-based structural datasets generated using finite element analysis, structure-specific training datasets can be built to train highly specialised hypernetworks. Once trained, these models can be used to simulate possible damage or loading scenarios which can be adapted according to severity or location, allowing industry practitioners to experiment with potential failure cases or forecast the potential degradation of assets into the future. The resulting structural response predicted by the trained hypernetworks may then be used to inform operational decisions and

maintenance procedures during the service life of the monitored asset. This thesis has demonstrated the predictive accuracy, resolution, and inference speed of efficiency hypernetwork architectures within the context of sensor-based real-time analysis, however the potential application of these models to enable simulation-based structural health monitoring workflows remains to be realised in future research.

Finally, this thesis has aimed to demonstrate that there are vast interdisciplinary fields of research that have not been investigated within the context of the built environment. Through a multifaceted investigation of the symbiotic applications of metaheuristics and cutting-edge ML models, this research has demonstrated entirely novel research capabilities and eliminated boundaries that have limited conventional approaches. From automating algorithmic behavior and hyperparameter selection, to introducing new visualisation capabilities to the field of real-time structural health monitoring, the concepts introduced within this thesis have far-reaching implications throughout the asset lifecycle. It is the author's hope that the performance of the models trained throughout this thesis will compel further interdisciplinary research throughout the fields of algorithmic structural engineering, predictive design, and sensor-based structural health monitoring.

APPENDICES

Appendix A – Supplementary material for Section 4.2

Table A.1 - Design data for 10-bar truss

Parameter	Definition
Objective function	$\min w(x) = \sum_{i=1}^{10} p_i x_i l_i$
Stress constraints	$\lvert\sigma\rvert_i \leq 25\ (ksi), i = 1,2,\dots,10$
Displacement constraints (Constraint case 2)	$\Delta_{(x,y)i} \leq 2.0\ (in), i = 1,2,\dots,4$
Discrete Cross-sectional areas	$0.1 \leq A_i \leq 34\ (in^2)$
Young Modulus	E = 10^4 (ksi)
Density	ρ = 0.1 (lb/in^3)

Table A.2 - Design data for 15-bar truss

Parameter	Definition
Objective function	$\min w(x) = \sum_{i=1}^{15} p_i x_i l_i$
Stress constraints	$\lvert\sigma\rvert_i \leq 25\ (ksi), i = 1,2,\dots,15$
Discrete Cross-sectional areas	$A_i \in S$ = [0.111, 0.141, 0.174, 0.22, 0.27, 0.287, 0.347, 0.44, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.8, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.3, 10.85, 13.33, 14.29, 17.17, 19.18] (in^2)
Nodal constraints	100$\leq x_2 \leq$ 140; 220 $\leq x_3 \leq$ 260; 100 $\leq y_2 \leq$ 140; 100 $\leq y_3 \leq$ 140; 50 $\leq y_4 \leq$ 90; -20 $\leq y_6 \leq$ 20; -20 $\leq y_7 \leq$ 20; 20 $\leq y_8 \leq$ 60; (in)
Young Modulus	E = 10^4 (ksi)
Density	ρ = 0.1 (lb/in^3)

Table A.3 - Design data for 17-bar truss

Parameter	Definition
Objective function	$\min w(x) = \sum_{i=1}^{17} p_i x_i l_i$
Stress constraints	$\lvert\sigma\rvert_i \leq 50\ (ksi), i = 1,2,\dots,17$
Displacement constraints	$\Delta_{(x,y)i} \leq 2.0\ (in), i = 1,2,\dots,7$
Cross-sectional areas constraints	$A_i \geq 0.1\ in^2$
Young Modulus	E = 30^4 (ksi)
Density	ρ = 0.268 (lb/in^3)

Table A.4 - Design data for 25-bar transmission truss

Parameter	Definition
Objective function	$\min w(x) = \sum_{i=1}^{25} p_i x_i l_i$
Stress constraints	$\lvert\sigma\rvert_i \leq 40\ (ksi), i = 1,2,\dots,25$
Displacement constraints	$\Delta_{(x,y,z)i} \leq 0.3\ (in), i = 1,2,\dots,6$
Discrete Cross-sectional areas	$A_i \in S$ = [0.1,0.2,0.3,0.4,0.5,0.6, 0.7,0.8,0.9,1.0,1.1,1.2,1.3,1.4,1.5,1.6, 1.7,1.8,1.9,2.0,2.1,2.2,2.3,2.4,2.5,2.6, 2.7,2.8,2.9,3.0,3.1,3.2,3.3,3.4] (in^2)
Nodal constraints	$20 \leq x_4 \leq 60$; $40 \leq x_8 \leq 80$; $40 \leq y_4 \leq 80$; $100 \leq y_8 \leq 140$; $90 \leq z_4 \leq 130\ (in)$
Young Modulus	E = 10^4 (ksi)
Density	ρ = 0.1 (lb/in^3)

Table A.5 - Loading conditions 10-bar truss

Node	F_y(kips)
5 (Case 1)	-100
6 (Case 2)	-100

Table A.6 - Loading conditions 15-bar truss

Node	F_y(kips)
8	-10

Table A.7 - Loading conditions 17-bar transmission truss

Node	F_y(kips)
9	-100

Table A.8 - 25 Bar Truss Results

Design variables	GA	PSO	$DE_{Explore}$	$DE_{Exploit}$	DE_{Rand}	DE_{PPO}
A_1	0.2	0.1	0.1	0.1	0.1	0.1
A_2	0.3	0.1	0.1	0.1	0.1	0.1
A_3	0.9	1.0	1.0	1.0	1.0	1.0
A_4	0.1	0.1	0.1	0.1	0.1	0.1
A_5	0.1	0.1	0.1	0.1	0.1	0.1
A_6	0.1	0.1	0.1	0.1	0.1	0.1
A_7	0.2	0.1	0.1	0.1	0.1	0.1
A_8	1.0	1.0	0.9	0.9	0.9	0.9
x_4	20.7369	38.3061	37.6559	37.6911	37.2786	36.3184
y_4	56.7805	56.4962	56.5849	54.3298	60.3216	55.3562
z_4	118.3050	135.4363	128.5169	129.9958	121.5657	129.0145
x_8	40.0772	60.2373	53.813	51.9304	50.3425	51.9709
y_8	129.5759	157.6270	140	139.6331	139.9676	140
Best mass	131.85	129.50	117.73	117.27	117.90	117.41
Average mass	160.45	150.77	122.54	124.41	122.83	121.06
Stdev	14.34	35.76	2.41	5.41	4.35	3.18

Table A.9 - 10 Bar Truss Results, Constraint case 1

Design	GA	PSO	$DE_{Explore}$	$DE_{Exploit}$	DE_{Rand}	DE_{PPO}
A_1	0.3738	0.1	0.1	0.1	0.1	0.1
A_2	0.2058	0.1	0.1	0.1	0.1	0.1
A_3	3.9493	3.9428	3.9368	3.9365	3.9365	3.9365
A_4	0.2559	0.1	0.1	0.1	0.1	0.1
A_5	0.2827	0.1	0.1	0.1	0.1	0.1
A_6	0.1927	0.1	0.1002	0.1	0.1	0.1
A_7	5.5590	5.5678	5.5672	5.5671	5.5671	5.5671
A_8	0.3174	0.1	0.1	0.1	0.1	0.1
A_9	0.4211	0.1	0.1	0.1	0.1	0.1
A_{10}	0.1050	0.1	0.1	0.1	0.1	0.1
Best mass	515.34	458.42	458.44	458.42	458.42	458.42
Average mass	695.06	559.76	462.33	504.03	473.64	460.50
Stdev	110.97	219.86	13.22	119.82	71.72	17.94

Table A.10 - 10 Bar Truss Results, Constraint case 2

Design variables	GA	PSO	$DE_{Explore}$	$DE_{Exploit}$	DE_{Rand}	DE_{PPO}
A_1	2.2201	0.6671	0.8003	0.8029	0.8849	0.7647
A_2	0.4330	0.1	0.1019	0.1	0.1	0.1
A_3	8.7610	7.0388	6.9207	6.6662	6.6706	6.7228
A_4	0.2949	0.1041	0.2382	0.3085	0.3059	0.2314
A_5	0.1455	0.1	0.2541	0.2765	0.2655	0.2958
A_6	0.2552	0.1	0.1	0.1	0.1004	0.1001
A_7	6.7567	7.3749	6.6999	6.6949	6.6964	6.7470
A_8	0.3263	0.3662	0.3148	0.3137	0.3097	0.3201
A_9	0.6949	0.2617	0.3119	0.3880	0.3451	0.3761
A_{10}	0.1551	0.1	0.1	0.1	0.1	0.1001
Best mass	839.84	702.34	681.05	678.81	679.14	679.78
Average mass	1087.36	855.74	703.46	750.09	725.49	696.22
Stdev	159.85	173.71	19.41	127.95	113.20	29.22

Table A.11 - 15 Bar Truss Results

Design	GA	PSO	$DE_{Explore}$	$DE_{Exploit}$	DE_{Rand}	DE_{PPO}
A_1	0.954	0.954	0.954	0.954	0.954	0.954
A_2	0.539	0.539	0.539	0.539	0.539	0.539
A_3	0.270	0.111	0.174	0.220	0.174	0.111
A_4	1.081	0.954	0.954	0.954	0.954	0.954
A_5	0.539	0.539	0.539	0.539	0.539	0.539
A_6	0.141	0.287	0.270	0.220	0.270	0.347
A_7	0.111	0.111	0.111	0.111	0.111	0.111
A_8	0.111	0.111	0.111	0.111	0.111	0.111
A_9	0.270	0.111	0.347	0.220	0.174	0.111
A_{10}	0.44	0.44	0.44	0.44	0.44	0.44
A_{11}	0.347	0.44	0.44	0.44	0.44	0.44
A_{12}	0.270	0.22	0.220	0.270	0.174	0.174
A_{13}	0.270	0.27	0.220	0.220	0.174	0.174
A_{14}	0.141	0.347	0.270	0.220	0.270	0.347
A_{15}	0.270	0.111	0.174	0.220	0.174	0.111
x_2	106.836	100.9605	108.8043	100.0000	104.5228	104.4987
x_3	255.310	244.1584	241.5257	226.3391	259.9883	259.9066
y_2	126.685	138.6257	134.9495	136.8152	136.7618	136.4470
y_3	124.992	128.4445	123.2012	131.1371	110.4611	109.9109
y_4	55.5573	52.1720	57.4724	56.5675	60.0001	59.3918
y_6	-12.4127	-12.7069	-14.4307	-15.8742	-16.7569	-17.0394
y_7	2.6396	-4.7362	-0.7277	-8.0120	9.5387	9.0587
y_8	55.7448	52.0353	56.4540	56.5628	60.0000	59.4751
Best mass	74.70	74.39	73.89	73.81	72.10	72.41
Average mass	82.16	80.16	79.91	79.48	77.71	77.62
Stdev	3.20	3.25	2.40	2.37	2.30	2.43

Table A.12 - 17 Bar Truss Results

Design variables	GA	PSO	$DE_{Explore}$	$DE_{Exploit}$	DE_{Rand}	DE_{PPO}
A_1	17.6828	15.8099	15.5102	15.8806	16.0760	16.0647
A_2	11.0916	8.0579	8.07915	8.0894	8.1151	8.1026
A_3	7.9567	7.8936	8.4807	8.0608	8.0042	7.6778
A_4	4.2624	5.6457	5.2015	5.5247	5.6930	5.5064
A_5	9.5552	12.3225	12.0382	12.1696	11.9776	12.0360
A_6	9.9928	11.8798	11.7029	11.7446	12.0661	11.9921
A_7	4.2502	4.1179	3.9243	4.2869	4.0962	4.0648
A_8	5.6681	4.0806	4.0081	4.0280	3.9569	4.0364
A_9	0.1529	0.1	0.1742	0.1002	0.1016	0.1196
A_{10}	0.8658	0.1	0.1	0.1	0.1011	0.1
A_{11}	0.2541	0.1024	0.1	0.1004	0.1	0.1145
A_{12}	0.5766	0.1	0.2508	0.1003	0.1	0.1854
A_{13}	8.4519	5.5237	5.8476	5.4849	5.4082	5.5443
A_{14}	0.5534	0.1	0.1076	0.1000	0.1002	0.1
A_{15}	5.0900	5.2618	5.9384	5.5392	5.5895	5.5772
A_{16}	0.6115	0.1	0.1238	0.1	0.1066	0.1
A_{17}	5.3239	5.6294	5.5681	5.6402	5.4632	5.7060
Best mass	2750.78	2582.38	2591.51	2582.58	2582.42	2584.55
Average mass	2962.45	2711.64	2652.29	2637.73	2626.12	2620.09
Stdev	123.64	159.97	54.50	64.82	62.95	45.12

Table A.13 - Loading conditions 25-bar transmission truss

Node	F_x(kips)	F_y(kips)	F_z(kips)
1	1.0	-10	-10
2	0.0	-10	-10
3	0.5	0.0	0.0
6	0.6	0.0	0.0

Appendix B – Supplementary material for Section 4.3

AlexNet

In order to benchmark the proposed MEMODE method, a series of additional models have been developed and evaluated for the given multilabel classification task. The structure of the AlexNet model outlined in Figure 8B.1 has been adapted from the original paper to meet the benchmarking needs of this research project [452]. The 3-channel RGB input outlined in the original paper has been replaced with the proposed 4-channel signal, including the generated word matrix for each caption. The first 2D convolutional layer contains a kernel size of 11 and a stride of 4, outputting 64 feature maps processed by a ReLU activation. A max-pooling layer with a kernel size of 3 and stride of 2 then down samples the feature maps to reduce spatial dimensions of the signal. The next two convolutional blocks follow an analogous structure, with kernel sizes of 5 and 3 outputting 192 and 384 filter maps respectively. The final two convolutional layers have a consistent kernel size of 3×3, outputting 256 feature maps per layer. The fully connected region of the AlexNet model has been retained, consisting of two dense layers containing 4096 neurons each. The AlexNet framework tested within this thesis has a downstream connection to the parallel natural language processing ALBERT module, with signals combined through element-wise multiplication.

