



**UNIVERSITY OF PATRAS
DEPARTMENT OF MECHANICAL ENGINEERING &
AERONAUTICS
DIVISION OF APPLIED MECHANICS, TECHNOLOGY
OF
MATERIALS & BIOMECHANICS
LABORATORY OF APPLIED MECHANICS &
VIBRATIONS**

DIPLOMA THESIS

**From Coupon to System: A