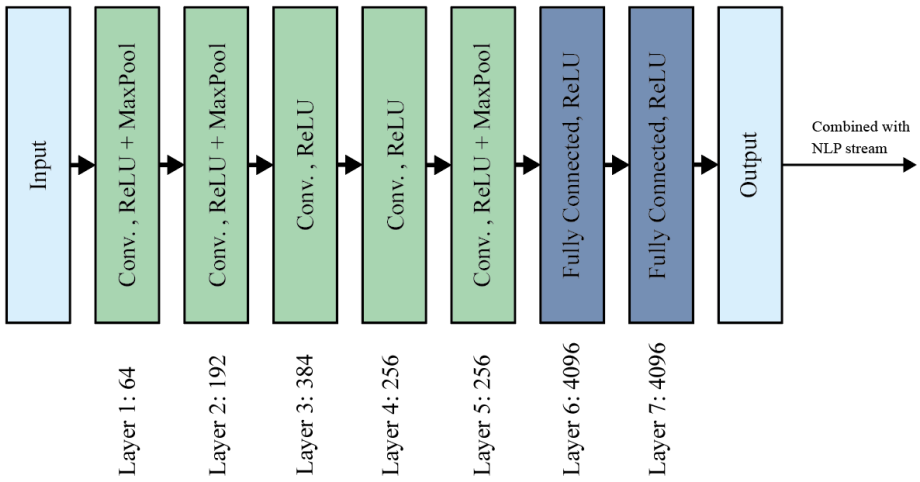


Figure B.1 - Adapted AlexNet architecture

ResNet

The residual neural network (ResNet) is a deep neural network architecture that uses skip connections to pass input signals to deeper layers within the network without modification [453]. ResNets address key issues relating to the vanishing gradient problem by introducing skip connections, which allow the network to learn residual mappings. In turn, the flow of gradients during backpropagation is improved, reducing the influence of the vanishing gradient as signal propagation is retained [454]. Two different ResNet architectures have been benchmarked within this research project, with the intention of reducing model sizes to below 100 MB while maximising performance. The first, ResNet_{small}, starts with a convolutional layer with a kernel size of 3, outputting 64 feature maps. These feature maps are normalised using batch normalisation, before being

processed by a ReLU activation and a max pooling layer. This is followed by four residual blocks, each containing two convolutional layers with batch normalisation and ReLU processing. The first residual block has 64 input and output filter maps, while the second, third and fourth blocks output 128, 256, and 512 filter maps respectively. Down sampling layers containing 1×1 convolutional kernels and a stride of 2 are used after the second, third and fourth residual blocks to ensure signals propagated through residual connections match the outputs from residual blocks. The last residual block is passed to an adaptive average pooling layer, before a dense layer is used to map the signal output to 19 nodes, representing each class. These signals are combined with the output from the parallel NLP stream through element-wise multiplication, before being passed to the final fully connected classification module. The second ResNet architecture developed for benchmarking, ResNet_{large}, starts with the same cov-norm-relu framework as the first residual model and all of its convolutional kernels have a size of 3×3, besides down sampling layers. ResNet_{large} contains a total of eight residual blocks, the first two of which have 64 input and output filter maps. The second and third residual blocks output 128 filter maps, with a corresponding down sampling convolutional layer. This pattern of paired residual blocks is repeated for the remainder of the residual layers, with grouped blocks following output filter maps sizes of 128, 256 and 512. Following the residual region, two fully connected layers containing 2560 neurons, 19 outputs and ReLU activations are used. These outputs are processed with the signal output from the ALBERT module, with multimodal streams combined through element-wise multiplication.

Inception Network

While a traditional architecture would require deeper layers to extract different scales of features within the feature map, the inception model effectively provides a variety of kernel sizes, allowing the architecture to be more efficient with respect to its depth. The Inception Network used within this research has been inspired by the Google Inception paper, comprising multiple Inception modules that capture different scales of features [455]. The size of the model itself has been adjusted to align with the memory requirements, allowing for a comparable benchmarking procedure. The Inception model used consists of 3 initial convolutional layers with ReLU activations and two max pooling operations after the first and third layers. The output feature maps are then passed to the first inception module. Each Inception module consists of four branches, the first of which contains a 1×1 convolution. The second branch reduces the spatial dimension of the input with at 1×1 computation followed by a 3×3 convolution layer. The third branch consists of a similar 1×1 convolution, followed by a 5×5 convolutional layer. Finally, the fourth branch applies a max-pooling operation with a 3×3 kernel, followed by a final 1×1 convolution. These branches process the input image in parallel, and their outputs are concatenated to obtain diverse and informative features. The Inception network learns hierarchical representations through these varied processing branches, contributing to improved image feature extraction. There are a total of 9 inception modules sequentially staged within this architecture, with the first inception modules taking an input channel size of 192 and outputting 256 feature maps. The remaining inception modules output 480, 512, 512, 512, 528, 832, 832 and 1024 feature maps respectively. An adaptive pooling layer is used

to process the output feature maps from the final inception module before being input to a fully connected layer containing a node for each of the 19 classes present within the training dataset. The framework for the inception module that has been used within this thesis is provided in Figure 8B.2.

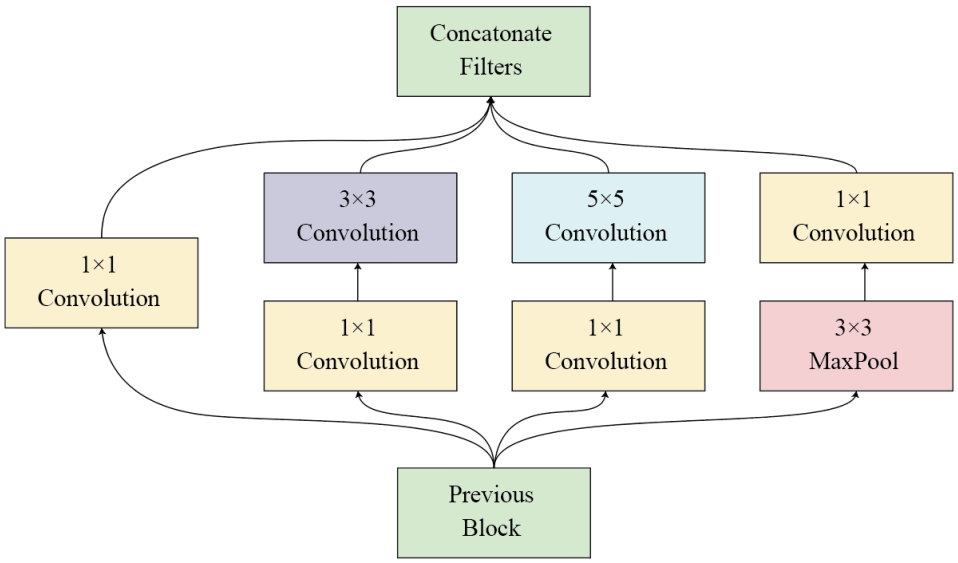


Figure B.2 - Example Inception module

Manual CNN

In order to evaluate the success of the final neural network that has been evolved through Differential Evolution, a manually designed convolutional neural network has been created. This provides a performance baseline for neural network architectures to evaluate the success of the proposed DE optimisation algorithm. The network architecture used within this baseline consists of an initial MaxPooling layer with a kernel size of 2, which effectively down-samples the data, reducing its spatial dimension by a factor of 2. This is followed by a convolutional layer with 64 outputs and a kernel size of 3, followed by another MaxPooling layer. This pattern is repeated for convolutional output channels of size 128, 256 and 512, which are each followed by MaxPooling modules. A fully connected region is then used to process the final flattened feature maps. All of the models benchmarked within this report benefit from the multimodality of the proposed architecture, fusing the outputs from the convolutional and NLP regions of the model before being processed by a final fully connected region.

MEGA, MEPSO, and MEBO implementations

The benchmarking of the proposed Differential Evolution algorithm required additional algorithms to be developed. For this research project, the Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO) were chosen, due to their widespread application within the field of metaheuristics and neural architecture search. In order to ensure a fair benchmarking comparison, all implementations were provided with the same population size and generation count, with a total of 1,600 networks generated, trained and evaluated across all algorithms. For the implemented Genetic algorithm, the same crossover and scaling factors were used compared to the

proposed Differential Evolution algorithm (Cr = F= 0.5). For the MEPSO algorithm, the cognitive and social weights were both set to 1.5, such that personal and social influences are given equal precedence. The Bayesian Optimisation implementation (MEBO) introduced within this thesis incorporated a traditional Expected improvement (EI) utility function, an initial population of 20, and a total of 400 trained networks.

Appendix C – Supplementary material for Section 5.2

Table C.1 – Comparison of traditional optimisation methods and hypernetworks

Attribute	Trained Hypernetworks	Metaheuristic Algorithms	Topology Optimisation	Bayesian Optimisation
Design approach	Deep learning-based architecture	Stochastic search	Gradient-based mathematical modelling	Probability-based surrogate modelling
Model flexibility	Capable of generating structures of any topology with unlimited resolution	Fixed, defined algorithm with limited flexibility	Fixed mathematical optimisation with limited control over final topology	Fixed probabilistic surrogate model
Adaptability	Limited by scope of training data and network size	Problem-specific tuning of algorithm hyperparameters	Requires specialist setup for each design	Specifically used for low-dimensional, high cost functions
Engineering Scope	Any engineering scope present within training data	Tuning of design parameters such as size, shape and schematic optimisation	Additive manufacturing, limited use with traditional construction techniques	Low-dimensional, high cost black-box functions
Optimisation speed	Inference time for trained model is extremely quick	Slow for computational expensive objective functions	Performance hindered for complex structures	Primarily efficient for computationally expensive objective functions
Application for high-dimensional tasks	Capable of training off high-dimensional solutions	Difficulty converging consistently	Known challenges in handling complex structures	Less effective beyond 20+ dimensions
Capability for training	Learns from optimal designs that have been generated previously	Data-independent. Only uses data from current optimisation run	Data-independent. Only uses data from current optimisation run	Data-independent. Only uses data from current optimisation run
Solution Quality	Inherits quality of training data	Solutions depend on parameter tuning, quality varying between runs	Mathematically optimises topology to find near-optimal designs	High-quality global optimisation for low-dimensional tasks
Ease of use	Extremely simple once trained	Hyperparameter selection requires experience	Requires expertise in model formulation	Statistical theory is a barrier to entry
Output	Manifold geometry, unlimited resolution, any desired topological form	Tuned set of parameters for the chosen task	Manifold geometry according to mathematical modelling	Optimal parameter set for expensive black-box functions
Advantages	Can be trained across all methods, extremely fast inference time	Emphasis on constructability, control over output bounds	Freedom of topology for output geometry	Efficient for computationally expensive tasks
Limitations	Requires high-quality training data	Premature/slow convergence, susceptible to chosen parameters	Output topology sacrifices practically / constructability	Suffers from the curse of dimensionality

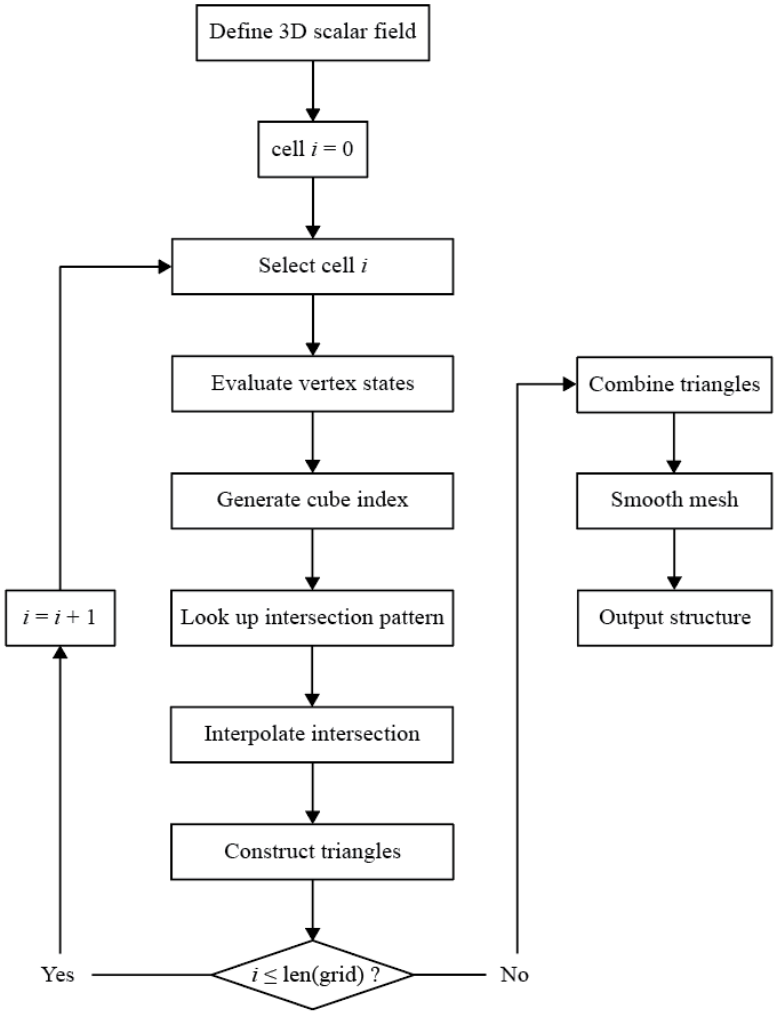


Figure C.1 – Marching Cubes algorithm for structural meshing

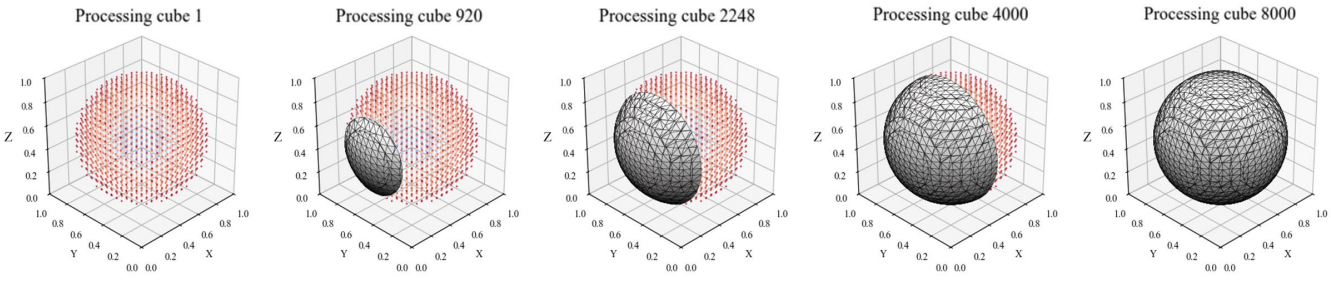


Figure C.2 — Diagrammatic example of the Marching Cubes algorithm across a sphere point cloud

REFERENCES

1. Zavala, G.R., et al., *A survey of multi-objective metaheuristics applied to structural optimization.* Structural and Multidisciplinary Optimization, 2014. **49**(4): p. 537-558.
2. Yi, H., et al., *Rapid simulation of optimally responsive façade during schematic design phases: Use of a new hybrid metaheuristic algorithm.* Sustainability, 2019. **11**(9): p. 2681.
3. Yuan, X., et al., *An effective metaheuristic approach for building energy optimization problems.* Buildings, 2022. **13**(1): p. 80.
4. Delgarm, N., B. Sajadi, and S. Delgarm, *Multi-objective optimization of building energy performance and indoor thermal comfort: A new method using artificial bee colony (ABC).* Energy and Buildings, 2016. **131**: p. 42-53.
5. Prayogo, D., et al., *A novel hybrid metaheuristic algorithm for optimization of construction management site layout planning.* Algorithms, 2020. **13**(5): p. 117.
6. John, M.P., *Metaheuristics for designing efficient routes & schedules for urban transportation networks*. 2016, Cardiff University.
7. Hassani, S. and U. Dackermann, *A systematic review of optimization algorithms for structural health monitoring and optimal sensor placement.* Sensors, 2023. **23**(6): p. 3293.
8. Deb, K., et al. *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*. in *International conference on parallel problem solving from nature*. 2000. Springer.
9. Fieldsend, J.E., *Multi-objective particle swarm optimisation methods.* 2004.
10. Xue, F., A.C. Sanderson, and R.J. Graves. *Pareto-based multi-objective differential evolution*. in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.* 2003. IEEE.
11. Goolsbee, A. and C. Syverson, *The strange and awful path of productivity in the US construction sector.* 2023, National Bureau of Economic Research Cambridge, MA, USA.
12. Baily, M.N. and B.P. Bosworth, *US manufacturing: Understanding its past and its potential future.* Journal of Economic Perspectives, 2014. **28**(1): p. 3-26.
13. Wang, S.L., et al., *Agricultural productivity growth in the United States: Measurement, trends, and drivers.* Economic Research Service, Paper No. Err-189, 2015.
14. Neve, H.H., et al., *Determining the relationship between direct work and construction labor productivity in North America: Four decades of insights.* Journal of Construction Engineering and Management, 2020. **146**(9): p. 04020110.
15. Statistics, U.B.o.L., *Employment and output in construction industries*. 2023.
16. Abdel-Wahab, M. and B. Vogl, *Trends of productivity growth in the construction industry across Europe, US and Japan.* Construction management and economics, 2011. **29**(6): p. 635-644.
17. Commision, P., *Housing construction productivity: Can we fix it?* 2025.
18. PWC, *Reconstructing productivity*. 2013.
19. Dufty, N. and T. Jackson, *Information and communication technology use in Australian agriculture: a survey of broadacre, dairy and vegetable farms.* 2018.
20. Wilson, J. and C. Dowling, *Technology adoption in Australian industry: commercial, workforce and regulatory drivers.*
21. Administration, F.T., *2015 status of the Nation's highways, bridges, and transit conditions and performance report to congress*. 2017: Government Printing Office.
22. Woodward, R., et al., *Bridge management in Europe (BRIME)-deliverable D14-final report.* 2001.
23. Caprani, C., *Are Australia's bridges at risk of a catastrophic, Genoa-style collapse?* 2018.

24. Board, N.T.S., *Highway Investigation Report HIR-24-02: Collapse of the Fern Hollow Bridge.* 2024.

25. Tamimi, M.F., A.A. Alshannaq, and M.a.I. AbuQamar, *Deterioration of steel structures due to corrosion considering the global effects of climate change.* Corrosion Engineering, Science and Technology, 2024. **59**(4): p. 273-293.

26. Davidson, K., *'We have not forgotten you' – Residents remain without relief 5 years after failure of Edenville Dam*, in *Michigan Advance*. 2025.

27. Booty, C.S.N., *Libyan officials jailed over deadly floods*, in *BBC*. 2024.

28. Hub, T.G.I., *Global Infrastructure Outlook*. 2018.

29. Engineers, A.S.o.C., *Failure to Act: Economic Impacts of Status Quo Investment Across Infrastructure System*. 2021, EBP.

30. Huang, J., et al., *Framework for a practical and cost-effective IoT-enhanced structural health monitoring and damage diagnostics system with digital twinning.* Journal of Civil Structural Health Monitoring, 2025: p. 1-26.

31. Seventekidis, P., et al., *Structural Health Monitoring using deep learning with optimal finite element model generated data.* Mechanical Systems and Signal Processing, 2020. **145**: p. 106972.

32. Jayawickrema, U., et al., *Fibre-optic sensor and deep learning-based structural health monitoring systems for civil structures: A review.* Measurement, 2022. **199**: p. 111543.

33. Altabey, W.A., et al., *Structural health monitoring of composite pipelines utilizing fiber optic sensors and an AI-based algorithm—A comprehensive numerical study.* Sensors, 2023. **23**(8): p. 3887.

34. Zhou, Y., et al., *Application of machine learning in optical fiber sensors.* Measurement, 2024. **228**: p. 114391.

35. Bandara, S., M. Herath, and J. Epaarachchi, *Sensory methods and machine learning based damage identification of fibre-reinforced composite structures: An introductory review.* Journal of Reinforced Plastics and Composites, 2023. **42**(21-22): p. 1119-1146.

36. Yu, J., et al., *Localization of impact on CFRP structure based on fiber Bragg gratings and CNN-LSTM-Attention.* Optical Fiber Technology, 2024. **87**: p. 103943.

37. Venketeswaran, A., et al., *Recent advances in machine learning for fiber optic sensor applications.* Advanced Intelligent Systems, 2022. **4**(1): p. 2100067.

38. Phares, B.M., et al., *Reliability and Accuracy of Routine Inspection of Highway Bridges.* Transportation Research Record, 2001. **1749**(1): p. 82-92.

39. Zhu, L., et al. *Development of a High-Sensitivity Wireless Accelerometer for Structural Health Monitoring*. Sensors, 2018. **18**, DOI: 10.3390/s18010262.

40. Tcherniak, D. and L.L. Mølgaard, *Vibration-based SHM System: Application to Wind Turbine Blades.* Journal of Physics: Conference Series, 2015. **628**(1): p. 012072.

41. Yoon, J., et al., *Deep neural network-based structural health monitoring technique for real-time crack detection and localization using strain gauge sensors.* Scientific Reports, 2022. **12**(1): p. 20204.

42. Nagulapally, P., et al., *Distributed Fiber Optic Sensor-Based Strain Monitoring of a Riveted Bridge Joint Under Fatigue Loading.* IEEE Transactions on Instrumentation and Measurement, 2021. **70**: p. 1-10.

43. Sieńko, R., et al., *Strain and crack analysis within concrete members using distributed fibre optic sensors.* Structural Health Monitoring, 2019. **18**(5-6): p. 1510-1526.

44. Nhung, N.T.C., et al., *Development and Application of Linear Variable Differential Transformer (LVDT) Sensors for the Structural Health Monitoring of an Urban Railway Bridge in Vietnam.* Engineering, Technology & Applied Science Research, 2023. **13**(5): p. 11622-11627.

45. Park, H.S., et al. *A Wireless Laser Displacement Sensor Node for Structural Health Monitoring*. Sensors, 2013. **13**, 13204-13216 DOI: 10.3390/s131013204.

46. Schmit, L.A. *Structural design by systematic synthesis*. in *Proceedings of the Second National Conference on Electronic Computation, ASCE, Sept., 1960*. 1960.

47. Bo, G., et al., *Optimum structure of a combined wind/photovoltaic/fuel cell-based on amended Dragon Fly optimization algorithm: a case study.* Energy sources, part A: recovery, utilization, and environmental effects, 2022. **44**(3): p. 7109-7131.

48. Aydın, Z. and Y. Ayvaz, *Overall cost optimization of prestressed concrete bridge using genetic algorithm.* KSCE Journal of Civil Engineering, 2013. **17**(4): p. 769-776.

49. Rana, S., et al., *Application of evolutionary operation to the minimum cost design of continuous prestressed concrete bridge structure.* Engineering Structures, 2013. **46**: p. 38-48.

50. Senouci, A.B. and M.S. Al-Ansari, *Cost optimization of composite beams using genetic algorithms.* Advances in Engineering Software, 2009. **40**(11): p. 1112-1118.

51. Kim, H. and H. Adeli, *Discrete cost optimization of composite floors using a floating-point genetic algorithm.* Engineering Optimization, 2001. **33**(4): p. 485-501.

52. Gan, V.J.L., et al., *Parametric modelling and evolutionary optimization for cost-optimal and low-carbon design of high-rise reinforced concrete buildings.* Advanced Engineering Informatics, 2019. **42**: p. 100962.

53. Aslay, S.E. and T. Dede, *Reduce the construction cost of a 7-story RC public building with metaheuristic algorithms.* Architectural Engineering and Design Management, 2024. **20**(2): p. 214-229.

54. Chaudhuri, P. and D. Maity, *Cost optimization of rectangular RC footing using GA and UPSO.* Soft Computing, 2020. **24**(2): p. 709-721.

55. Bel Hadj Ali, N., et al., *Multi-stage production cost optimization of semi-rigid steel frames using genetic algorithms.* Engineering Structures, 2009. **31**(11): p. 2766-2778.

56. Pham, B.T., *Optimum cost design of frames using genetic algorithms.* Steel Compos Struct, 2019.

57. Kanyilmaz, A., P.R.N. Tichell, and D. Loiacono, *A genetic algorithm tool for conceptual structural design with cost and embodied carbon optimization.* Engineering Applications of Artificial Intelligence, 2022. **112**: p. 104711.

58. Vukadinović, A., et al., *Multi-objective optimization of energy performance for a detached residential building with a sunspace using the NSGA-II genetic algorithm.* Solar Energy, 2021. **224**: p. 1426-1444.

59. Mei, L. and Q. Wang *Structural Optimization in Civil Engineering: A Literature Review*. Buildings, 2021. **11**, DOI: 10.3390/buildings11020066.

60. Faghihi, V., K.F. Reinschmidt, and J.H. Kang, *Construction scheduling using Genetic Algorithm based on Building Information Model.* Expert Systems with Applications, 2014. **41**(16): p. 7565-7578.

61. Gandomi, A.H. and X.-S. Yang, *Benchmark Problems in Structural Optimization*, in *Computational Optimization, Methods and Algorithms*, S. Koziel and X.-S. Yang, Editors. 2011, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 259-281.

62. Hare, W., J. Nutini, and S. Tesfamariam, *A survey of non-gradient optimization methods in structural engineering.* Advances in Engineering Software, 2013. **59**: p. 19-28.

63. Talbi, E.G., *Metaheuristics: from design to implementation.* John Wiley & Sons., 2009.

64. Kumar, A., *Chapter 5 - Application of nature-inspired computing paradigms in optimal design of structural engineering problems—a review*, in *Nature-Inspired Computing Paradigms in Systems*, M.A. Mellal and M.G. Pecht, Editors. 2021, Academic Press. p. 63-74.

65. Rutten, D., *Galapagos: On the logic and limitations of generic solvers.* Architectural Design, 83(2), pp.132-135., 2013.

66. Pitzer, E. and M. Affenzeller, *A Comprehensive Survey on Fitness Landscape Analysis*, in *Recent Advances in Intelligent Engineering Systems*, J. Fodor, R. Klempous, and C.P. Suárez Araujo, Editors. 2012, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 161-191.

67. Sörensen, *Metaheuristics-the metaphor exposed.* International Transactions in Operational Research, 22(1), pp.3-18., 2013.

68. Walbridge, *Genetic algorithms: What computers can learn from Darwin.* Technology Review, 1989.

69. Zhang, X. and X. Zhang, *Low-carbon design optimization of reinforced concrete building structures using genetic algorithm.* Journal of Asian Architecture and Building Engineering, 2024. **23**(6): p. 1888-1902.

70. Feng, R., et al., *A hybrid of genetic algorithm and particle swarm optimization for reducing material waste in extrusion-basedadditive manufacturing.* Rapid Prototyping Journal, 2021. **27**(10): p. 1872-1885.

71. Hsu, C.-Y., Y.-R. Chen, and C.-K. Ting. *Structural Topology Optimization Using Genetic Algorithm and Fractals*. in *Technologies and Applications of Artificial Intelligence*. 2024. Singapore: Springer Nature Singapore.

72. Zhang, Q., *Research on the Construction Schedule and Cost Optimization of Grid Structure based on BIM and Genetic Algorithm.* Journal of Physics: Conference Series, 2021. **1744**(2): p. 022065.

73. Jebari, K.a.M., M., *Selection methods for genetic algorithms.* International Journal of Emerging Sciences, 2013.

74. Mirjalili, S., *Evolutionary algorithms and neural networks.* Studies in computational intelligence, 780(1), pp.43-53., 2019.

75. Hassanat, A., et al. *Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach*. Information, 2019. **10**, DOI: 10.3390/info10120390.

76. Hegerty, B., Hung, C.C. and Kasprak, K., *A comparative study on differential evolution and genetic algorithms for some combinatorial problems.* In Proceedings of 8th Mexican international conference on artificial intelligence (Vol. 9, p. 13). 2009.

77. Ahmad, M.F., et al., *Differential evolution: A recent review based on state-of-the-art works.* Alexandria Engineering Journal, 2022. **61**(5): p. 3831-3872.

78. Sacco, W.F. and N. Henderson, *Balancing exploration and exploitation in differential evolution via variable scaling factors: An application to practical problems.* Progress in Nuclear Energy, 2015. **83**: p. 365-373.

79. Venter, G. and S.-S. Jaroslaw, *Particle Swarm Optimization.* AIAA Journal, 2002. **41**.

80. Malan, K. and A. Engelbrecht, *Ruggedness, funnels and gradients in fitness landscapes and the effect on PSO performance*. 2013. 963-970.

81. Malan, K.M. and A.P. Engelbrecht. *Particle swarm optimisation failure prediction based on fitness landscape characteristics*. in *2014 IEEE Symposium on Swarm Intelligence*. 2014.

82. Greenhill, S., et al., *Bayesian Optimization for Adaptive Experimental Design: A Review.* IEEE Access, 2020. **8**: p. 13937-13948.

83. Kallioras, N.A. and N.D. Lagaros, *DzAI𝕟: Deep learning based generative design.* Procedia Manufacturing, 2020. **44**: p. 591-598.

84. Meng, X.B., Li, H.X. and Gao, X.Z, *An adaptive reinforcement learning-based bat algorithm for structural design problems.* International Journal of Bio-Inspired Computation, 14(2), pp.114-124, 2019.

85. Salleh, M.N.M., et al. *Exploration and Exploitation Measurement in Swarm-Based Metaheuristic Algorithms: An Empirical Analysis*. in *Recent Advances on Soft Computing and Data Mining*. 2018. Cham: Springer International Publishing.

86. Beyer, H.-G. *On the "Explorative power" of ES/EP-like algorithms*. in *Evolutionary Programming VII*. 1998. Berlin, Heidelberg: Springer Berlin Heidelberg.

87. Jalali, A., et al. *A Lipschitz Exploration-Exploitation Scheme for Bayesian Optimization*. in *Machine Learning and Knowledge Discovery in Databases*. 2013. Berlin, Heidelberg: Springer Berlin Heidelberg.

88. Hoffman, M., Brochu, E. and De Freitas, N., *Portfolio Allocation for Bayesian Optimisation.* UAI 2011.

89. Huynh, T.N., D.T.T. Do, and J. Lee, *Q-Learning-based parameter control in differential evolution for structural optimization.* Applied Soft Computing, 2021. **107**: p. 107464.

90. Wang, F., X. Wang, and S. Sun, *A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization.* Information Sciences, 2022. **602**: p. 298-312.

91. Sehgal, A., et al. *Deep Reinforcement Learning Using Genetic Algorithm for Parameter Optimization*. in *2019 Third IEEE International Conference on Robotic Computing (IRC)*. 2019.

92. Shende, S., et al., *Bayesian topology optimization for efficient design of origami folding structures.* Structural and Multidisciplinary Optimization, 2021. **63**(4): p. 1907-1926.

93. Gramacy, R.B., *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences.* CRC Press, 2020.

94. Williams, C.K.a.R., C.E., *Gaussian processes for machine learning.* Cambridge, MA: MIT press., 2006.

95. Brochu, E., V. M. Cora and N. de Freitas *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning.* 2010.

96. Mathern, A., et al., *Multi-objective constrained Bayesian optimization for structural design.* Structural and Multidisciplinary Optimization, 2021. **63**(2): p. 689-701.

97. Do, B., M. Ohsaki, and M. Yamakawa, *Bayesian optimization for robust design of steel frames with joint and individual probabilistic constraints.* Engineering Structures, 2021. **245**: p. 112859.

98. Penadés-Plà, V., T. García-Segura, and V. Yepes, *Accelerated optimization method for low-embodied energy concrete box-girder bridge design.* Engineering Structures, 2019. **179**: p. 556-565.

99. Forrester, A., Sobester, A. and Keane, A., *Forrester, A., Sobester, A. and Keane, A., 2008. Engineering design via surrogate modelling: a practical guide.* John Wiley & Sons., 2008.

100. Shahriari, B., et al., *Taking the Human Out of the Loop: A Review of Bayesian Optimization.* Proceedings of the IEEE, 2016. **104**(1): p. 148-175.

101. Shahriari, B., Wang, Z., Hoffman, M.W., Bouchard-Côté, A. and de Freitas, N., *An entropy search portfolio for Bayesian optimization.* 2014.

102. Jakkala, K., *Deep Gaussian Processes: A Survey.* 2021.

103. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors.* Nature, 1986. **323**(6088): p. 533-536.

104. Moriguchi, H. and S. Honiden, *CMA-TWEANN: efficient optimization of neural networks via self-adaptation and seamless augmentation*, in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. 2012, Association for Computing Machinery: Philadelphia, Pennsylvania, USA. p. 903–910.

105. Stanley, K.O., D.B. D'Ambrosio, and J. Gauci, *A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks.* Artificial Life, 2009. **15**(2): p. 185-212.

106. Lorenzo, P.R., et al., *Particle swarm optimization for hyper-parameter selection in deep neural networks*, in *Proceedings of the Genetic and Evolutionary Computation Conference*. 2017, Association for Computing Machinery: Berlin, Germany. p. 481–488.

107. Schmidt, M., et al. *On the Performance of Differential Evolution for Hyperparameter Tuning*. in *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019.

108. Xiao, X., Yan, M., Basodi, S., Ji, C. and Pan, Y., *Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm.* 2020.

109. Ha, D., Dai, A. and Le, Q.V., *Hypernetworks.* 2016.

110. Brock, A., Lim, T., Ritchie, J.M. and Weston, N., *Smash: one-shot model architecture search through hypernetworks.* 2017.

111. Sukthanker, R.S., Zela, A., Staffler, B., Dooley, S., Grabocka, J. and Hutter, F., *Multi-objective differentiable neural architecture search.* 2024.

112. Yoo, Y.H., Cha, J., Kim, C. and Kim, T., *Hyper-cl: Conditioning sentence representations with hypernetworks.* 2024.

113. Volk, T., et al. *Example-based Hypernetworks for Multi-source Adaptation to Unseen Domains*. 2023. Singapore: Association for Computational Linguistics.

114. Akinwande, V., Norouzzadeh, M.S., Willmott, D., Bair, A., Ganesh, M.R. and Kolter, J.Z., *HyperCLIP: Adapting Vision-Language models with Hypernetworks.* 2024.

115. Ferens, R.a.K., Y., *HyperPose: Hypernetwork-Infused Camera Pose Localization and an Extended Cambridge Landmarks Dataset.* Proceedings of the Computer Vision and Pattern Recognition Conference, 2025.

116. Spurek, P., Winczowski, S., Tabor, J., Zamorski, M., Zięba, M. and Trzciński, T., *Hypernetwork approach to generating point clouds.* 2020.

117. Li, J., N. Qi, and Q. Zhu. *Hyper-NeuS: Hypernetworks for Neural SDF Implicit Surface Reconstruction by Volume Rendering*. in *MultiMedia Modeling*. 2025. Singapore: Springer Nature Singapore.

118. Simon, C., He, S., Perez-Rua, J.M., Xu, M., Benhalloum, A. and Xiang, T., *Hyper-VolTran: fast and generalizable one-shot image to 3D object structure via hypernetworks.* 2023.

119. Littwin, G.a.W., L., *Deep meta functionals for shape representation.* Proceedings of the IEEE/CVF international conference on computer vision, 2019.

120. Liu, F.a.L., X., *Voxel-based 3D detection and reconstruction of multiple objects from a single image.* Advances in neural information processing systems, 2021.

121. Xiong, W., et al., *3D voxel reconstruction from single-view image based on cross-domain feature fusion.* Expert Systems with Applications, 2024. **256**: p. 124957.

122. Xie, H., et al., *Weighted voxel: a novel voxel representation for 3D reconstruction*, in *Proceedings of the 10th International Conference on Internet Multimedia Computing and Service*. 2018, Association for Computing Machinery: Nanjing, China. p. Article 33.

123. Paschalidou, D., Ulusoy, O., Schmitt, C., Van Gool, L. and Geiger, A., *Raynet: Learning volumetric 3d reconstruction with ray potentials.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

124. Li, Z., M. Oskarsson, and A. Heyden, *Detailed 3D human body reconstruction from multi-view images combining voxel super-resolution and learned implicit representation.* Applied Intelligence, 2022. **52**(6): p. 6739-6759.

125. Yao, L., Xu, R., Gao, Z., Ke, G. and Wang, Y., *Boosted ab initio Cryo-EM 3D Reconstruction with ACE-EM.* 2023.

126. Osher, S. and R. Fedkiw, *Signed Distance Functions*, in *Level Set Methods and Dynamic Implicit Surfaces*, S. Osher and R. Fedkiw, Editors. 2003, Springer New York: New York, NY. p. 17-22.

127. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. and Jackel, L., *Handwritten digit recognition with a back-propagation network.* Advances in neural information processing systems, 1989.

128. Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Van Esesn, B.C., Awwal, A.A.S. and Asari, V.K., *The history began from alexnet: A comprehensive survey on deep learning approaches.* 2018.

129. Nex, F., Duarte, D., Tonolo, F.G. and Kerle, N, *Structural building damage detection with deep learning: Assessment of a state-of-the-art CNN in operational conditions.* Remote sensing, 2019.

130. Su, Y., Mao, C., Jiang, R., Liu, G. and Wang, J., *Data-driven fire safety management at building construction sites: Leveraging CNN.* Journal of management in engineering, 2021.

131. Sharma, D., et al. *DANIEL: A Deep Architecture for Automatic Analysis and Retrieval of Building Floor Plans*. in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 2017.

132. Boulila, W., et al., *A novel CNN-LSTM-based approach to predict urban expansion.* Ecological Informatics, 2021. **64**: p. 101325.

133. Bambharolia, P., *Overview of Convolutional Neural Networks.* Proceedings of the International Conference on Academic Research in Engineering and Management, 2017.

134. Wu, J., *Introduction to convolutional neural networks.* National Key Lab for Novel Software Technology, 2017.

135. Markov, A.A., *An Example of Statistical Investigation of the Text Eugene Onegin Concerning the Connection of Samples in Chains.* Science in Context, 2006. **19**(4): p. 591-600.

136. Ramya, C., Kavitha, G. and Shreedhara, D.K., *Recalling of images using Hopfield neural network model.* 2011.

137. Hochreiter, S. and J. Schmidhuber, *Long Short-Term Memory.* Neural Computation, 1997. **9**(8): p. 1735-1780.

138. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., *Attention is all you need.* Advances in neural information processing systems, 2017.

139. Wang, S., Tu, Z., Tan, Z., Wang, W., Sun, M. and Liu, Y., *Language models are good translators.* 2021.

140. Aswani, S., Choudhary, K., Shetty, S. and Nur, N., *Automatic text summarization of scientific articles using transformers—A brief review.* Journal of Autonomous Intelligence, 2024.

141. Cappellazzo, U., et al. *Large Language Models are Strong Audio-Visual Speech Recognition Learners*. in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2025.

142. He, S., Liao, W., Tavakoli, H.R., Yang, M., Rosenhahn, B. and Pugeault, N., *Image captioning through image transformer.* Proceedings of the Asian conference on computer vision., 2020.

143. Bansal, G., et al., *Transforming Conversations with AI—A Comprehensive Study of ChatGPT.* Cognitive Computation, 2024. **16**(5): p. 2487-2510.

144. Ghaemi, H., et al., *Transformers in source code generation: A comprehensive survey.* Journal of Systems Architecture, 2024. **153**: p. 103193.

145. Uparkar, O., et al., *Vision Transformer Outperforms Deep Convolutional Neural Network-based Model in Classifying X-ray Images.* Procedia Computer Science, 2023. **218**: p. 2338-2349.

146. Mittal, P., B. Sharma, and D.P. Yadav. *Comparative Analysis between CNN and ViT using Brain MRI Dataset*. in *2024 Eighth International Conference on Parallel, Distributed and Grid Computing (PDGC)*. 2024.

147. Zhai, X., Kolesnikov, A., Houlsby, N. and Beyer, L., *Scaling vision transformers.* Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022.

148. Liu, L., Liu, J. and Han, J., *Multi-head or single-head? an empirical comparison for transformer training.* 2021.

149. Cui, Y., Ren, J., He, P., Liu, H., Tang, J. and Xing, Y., *Superiority of Multi-Head Attention: A Theoretical Study in Shallow Transformers in In-Context Linear Regression.* The 28th International Conference on Artificial Intelligence and Statistics., 2025.

150. Feldman, R.M., Valdez-Flores, C., Feldman, R.M. and Valdez-Flores, C., *Markov Processes.* Applied Probability and Stochastic Processes, 2010.

151. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., *Proximal policy optimization algorithms.* 2017.

152. Renkavieski, C. and R.S. Parpinelli, *Meta-heuristic algorithms to truss optimization: Literature mapping and application.* Expert Systems with Applications, 2021. **182**: p. 115197.

153. Das, S. and P.N. Suganthan, *Differential Evolution: A Survey of the State-of-the-Art.* IEEE Transactions on Evolutionary Computation, 2011. **15**(1): p. 4-31.

154. Ali, M.M. and A. Törn, *Population set-based global optimization algorithms: some modifications and numerical studies.* Computers & Operations Research, 2004. **31**(10): p. 1703-1725.

155. Saruhan, H., *Differential evolution and simulated annealing algorithms for mechanical systems design.* Engineering Science and Technology, an International Journal, 2014. **17**(3): p. 131-136.

156. Pattanaik, J.K., M. Basu, and D.P. Dash, *Dynamic economic dispatch: a comparative study for differential evolution, particle swarm optimization, evolutionary programming, genetic algorithm, and simulated annealing.* Journal of Electrical Systems and Information Technology, 2019. **6**(1): p. 1.

157. Opara, K.R. and J. Arabas, *Differential Evolution: A survey of theoretical analyses.* Swarm and Evolutionary Computation, 2019. **44**: p. 546-558.

158. Zhang, J. and A.C. Sanderson, *JADE: Adaptive Differential Evolution With Optional External Archive.* IEEE Transactions on Evolutionary Computation, 2009. **13**(5): p. 945-958.

159. Tanabe, R. and A.S. Fukunaga. *Improving the search performance of SHADE using linear population size reduction*. in *2014 IEEE Congress on Evolutionary Computation (CEC)*. 2014.

160. Kizilay, D., et al. *A Differential Evolution Algorithm with Q-Learning for Solving Engineering Design Problems*. in *2020 IEEE Congress on Evolutionary Computation (CEC)*. 2020.

161. Hu, Z., W. Gong, and S. Li, *Reinforcement learning-based differential evolution for parameters extraction of photovoltaic models.* Energy Reports, 2021. **7**: p. 916-928.

162. Wiele, T., Warde-Farley, D., Mnih, A. and Mnih, V., *Q-learning in enormous action spaces via amortized approximate maximization.* NeurIPS Workshop., 2018.

163. Mitropoulou, C.C., Fourkiotis, Y., Lagaros, N.D. and Karlaftis, M.G., *Evolution Strategies-Based.* Metaheuristic Applications in Structures and Infrastructures, 2013.

164. Koziel, S.a.Y., X.S. eds., *Computational optimization, methods and algorithms.* Springer, 2011.

165. Artar, M. and A.T. Daloglu, *Teaching Learning Based Optimum Design of Transmission Tower Structures*, in *Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications*, S. Carbas, A. Toktas, and D. Ustun, Editors. 2021, Springer Singapore: Singapore. p. 49-64.

166. Rardin, R.L. and R. Uzsoy, *Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial.* Journal of Heuristics, 2001. **7**(3): p. 261-304.

167. Beiranvand, V., W. Hare, and Y. Lucet, *Best practices for comparing optimization algorithms.* Optimization and Engineering, 2017. **18**(4): p. 815-848.

168. Zoph, B.a.L., Q.V., *Neural architecture search with reinforcement learning.* 2016.

169. Liu, Y., et al., *A Survey on Evolutionary Neural Architecture Search.* IEEE Transactions on Neural Networks and Learning Systems, 2023. **34**(2): p. 550-570.

170. Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B. and Xing, E.P., *Neural architecture search with bayesian optimisation and optimal transport.* Advances in neural information processing systems, 2018.

171. Akay, B., D. Karaboga, and R. Akay, *A comprehensive survey on optimizing deep learning models by metaheuristics.* Artificial Intelligence Review, 2022. **55**(2): p. 829-894.

172. Morales-Hernández, A., I. Van Nieuwenhuyse, and S. Rojas Gonzalez, *A survey on multi-objective hyperparameter optimization algorithms for machine learning.* Artificial Intelligence Review, 2023. **56**(8): p. 8043-8093.

173. Devikanniga, D., K. Vetrivel, and N. Badrinath, *Review of Meta-Heuristic Optimization based Artificial Neural Networks and its Applications.* Journal of Physics: Conference Series, 2019. **1362**(1): p. 012074.

174. Kit Po, W. and D. Zhao Yang. *Differential Evolution, an Alternative Approach to Evolutionary Algorithm*. in *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*. 2005.

175. Wang, B., et al. *A Hybrid Differential Evolution Approach to Designing Deep Convolutional Neural Networks for Image Classification*. in *AI 2018: Advances in Artificial Intelligence*. 2018. Cham: Springer International Publishing.

176. Zhao, X., Jiang, L., Slowik, A., Zhang, Z. and Xue, Y., *Evolving blocks by segmentation for neural architecture search.* Electronic Research Archive, 2024.

177. Zhang, Z., et al., *Sle-CNN: a novel convolutional neural network for sleep stage classification.* Neural Computing and Applications, 2023. **35**(23): p. 17201-17216.

178. Ahmed, M., H. Du, and A. AlZoubi, *ENAS-B: Combining ENAS With Bayesian Optimization for Automatic Design of Optimal CNN Architectures for Breast Lesion Classification From Ultrasound Images.* Ultrasonic Imaging, 2024. **46**(1): p. 17-28.

179. Wang, B., et al. *Evolving Deep Convolutional Neural Networks by Variable-Length Particle Swarm Optimization for Image Classification*. in *2018 IEEE Congress on Evolutionary Computation (CEC)*. 2018.

180. Ghosh, A., et al., *Designing optimal convolutional neural network architecture using differential evolution algorithm.* Patterns, 2022. **3**(9).

181. Xie, L.a.Y., A., *Genetic cnn.* Proceedings of the IEEE international conference on computer vision, 2017.

182. Sun, Y., et al., *Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification.* IEEE Transactions on Cybernetics, 2020. **50**(9): p. 3840-3854.

183. Ramzan, Z., et al., *A Multimodal Data Fusion and Deep Neural Networks Based Technique for Tea Yield Estimation in Pakistan Using Satellite Imagery.* IEEE Access, 2023. **11**: p. 42578-42594.

184. Liang, X., Fu, P., Guo, Q., Zheng, K. and Qian, Y., *Dc-nas: Divide-and-conquer neural architecture search for multi-modal classification.* Proceedings of the AAAI conference on artificial intelligence (Vol. 38, No. 12, pp. 13754-13762). 2024.

185. Vidnerová, P. and J. Kalina. *Multi-objective Bayesian Optimization for Neural Architecture Search*. in *Artificial Intelligence and Soft Computing*. 2023. Cham: Springer International Publishing.

186. Agarwal, M., S.K. Gupta, and K.K. Biswas, *Genetic algorithm based approach to compress and accelerate the trained Convolution Neural Network model.* International Journal of Machine Learning and Cybernetics, 2023. **14**(7): p. 2367-2383.

187. Tahir, A.B.T., Khan, M.A., Alhaisoni, M., Khan, J.A. and Javed, K., *Deep Learning and Improved Particle Swarm Optimization Based Multimodal Brain Tumor Classification.* Computers, Materials & Continua, 2021.

188. Mikolov, T., Chen, K., Corrado, G. and Dean, J., *Efficient estimation of word representations in vector space.* 2013.

189. Gao, M., T. Li, and P. Huang. *Text Classification Research Based on Improved Word2vec and CNN*. in *Service-Oriented Computing – ICSOC 2018 Workshops*. 2019. Cham: Springer International Publishing.

190. Hong, S., et al., *Screening ideas in the early stages of technology development: A word2vec and convolutional neural network approach.* Technovation, 2022. **112**: p. 102407.

191. Song, P., C. Geng, and Z. Li, *Research on Text Classification Based on Convolutional Neural Network*. 2019. 229-232.

192. Guo, S. and N. Yao, *Generating word and document matrix representations for document classification.* Neural Computing and Applications, 2020. **32**(14): p. 10087-10108.

193. Jang, B., I. Kim, and J.W. Kim, *Word2vec convolutional neural networks for classification of news articles and tweets.* PloS one, 2019. **14**(8): p. e0220976.

194. Kale, A.S., F. Di Troia, and M. Stamp, *Malware classification with word embedding features.* arXiv preprint arXiv:2103.02711, 2021.

195. Tuba, E., et al., *Convolutional neural networks hyperparameters tuning*, in *Artificial intelligence: theory and applications*. 2021, Springer. p. 65-84.

196. Awad, N., N. Mallik, and F. Hutter, *Differential evolution for neural architecture search.* arXiv preprint arXiv:2012.06400, 2020.

197. Cantú, V.H., C. Azzaro-Pantel, and A. Ponsich, *Constraint-handling techniques within differential evolution for solving process engineering problems.* Applied Soft Computing, 2021. **108**: p. 107442.

198. Prechelt, L., *Early stopping-but when?*, in *Neural Networks: Tricks of the trade*. 2002, Springer. p. 55-69.

199. Mazari, A.C., N. Boudoukhani, and A. Djeffal, *BERT-based ensemble learning for multi-aspect hate speech detection.* Cluster Computing, 2024. **27**(1): p. 325-339.

200. Koshiry, A.M.E., et al., *Arabic toxic tweet classification: leveraging the AraBERT model.* Big Data and Cognitive Computing, 2023. **7**(4): p. 170.

201. Lan, Z., et al., *Albert: A lite bert for self-supervised learning of language representations.* arXiv preprint arXiv:1909.11942, 2019.

202. Chen, X., et al., *Microsoft coco captions: Data collection and evaluation server.* arXiv preprint arXiv:1504.00325, 2015.

203. Montminy, D.P., et al., *Improving cross-device attacks using zero-mean unit-variance normalization.* Journal of Cryptographic Engineering, 2013. **3**(2): p. 99-110.

204. Xu, M., et al., *A comprehensive survey of image augmentation techniques for deep learning.* Pattern Recognition, 2023. **137**: p. 109347.

205. Nam, J., et al. *Large-scale multi-label text classification—revisiting neural networks*. in *Joint european conference on machine learning and knowledge discovery in databases*. 2014. Springer.

206. Fujino, A., H. Isozaki, and J. Suzuki. *Multi-label text categorization with model combination based on f1-score maximization*. in *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*. 2008.

207. Ghamrawi, N. and A. McCallum. *Collective multi-label classification*. in *Proceedings of the 14th ACM international conference on Information and knowledge management*. 2005.

208. Zhang, L., C.P. Lim, and Y. Yu, *Intelligent human action recognition using an ensemble model of evolving deep networks with swarm-based optimization.* Knowledge-based systems, 2021. **220**: p. 106918.

209. Shams, M.Y., T. Abd El-Hafeez, and E. Hassan, *Acoustic data detection in large-scale emergency vehicle sirens and road noise dataset.* Expert Systems with Applications, 2024. **249**: p. 123608.

210. Zhang, X., et al., *A cost-sensitive attention temporal convolutional network based on adaptive top-k differential evolution for imbalanced time-series classification.* Expert Systems with Applications, 2023. **213**: p. 119073.

211. Zhang, J. and A.C. Sanderson, *Adaptive differential evolution*. 2009: Springer.

212. Qin, A.K. and P.N. Suganthan. *Self-adaptive differential evolution algorithm for numerical optimization*. in *2005 IEEE congress on evolutionary computation*. 2005. IEEE.

213. Tanabe, R. and A. Fukunaga. *Success-history based parameter adaptation for differential evolution*. in *2013 IEEE congress on evolutionary computation*. 2013. IEEE.

214. Tsafarakis, S., et al., *Fuzzy self-tuning differential evolution for optimal product line design.* European Journal of Operational Research, 2020. **287**(3): p. 1161-1169.

215. Zhang, B. and P. Wonka, *Training data generating networks: Shape reconstruction via bi-level optimization.* arXiv preprint arXiv:2010.08276, 2020.

216. Qu, C., et al. *BERT with history answer embedding for conversational question answering*. in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 2019.

217. Wang, Z., et al., *Multi-passage BERT: A globally normalized BERT model for open-domain question answering.* arXiv preprint arXiv:1908.08167, 2019.

218. Bilal, M. and A.A. Almazroi, *Effectiveness of fine-tuned BERT model in classification of helpful and unhelpful online customer reviews.* Electronic Commerce Research, 2023. **23**(4): p. 2737-2757.

219. Adhikari, A., et al., *Docbert: Bert for document classification.* arXiv preprint arXiv:1904.08398, 2019.

220. Hu, W., et al., *NLIRE: A natural language inference method for relation extraction.* Journal of Web Semantics, 2022. **72**: p. 100686.

221. Jiang, N. and M.-C. de Marneffe. *Evaluating BERT for natural language inference: A case study on the CommitmentBank*. in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 2019.

222. Devlin, J., et al. *Bert: Pre-training of deep bidirectional transformers for language understanding*. in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019.

223. Lagaros, N.D., M. Papadrakakis, and G. Kokossalakis, *Structural optimization using evolutionary algorithms.* Computers & structures, 2002. **80**(7-8): p. 571-589.

224. Georgioudakis, M. and V. Plevris, *A comparative study of differential evolution variants in constrained structural optimization.* Frontiers in Built Environment, 2020. **6**: p. 102.

225. Perez, R.E. and K. Behdinan, *Particle swarm optimization in structural design.* Swarm intelligence: Focus on ant and particle swarm optimization, 2007. **532**.

226. Rozvany, G.I., *A critical review of established methods of structural topology optimization.* Structural and multidisciplinary optimization, 2009. **37**(3): p. 217-237.

227. Wang, A.Q., A.V. Dalca, and M.R. Sabuncu, *Computing multiple image reconstructions with a single hypernetwork.* arXiv preprint arXiv:2202.11009, 2022.

228. Chen, Z. and H. Zhang. *Learning implicit fields for generative shape modeling*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

229. Yariv, L., et al., *Volume rendering of neural implicit surfaces.* Advances in neural information processing systems, 2021. **34**: p. 4805-4815.

230. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S. and Geiger, A., *Occupancy networks: Learning 3d reconstruction in function space.* Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.

231. Park, J.J., Florence, P., Straub, J., Newcombe, R. and Lovegrove, S., *Deepsdf: Learning continuous signed distance functions for shape representation.* Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.

232. Sitzmann, V., Chan, E., Tucker, R., Snavely, N. and Wetzstein, G., *Metasdf: Meta-learning signed distance functions.* Advances in Neural Information Processing Systems, 2020.

233. Lei, J., K. Jia, and Y. Ma, *Learning and Meshing From Deep Implicit Surface Networks Using an Efficient Implementation of Analytic Marching.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. **44**(12): p. 10068-10086.

234. Newman, T.S. and H. Yi, *A survey of the marching cubes algorithm.* Computers & Graphics, 2006. **30**(5): p. 854-879.

235. Wang, J., et al., *Three-dimensional Reconstruction of Jaw and Dentition CBCT Images Based on Improved Marching Cubes Algorithm.* Procedia CIRP, 2020. **89**: p. 239-244.

236. Khetan, A.a.K., Z., *schuBERT: Optimizing elements of BERT.* 2020.

237. Takikawa, T., Litalien, J., Yin, K., Kreis, K., Loop, C., Nowrouzezahrai, D., Jacobson, A., McGuire, M. and Fidler, S., *Neural geometric level of detail: Real-time rendering with implicit 3d shapes.* Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021.

238. Proszewska, M., Mazur, M., Trzciński, T. and Spurek, P., *Hypercube: Implicit field representations of voxelized 3d models.* 2021.

239. Kurz, F., et al. *Deep Learning Segmentation and 3D Reconstruction of Road Markings Using Multiview Aerial Imagery*. ISPRS International Journal of Geo-Information, 2019. **8**, DOI: 10.3390/ijgi8010047.

240. Frank, N., et al. *Comparing Vision-based to Sonar-based 3D Reconstruction*. in *2020 IEEE International Conference on Computational Photography (ICCP)*. 2020.

241. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T.. *Pix3d: Dataset and methods for single-image 3d shape modeling.* Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.

242. Yang, L., Li, B., Li, W., Jiang, B. and Xiao, J., *Semantic metric 3d reconstruction for concrete inspection.* Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2018.

243. Li, Y.D., Kuang, B. and Liu, J., *SIMP-based evolutionary structural optimization method for topology optimization.* Applied Mechanics and Materials, 2014.

244. Sigmund, O., *A 99 line topology optimization code written in Matlab.* Structural and Multidisciplinary Optimization, 2001. **21**(2): p. 120-127.

245. Andreassen, E., et al., *Efficient topology optimization in MATLAB using 88 lines of code.* Structural and Multidisciplinary Optimization, 2011. **43**(1): p. 1-16.

246. Pettersson, D.a.S., E., *PyTOpt: A Nonlinear Topology Optimisation Program in Python.* 2021.

247. Preisinger, C., *Linking Structure and Parametric Geometry.* Architectural Design, 2013. **83**(2): p. 110-113.

248. Achtziger, W. and M. Stolpe, *Truss topology optimization with discrete design variables—Guaranteed global optimality and benchmark examples.* Structural and Multidisciplinary Optimization, 2007. **34**(1): p. 1-20.

249. Avcı, M.S., et al., *Efficient Sizing and Layout Optimization of Truss Benchmark Structures Using ISRES Algorithm.* Applied Sciences, 2024. **14**(8): p. 3324.

250. Savsani, V., Tejani, G. and Patel, V., *Truss Optimization.* 2024.

251. Yang, H., R. Tam, and X. Tang. *Whole-Heart Reconstruction with Explicit Topology Integrated Learning*. in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*. 2023. Cham: Springer Nature Switzerland.

252. Ma, L., et al., *Simulation of Postoperative Facial Appearances via Geometric Deep Learning for Efficient Orthognathic Surgical Planning.* IEEE Transactions on Medical Imaging, 2023. **42**(2): p. 336-345.

253. Wu, T., Pan, L., Zhang, J., Wang, T., Liu, Z. and Lin, D., *Density-aware chamfer distance as a comprehensive metric for point cloud completion.* 2021.

254. Zeng, R., et al., *CD2: Fine-grained 3D Mesh Reconstruction with Twice Chamfer Distance.* ACM Trans. Multimedia Comput. Commun. Appl., 2023. **19**(6): p. Article 181.

255. Wagner, N.a.S., U., *NeuralQAAD: An Efficient Differentiable Framework for High Resolution Point Cloud Compression.* 2020.

256. Yi, L., Shao, L., Savva, M., Huang, H., Zhou, Y., Wang, Q., Graham, B., Engelcke, M., Klokov, R., Lempitsky, V. and Gan, Y., *Large-scale 3d shape reconstruction and segmentation from shapenet core55.* 2017.

257. Chauhan, V.K., et al., *A brief review of hypernetworks in deep learning.* Artificial Intelligence Review, 2024. **57**(9): p. 250.

258. Beck, J., et al., *Hypernetworks in Meta-Reinforcement Learning*, in *Proceedings of The 6th Conference on Robot Learning*, L. Karen, K. Dana, and I. Jeff, Editors. 2023, PMLR: Proceedings of Machine Learning Research. p. 1478--1487.

259. Chandra, D.S., Varshney, S., Srijith, P.K. and Gupta, S., *Continual learning with dependency preserving hypernetworks.* Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2023.

260. Alaluf, Y., Tov, O., Mokady, R., Gal, R. and Bermano, A., *Hyperstyle: Stylegan inversion with hypernetworks for real image editing.* Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition, 2022.

261. Sun, Z., Ozay, M. and Okatani, T., *Hypernetworks with statistical filtering for defending adversarial examples.* 2017.

262. Lorraine, J.a.D., D., *Stochastic hyperparameter optimization through hypernetworks.* 2018.

263. Höfer, T., Kiefer, B., Messmer, M. and Zell, A., *Hyperposepdf-hypernetworks predicting the probability distribution on so (3).* Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023.

264. Ruta, D., et al. *HyperNST: Hyper-Networks for Neural Style Transfer*. in *Computer Vision – ECCV 2022 Workshops*. 2023. Cham: Springer Nature Switzerland.

265. Huang, Y., et al. *Continual Model-Based Reinforcement Learning with Hypernetworks*. in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021.

266. Szatkowski, F., Piczak, K.J., Spurek, P., Tabor, J. and Trzciński, T., *Hypernetworks build implicit neural representations of sounds.* . Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2023.

267. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H. and Xiao, J., *Shapenet: An information-rich 3d model repository.* 2015.

268. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J. and Su, H., *Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding.* Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.

269. Spurek, P., Zięba, M., Tabor, J. and Trzciński, T., *Hyperflow: Representing 3d objects as surfaces.* 2020.

270. Gao, J., Chen, W., Xiang, T., Jacobson, A., McGuire, M. and Fidler, S., *Learning deformable tetrahedral meshes for 3d reconstruction.* Advances in neural information processing systems, 2020.

271. Wang, Z., Liu, Y. and Lu, F., *Cloud Sphere: A 3D Shape Representation via Progressive Deformation.* 2021.

272. Jaramillo, P. and I. Sipiran. *Cultural Heritage 3D Reconstruction with Diffusion Networks*. in *Computer Vision – ECCV 2024 Workshops*. 2025. Cham: Springer Nature Switzerland.

273. Lin, K., Wang, L. and Liu, Z., *End-to-end human pose and mesh reconstruction with transformers.* Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021.

274. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J. and Lu, J., *Point-bert: Pre-training 3d point cloud transformers with masked point modeling.* Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022.

275. Fournier, Q., G.M. Caron, and D. Aloise, *A Practical Survey on Faster and Lighter Transformers.* ACM Comput. Surv., 2023. **55**(14s): p. Article 304.

276. Shen, H., Zhang, J., Xiong, B., Hu, R., Chen, S., Wan, Z., Wang, X., Zhang, Y., Gong, Z., Bao, G. and Tao, C., *Efficient diffusion models: A survey.* 2025.

277. Peng, K., Islam, R., Quarles, J. and Desai, K., *Tmvnet: Using transformers for multi-view voxel-based 3d reconstruction.* Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022.

278. Shi, Z., Meng, Z., Xing, Y., Ma, Y. and Wattenhofer, R., *3d-retr: End-to-end single and multi-view 3d reconstruction with transformers.* 2021.

279. Wiesner, D., et al. *Implicit Neural Representations for Generative Modeling of Living Cell Shapes*. in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*. 2022. Cham: Springer Nature Switzerland.

280. Jack, D., et al. *Learning Free-Form Deformations for 3D Object Reconstruction*. in *Computer Vision – ACCV 2018*. 2019. Cham: Springer International Publishing.

281. Ping, G. and H. Wang. *3D Reconstruction from A Single Image*. in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. 2019.

282. Henrich, P., Gyenes, B., Scheikl, P.M., Neumann, G. and Mathis-Ullrich, F., *Registered and segmented deformable object reconstruction from a single view point cloud.* Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024.

283. Pan, J., et al. *Residual MeshNet: Learning to Deform Meshes for Single-View 3D Reconstruction*. in *2018 International Conference on 3D Vision (3DV)*. 2018.

284. Spurek, P., et al., *General Hypernetwork Framework for Creating 3D Point Clouds.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. **44**(12): p. 9995-10008.

285. Batorski, P., et al., *Hyperplanes: Hypernetwork approach to rapid nerf adaptation.* arXiv preprint arXiv:2402.01524, 2024.

286. Gu, J. and S. Yeung-Levy, *Foundation models secretly understand neural network weights: Enhancing hypernetwork architectures with foundation models.* arXiv preprint arXiv:2503.00838, 2025.

287. Ma, X., et al. *Generating 3D House Wireframes with Semantics*. in *Computer Vision – ECCV 2024*. 2025. Cham: Springer Nature Switzerland.

288. Nauata, N., et al. *House-GAN: Relational Generative Adversarial Networks for Graph-Constrained House Layout Generation*. in *Computer Vision – ECCV 2020*. 2020. Cham: Springer International Publishing.

289. He, Z., Y.-H. Wang, and J. Zhang, *Generative AIBIM: An automatic and intelligent structural design pipeline integrating BIM and generative AI.* Information Fusion, 2025. **114**: p. 102654.

290. Trevithick, A.a.Y., B., *Grf: Learning a general radiance field for 3d representation and rendering.* Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021.

291. Huang, H., et al., *Learning to learn point signature for 3D shape geometry.* Pattern Recognition Letters, 2024. **184**: p. 140-147.

292. Ma, B., Han, Z., Liu, Y.S. and Zwicker, M., *Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces.* 2020.

293. Ju, Y., et al., *Adaptive grid generation for discretizing implicit complexes.* ACM Trans. Graph., 2024. **43**(4): p. Article 82.

294. Zhang, J., Yang, G., Tulsiani, S. and Ramanan, D., *Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild.* Advances in Neural Information Processing Systems, 2021.

295. Li, R., et al., *SP-GAN: Sphere-guided 3D shape generation and manipulation.* ACM Transactions on Graphics (TOG), 2021. **40**(4): p. 1-12.

296. González-Carvajal, S. and E.C. Garrido-Merchán, *Comparing BERT against traditional machine learning text classification.* arXiv preprint arXiv:2005.13012, 2020.

297. Cariello, M.C., A. Lenci, and R. Mitkov. *A comparison between named entity recognition models in the biomedical domain*. in *Proceedings of the Translation and Interpreting Technology Online Conference*. 2021.

298. Nassiri, K. and M. Akhloufi, *Transformer models used for text-based question answering systems.* Applied Intelligence, 2023. **53**(9): p. 10602-10635.

299. Koroteev, M.V., *BERT: a review of applications in natural language processing and understanding.* arXiv preprint arXiv:2103.11943, 2021.

300. Zhao, X., et al. *Fine-Tuning BERT Model for Materials Named Entity Recognition*. in *2021 IEEE International Conference on Big Data (Big Data)*. 2021.

301. Müller, A., C. Curino, and R. Ramakrishnan, *Mothernet: A foundational hypernetwork for tabular classification.* arXiv preprint arXiv:2312.08598, 2023.

302. Lell, N. and A. Scherp. *HyperAggregation: Aggregating over Graph Edges with Hypernetworks*. in *2024 International Joint Conference on Neural Networks (IJCNN)*. 2024.

303. Mai, F., et al., *Hypermixer: An mlp-based low cost alternative to transformers.* arXiv preprint arXiv:2203.03691, 2022.

304. Hemati, H., et al. *Partial hypernetworks for continual learning*. in *Conference on Lifelong Learning Agents*. 2023. PMLR.

305. Dubey, S.R., S.K. Singh, and B.B. Chaudhuri, *Activation functions in deep learning: A comprehensive survey and benchmark.* Neurocomputing, 2022. **503**: p. 92-108.

306. Oleynikova, H., et al. *Signed distance fields: A natural representation for both mapping and planning*. in *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*. 2016. University of Michigan.

307. Chen, X., L. Zhao, and D. Zou, *How transformers utilize multi-head attention in in-context learning? a case study on sparse linear regression.* Advances in Neural Information Processing Systems, 2024. **37**: p. 119573-119613.

308. Oyedotun, O.K., K.A. Ismaeil, and D. Aouada, *Why Is Everyone Training Very Deep Neural Network With Skip Connections?* IEEE Transactions on Neural Networks and Learning Systems, 2023. **34**(9): p. 5961-5975.

309. Zhang, J., et al., *Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild.* Advances in Neural Information Processing Systems, 2021. **34**: p. 29835-29847.

310. Yu, Y., et al., *Identifying Irregular Potatoes Using Hausdorff Distance and Intersection over Union.* Sensors, 2022. **22**(15): p. 5740.

311. Masala, G.L., B. Golosio, and P. Oliva, *An improved Marching Cube algorithm for 3D data segmentation.* Computer Physics Communications, 2013. **184**(3): p. 777-782.

312. Perng, K.-L., et al. *A real-time 3D virtual sculpting tool based on modified marching cubes*. in *Proceedings of International Conference on Artificial Reality and Teleexistence*. 2001.

313. Nugroho, P.A., D.K. Basuki, and R. Sigit. *3D heart image reconstruction and visualization with marching cubes algorithm*. in *2016 International Conference on Knowledge Creation and Intelligent Computing (KCIC)*. 2016.

314. Kim, Y., et al., *Robust generation of the delta volume for the damaged area of a part using the marching cubes algorithm to support additive manufacturing–based part maintenance.* The International Journal of Advanced Manufacturing Technology, 2021. **117**(5): p. 1473-1489.

315. Kim, S., D. Sohn, and S. Im, *Construction of polyhedral finite element meshes based upon marching cube algorithm.* Advances in Engineering Software, 2019. **128**: p. 98-112.

316. Westfalen, L., *Procedural Generation of Buildings with Wave Function Collapse and Marching Cubes*. 2024, Hochschule für Angewandte Wissenschaften Hamburg.

317. Xie, J., L. Sun, and Y.F. Zhao, *On the data quality and imbalance in machine learning-based design and manufacturing—A systematic review.* Engineering, 2025. **45**(2): p. 105-131.

318. Wang, Y., et al., *Characterizing data sharing in civil infrastructure engineering: Current practice, future vision, barriers, and promotion strategies.* Journal of Computing in Civil Engineering, 2023. **37**(2): p. 04023001.

319. Garozzo, R., et al., *CulTO: An ontology-based annotation tool for data curation in cultural heritage.* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2017. **42**: p. 267-274.

320. Kandelan, S.N., et al., *Geometric data in urban building energy modeling: Current practices and the case for automation.* Journal of Building Engineering, 2024. **97**: p. 110836.

321. Yetiş, G., et al. *A novel approach for classification of structural elements in a 3d model by supervised learning*. in *Proceedings of the 36th eCAADe Conference*. 2018.

322. Andreou, A., Kontovourkis, O., Solomou, S. and Savvides, A., *Rethinking architectural design process using integrated parametric design and machine learning principles.* Rethinking architectural design process using integrated parametric design and machine learning principles, 2023.

323. Wefki, H., M. Elnahla, and E. Elbeltagi, *BIM-based schedule generation and optimization using genetic algorithms.* Automation in Construction, 2024. **164**: p. 105476.

324. Park, J., et al., *Auxetic pattern design for concentric-tube robots using an active DNN-metaheuristics optimization.* Thin-Walled Structures, 2024. **197**: p. 111603.

325. Srivastava, P.R., et al. *Generation of test data using meta heuristic approach*. in *TENCON 2008 - 2008 IEEE Region 10 Conference*. 2008.

326. Vadyala, S.R., et al., *A review of physics-based machine learning in civil engineering.* Results in Engineering, 2022. **13**: p. 100316.

327. Weber, B., et al. *Interactive geometric simulation of 4d cities*. in *Computer Graphics Forum*. 2009. Wiley Online Library.

328. Hu, X. and K. Liu, *Structural deterioration knowledge ontology towards physics-informed machine learning for enhanced bridge deterioration prediction.* Journal of Computing in Civil Engineering, 2023. **37**(1): p. 04022051.

329. Tang, S., et al., *High-resolution volumetric reconstruction for clothed humans.* ACM Transactions on Graphics, 2023. **42**(5): p. 1-15.

330. Cruz, R.S., et al. *DeepCSR: A 3D deep learning approach for cortical surface reconstruction*. in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021.

331. Zhao, M., et al., *3D-RVP: A method for 3D object reconstruction from a single depth view using voxel and point.* Neurocomputing, 2021. **430**: p. 94-103.

332. Xie, H., et al. *Pix2vox: Context-aware 3d reconstruction from single and multi-view images*. in *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.

333. Balakrishnan, A., S. S, and L. Shine, *Refine3DNet: Scaling Precision in 3D Object Reconstruction from Multi-View RGB Images using Attention✱*, in *Proceedings of the Fifteenth Indian Conference on Computer Vision Graphics and Image Processing*. 2025, Association for Computing Machinery. p. Article 42.

334. Wang, S., et al., *Single Target SAR 3D Reconstruction Based on Deep Learning.* Sensors, 2021. **21**(3): p. 964.

335. Borkowski, A.S., *File Hygiene and BIM Models Restrictions,".* Trends in Civil Engineering and its Architecture, 2019. **3**(3): p. 430-435.

336. Lin, C.H., Wang, C. and Lucey, S., *Sdf-srn: Learning signed distance 3d object reconstruction from static images.* Advances in Neural Information Processing Systems, 2020.

337. Zheng, Y., et al. *Hyperdet3d: Learning a scene-conditioned 3d object detector*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

338. Wang, K., et al., *Variable Radiance Field for Real-Life Category-Specifc Reconstruction from Single Image.* arXiv preprint arXiv:2306.05145, 2023.

339. Płoszaj-Mazurek, M., E. Ryńska, and M. Grochulska-Salak, *Methods to Optimize Carbon Footprint of Buildings in Regenerative Architectural Design with the Use of Machine Learning, Convolutional Neural Network, and Parametric Design.* Energies, 2020. **13**(20): p. 5289.

340. Shaghaghian, Z. and W. Yan, *Application of deep learning in generating desired design options: experiments using synthetic training dataset.* arXiv preprint arXiv:2001.05849, 2019.

341. Lu, Y., et al., *Multi-Objective Optimization of Building Environmental Performance: An Integrated Parametric Design Method Based on Machine Learning Approaches.* Energies, 2022. **15**(19): p. 7031.

342. Labib, R. *Integrating machine learning with parametric modeling environments to predict building daylighting performance*. in *IOP Conference Series: Earth and Environmental Science*. 2022. IOP Publishing.

343. Selvaraju, P., et al. *Buildingnet: Learning to label 3d buildings*. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

344. Eftekhar, A., et al. *Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans*. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

345. Che, Y., et al., *3D-GloBFP: the first global three-dimensional building footprint dataset.* Earth Syst. Sci. Data, 2024. **16**(11): p. 5357-5374.

346. Pepe, M., et al., *From Point Cloud to BIM: A New Method Based on Efficient Point Cloud Simplification by Geometric Feature Analysis and Building Parametric Objects in Rhinoceros/Grasshopper Software.* Remote Sensing, 2024. **16**(9): p. 1630.

347. Fu, H., et al. *3d-front: 3d furnished rooms with layouts and semantics*. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

348. Tochilkin, D., et al., *Triposr: Fast 3d object reconstruction from a single image.* arXiv preprint arXiv:2403.02151, 2024.

349. Watanabe, T., et al., *An infinite-resolution grid snapping technique based on fuzzy theory.* Applied Soft Computing, 2020. **89**: p. 106112.

350. Pakhale, K., *Comprehensive overview of named entity recognition: Models, domain-specific applications and challenges.* arXiv preprint arXiv:2309.14084, 2023.

351. Qiao, B., et al., *A joint model for entity and relation extraction based on BERT.* Neural Computing and Applications, 2022. **34**(5): p. 3471-3481.

352. Shen, Y. and J. Liu. *Comparison of Text Sentiment Analysis based on Bert and Word2vec*. in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*. 2021.

353. Aliero, A.A., Bashir, S.A., Aliyu, H.O., Tafida, A.G., Bashar, U.K. and Nasiru, M.D., *Systematic review on text normalization techniques and its approach to non-standard words.* 2023.

354. Jehangir, B., S. Radhakrishnan, and R. Agarwal, *A survey on named entity recognition—datasets, tools, and methodologies.* Natural Language Processing Journal, 2023. **3**: p. 100017.

355. Šteflovič, K. and J. Kapusta, *Coreference Resolution for Improving Performance Measures of Classification Tasks.* Applied Sciences, 2023. **13**(16): p. 9272.

356. Lee, D.-H., Z. Hu, and R.K.-W. Lee, *Improving text auto-completion with next phrase prediction.* arXiv preprint arXiv:2109.07067, 2021.

357. Pepi, C., M. Gioffrè, and M. Grigoriu, *Bayesian inference for parameters estimation using experimental data.* Probabilistic Engineering Mechanics, 2020. **60**: p. 103025.

358. Li, Y., L. Liu, and S. Shi, *Rethinking negative sampling for handling missing entity annotations.* arXiv preprint arXiv:2108.11607, 2021.

359. Kemp, R.L., *D for deterioration: The high cost of low infrastructure spending.* Nat'l Civic Rev., 2017.

360. Zhang, M.a.B., T., *Review on new spending of United States bipartisan infrastructure bill.* Journal of Infrastructure, Policy and Development, 2022.

361. Stewart, M.G., X. Wang, and M.N. Nguyen, *Climate change adaptation for corrosion control of concrete infrastructure.* Structural Safety, 2012. **35**: p. 29-39.

362. Chen, H.P., *Structural health monitoring of large civil engineering structures.* 2018.

363. Li, J.a.H., H., *A review of recent research advances on structural health monitoring in Western Australia.* Structural Monitoring and Maintenance, 2016.

364. Ko, J.M. and Y.Q. Ni, *Technology developments in structural health monitoring of large-scale bridges.* Engineering Structures, 2005. **27**(12): p. 1715-1725.

365. Bolton, A., Butler, L., Dabson, I., Enzer, M., Evans, M., Fenemore, T., Harradence, F., Keaney, E., Kemp, A., Luck, A. and Pawsey, N., *Gemini principles.* 2018.

366. Dang, H.V., M. Tatipamula, and H.X. Nguyen, *Cloud-Based Digital Twinning for Structural Health Monitoring Using Deep Learning.* IEEE Transactions on Industrial Informatics, 2022. **18**(6): p. 3820-3830.

367. Moi, T., A. Cibicik, and T. Rølvåg, *Digital twin based condition monitoring of a knuckle boom crane: An experimental study.* Engineering Failure Analysis, 2020. **112**: p. 104517.

368. Guo, Y., et al., *Fiber Bragg Grating Displacement Sensor with High Abrasion Resistance for a Steel Spring Floating Slab Damping Track.* Sensors, 2018. **18**(6): p. 1899.

369. Thakur, H.V., et al., *All-fiber embedded PM-PCF vibration sensor for Structural Health Monitoring of composite.* Sensors and Actuators A: Physical, 2011. **167**(2): p. 204-212.

370. Ye, C., Butler, L., Bartek, C., Iangurazov, M., Lu, Q., Gregory, A., Girolami, M. and Middleton, C., *A digital twin of bridges for structural health monitoring.* 12th International Workshop on Structural Health Monitoring, 2019.

371. Fedorov, A., et al., *Structural monitoring system with fiber Bragg grating sensors: implementation and software solution.* Journal of Physics: Conference Series, 2015. **594**(1): p. 012049.

372. Kreuzer, M., *Strain measurement with fiber Bragg grating sensors.* 2006.

373. Boateng, E.K.G., P. Schubel, and R. Umer, *Thermal isolation of FBG optical fibre sensors for composite cure monitoring.* Sensors and Actuators A: Physical, 2019. **287**: p. 158-167.

374. Sousa, H., Cavadas, F., Henriques, A., Bento, J. and Figueiras, J., *Bridge deflection evaluation using strain and rotation measurements.* Smart structures and systems, 2013.

375. Xu, H., W.-X. Ren, and Z.-C. Wang, *Deflection Estimation of Bending Beam Structures Using Fiber Bragg Grating Strain Sensors.* Advances in Structural Engineering, 2015. **18**(3): p. 395-403.

376. Figueiredo Filho, D.B., Júnior, J.A.S. and Rocha, E.C., *What is R2 all about?* Leviathan (São Paulo), 2011.

377. Sun, L., et al., *Strain Transfer Analysis of a Clamped Fiber Bragg Grating Sensor.* Applied Sciences, 2017. **7**(2): p. 188.

378. Battista, N.D., Kechavarzi, C., Cheal, N., Harvey, R. and Wong, S., *Monitoring the axial shortening of principal tower using embedded distributed fibre optic sensors.* International Conference on Smart Infrastructure and Construction 2019, 2019.

379. Webb, G.T., Vardanega, P.J., Hoult, N.A., Fidler, P.R.A., Bennett, P.J. and Middleton, C.R., *Analysis of fiber-optic strain-monitoring data from a prestressed concrete bridge.* Journal of Bridge Engineering, 2017.

380. Australia, S., *AS 5100.5-2017 Bridge design, Part 5: Concrete, 2nd ed.* SAI Global., 2017.

381. Sabih, G. and R. Tarefder, *Impact of variability of mechanical and thermal properties of concrete on predicted performance of jointed plain concrete pavements.* International Journal of Pavement Research and Technology, 2016. **9**.

382. Sarkar, S., et al. *Discrimination of Strain and Temperature effects on FBG-based Sensor using Machine Learning*. in *2020 IEEE Photonics Conference (IPC)*. 2020.

383. Aghabalaei Baghaei, K. and S.A. Hadigheh, *Durability assessment of FRP-to-concrete bonded connections under moisture condition using data-driven machine learning-based approaches.* Composite Structures, 2021: p. 114576.

384. Neupane, K. and S.A. Hadigheh, *Sodium hydroxide-free geopolymer binder for prestressed concrete applications.* Construction and Building Materials, 2021. **293**: p. 123397.

385. Moreno-Gomez, A., et al., *Sensors Used in Structural Health Monitoring.* Archives of Computational Methods in Engineering, 2018. **25**(4): p. 901-918.

386. Navabian, N., *Development of an automated structural health monitoring system based on wireless sensor network for civil structures.* 2020.

387. O'Shea, M. and J. Murphy, *Design of a BIM Integrated Structural Health Monitoring System for a Historic Offshore Lighthouse.* Buildings, 2020. **10**(7): p. 131.

388. Cremona, C.a.S., J., *Structural health monitoring as a big-data problem.* Structural Engineering International, 2018.

389. Farrar, C.R.a.W., K., *An introduction to structural health monitoring.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2007.

390. Guérineau, L., Mathé, L. and Le Roy, A., *Development of an integrated SHM framework based on a novel wireless sensing system for dynamic structural monitoring of bridges using ambient vibrations.* Proceedings of the 10th International Conference on Structural Health Monitoring of Intelligent Infrastructure, 2021.

391. Aktan, E., et al., *Lessons from Bridge Structural Health Monitoring (SHM) and Their Implications for the Development of Cyber-Physical Systems.* Infrastructures, 2024. **9**(2): p. 30.

392. Aktan, A.E.a.B., J.M.W., *Structural identification: opportunities and challenges. Journal of Structural Engineering.* Journal of Structural Engineering, 2013.

393. Cawley, P., *Structural health monitoring: Closing the gap between research and industrial deployment.* Structural Health Monitoring, 2018. **17**(5): p. 1225-1244.

394. Zonta, D., Glisic, B. and Adriaenssens, S., *Value of information: impact of monitoring on decision-making.* Structural Control and Health Monitoring, 2014.

395. Torres, B., et al., *Analysis of the strain transfer in a new FBG sensor for Structural Health Monitoring.* Engineering Structures, 2011. **33**(2): p. 539-548.

396. Shen, W., et al., *Application study on FBG sensor applied to hull structural health monitoring.* Optik, 2015. **126**(17): p. 1499-1504.

397. Braunfelds, J., Senkans, U., Skels, P., Janeliukstis, R., Salgals, T., Redka, D., Lyashuk, I., Porins, J., Spolitis, S., Haritonovs, V. and Bobrovs, V., *FBG-based sensing for structural health monitoring of road infrastructure.* Journal of Sensors, 2021.

398. Karatas, C., et al., *Fibre Bragg grating sensor applications for structural health monitoring.* Aircraft Engineering and Aerospace Technology, 2019. **92**(3): p. 355-367.

399. Ibrar Jahan, M.A., et al., *Deciphering the sensory landscape: a comparative analysis of fiber Bragg grating and strain gauge systems in structural health monitoring.* Journal of Optics, 2024.

400. Vandi, L., et al., *Structural Health Monitoring for Prefabricated Building Envelope under Stress Tests.* Applied Sciences, 2024. **14**(8): p. 3260.

401. Liu, Z., et al., *Sensor-Based Structural Health Monitoring of Asphalt Pavements with Semi-Rigid Bases Combining Accelerated Pavement Testing and a Falling Weight Deflectometer Test.* Sensors, 2024. **24**(3): p. 994.

402. Ho, M.-p., et al., *Structural health monitoring of an asymmetrical SMA reinforced composite using embedded FBG sensors.* Smart Materials and Structures, 2013. **22**(12): p. 125015.

403. Okagawa, S., P. Bernus, and O. Noran, *Realtime Health Monitoring of Composite Structures Using FBG Sensors.* IFAC-PapersOnLine, 2022. **55**(19): p. 157-162.

404. Amaya, A.a.S.-P., J., *Toward a structural health monitoring methodology for concrete structures under dynamic loads using embedded FBG sensors and strain mapping techniques.* Sensors, 2022.

405. AlHamaydeh, M.a.G.A., N., *Structural health monitoring techniques and technologies for large-scale structures: Challenges, limitations, and recommendations.* Practice Periodical on Structural Design and Construction, 2022.

406. Almohsen, A.S., *Challenges Facing the Use of Remote Sensing Technologies in the Construction Industry: A Review.* Buildings, 2024. **14**(9): p. 2861.

407. Murayama, H., D. Wada, and H. Igawa, *Structural health monitoring by using fiber-optic distributed strain sensors with high spatial resolution.* Photonic Sensors, 2013. **3**(4): p. 355-376.

408. Jayalakshmi, V. and A.R.M. Rao, *Simultaneous identification of damage and input dynamic force on the structure for structural health monitoring.* Structural and Multidisciplinary Optimization, 2017. **55**(6): p. 2211-2238.

409. Güemes, A., et al., *Structural Health Monitoring in Composite Structures by Fiber-Optic Sensors.* Sensors, 2018. **18**(4): p. 1094.

410. Wu, T., et al., *Recent Progress of Fiber-Optic Sensors for the Structural Health Monitoring of Civil Infrastructure.* Sensors, 2020. **20**(16): p. 4517.

411. Drissi-Habti, M. and V. Raman, *Fatigue Behavior of Smart Composites with Distributed Fiber Optic Sensors for Offshore Applications.* Journal of Composites Science, 2022. **6**(1): p. 2.

412. Leung, C.K.Y., et al., *A novel distributed optical crack sensor for concrete structures.* Engineering Fracture Mechanics, 2000. **65**(2): p. 133-148.

413. Zhao, J., et al., *Layout Optimization of FBG Sensor for Aluminum Alloy Beam Based on MOLA Multiobjective Optimization Algorithm.* IEEE Sensors Journal, 2023. **23**(13): p. 14129-14141.

414. Zhou, J., et al., *Efficient Sensor Placement Optimization for Shape Deformation Sensing of Antenna Structures with Fiber Bragg Grating Strain Sensors.* Sensors, 2018. **18**(8): p. 2481.

415. Lu, G., et al. *Optimal placement of FBG sensors for reconstruction of flexible plate structures using modal approach*. in *2015 34th Chinese Control Conference (CCC)*. 2015.

416. Mohammed, A., J.I. Melecio, and S. Djurović, *Stator Winding Fault Thermal Signature Monitoring and Analysis by In Situ FBG Sensors.* IEEE Transactions on Industrial Electronics, 2019. **66**(10): p. 8082-8092.

417. Zhou, Z., et al., *Optimal sensor placement and Bi-type response reconstruction for structural health monitoring using long-gauge FBG strain sensing network.* Structures, 2024. **63**: p. 106406.

418. Mishra, M., P.B. Lourenço, and G.V. Ramana, *Structural health monitoring of civil engineering structures by using the internet of things: A review.* Journal of Building Engineering, 2022. **48**: p. 103954.

419. Sofi, A., et al., *Structural health monitoring using wireless smart sensor network – An overview.* Mechanical Systems and Signal Processing, 2022. **163**: p. 108113.

420. Warsi, Z., et al., *Sensors for Structural Health Monitoring: A Review*. 2019. 1-6.

421. Biondi, A.M., et al., *Pipeline structural health monitoring using distributed fiber optic sensing textile.* Optical Fiber Technology, 2022. **70**: p. 102876.

422. Lu, R. and J. Judd, *Field-Deployable Fiber Optic Sensor System for Structural Health Monitoring of Steel Girder Highway Bridges.* Infrastructures, 2022. **7**(2): p. 16.

423. Marceddu, A.C., et al., *Comprehensive Visualization of Data Generated by Fiber Bragg Grating Sensors.* IEEE Access, 2023. **11**: p. 121945-121955.

424. Ding, G., et al., *Reconstruction of propeller deformation based on FBG sensor network.* Ocean Engineering, 2022. **249**: p. 110884.

425. Liu, M., et al., *Inverse Finite Element Method for Reconstruction of Deformation in the Gantry Structure of Heavy-Duty Machine Tool Using FBG Sensors.* Sensors, 2018. **18**(7): p. 2173.

426. Zhang, C., Ren, M. and Urtasun, R., *Graph hypernetworks for neural architecture search.* 2018.

427. Wullach, T., A. Adler, and E. Minkov, *Character-level HyperNetworks for Hate Speech Detection.* Expert Systems with Applications, 2022. **205**: p. 117571.

428. Chabra, R., et al. *Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction*. in *Computer Vision – ECCV 2020*. 2020. Cham: Springer International Publishing.

429. Zhang, L., et al., *Reference-free damage identification method for highway continuous girder bridges based on long-gauge fibre Bragg grating strain sensors.* Measurement, 2022. **195**: p. 111064.

430. Kashaganova, G., et al., *Research of a Fiber Sensor Based on Fiber Bragg Grating for Road Surface Monitoring.* Electronics, 2023. **12**(11): p. 2491.

431. Li, D.S., Ren, L., Li, H.N. and Song, G.B., *Structural health monitoring of a tall building during construction with fiber Bragg grating sensors.* International Journal of Distributed Sensor Networks, 2012.

432. Johnson Singh, M., et al., *Applications of fibre Bragg grating sensors for monitoring geotechnical structures: A comprehensive review.* Measurement, 2023. **218**: p. 113171.

433. Roveri, N., A. Carcaterra, and A. Sestieri, *Real-time monitoring of railway infrastructures using fibre Bragg grating sensors.* Mechanical Systems and Signal Processing, 2015. **60-61**: p. 14-28.

434. Tang, F., et al., *A review on fiber optic sensors for rebar corrosion monitoring in RC structures.* Construction and Building Materials, 2021. **313**: p. 125578.

435. Sun, Y.a.M., H., *Progress of fiber Bragg grating sensors in state perception of electrical equipment.* Measurement, 2023.

436. Gallagher, N.B., *Savitzky-Golay smoothing and differentiation filter.* Eigenvector Research Incorporated., 2020.

437. Sadeghi, M., F. Behnia, and R. Amiri, *Window Selection of the Savitzky–Golay Filters for Signal Recovery From Noisy Measurements.* IEEE Transactions on Instrumentation and Measurement, 2020. **69**(8): p. 5418-5427.

438. Malu, M., G. Dasarathy, and A. Spanias. *Bayesian Optimization in High-Dimensional Spaces: A Brief Survey*. in *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*. 2021.

439. Wang, X., et al., *Recent Advances in Bayesian Optimization.* ACM Comput. Surv., 2023. **55**(13s): p. Article 287.

440. Li, C.L., Kandasamy, K., Póczos, B. and Schneider, J., *High dimensional Bayesian optimization via restricted projection pursuit models.* Artificial Intelligence and Statistics, 2016.

441. Frazier, P.I., *A tutorial on Bayesian optimization.* 2018.

442. Schulz, E., M. Speekenbrink, and A. Krause, *A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions.* Journal of Mathematical Psychology, 2018. **85**: p. 1-16.

443. Hielscher, T., et al., *A neural network based digital twin model for the structural health monitoring of reinforced concrete bridges.* Structures, 2023. **57**: p. 105248.

444. Sitzmann, V., Martel, J., Bergman, A., Lindell, D. and Wetzstein, G., *Implicit neural representations with periodic activation functions.* Advances in neural information processing systems, 2020.

445. Duggal, S.a.P., D., *Topologically-aware deformation fields for single-view 3d reconstruction.* Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

446. Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M. and Adams, R., *Scalable bayesian optimization using deep neural networks* International conference on machine learning, 2015.

447. Ma, Z., J. Choi, and H. Sohn, *Structural displacement sensing techniques for civil infrastructure: A review.* Journal of Infrastructure Intelligence and Resilience, 2023. **2**(3): p. 100041.

448. Reddy, M.S.B., et al., *Sensors in advancing the capabilities of corrosion detection: A review.* Sensors and Actuators A: Physical, 2021. **332**: p. 113086.

449. Kral, Z., W. Horn, and J. Steck, *Crack Propagation Analysis Using Acoustic Emission Sensors for Structural Health Monitoring Systems.* The Scientific World Journal, 2013. **2013**(1): p. 823603.

450. Todd, M.D., et al., *Flexural beam-based fiber Bragg grating accelerometers.* IEEE Photonics Technology Letters, 1998. **10**(11): p. 1605-1607.

451. Farhan, K.Z., et al., *Temperature and humidity sensor technology for concrete health assessment: a review.* Innovative Infrastructure Solutions, 2023. **8**(10): p. 276.

452. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks.* Advances in neural information processing systems, 2012. **25**.

453. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

454. Zaeemzadeh, A., N. Rahnavard, and M. Shah, *Norm-preservation: Why residual networks can become extremely deep?* IEEE transactions on pattern analysis and machine intelligence, 2020. **43**(11): p. 3980-3990.

455. Szegedy, C., et al. *Going deeper with convolutions*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.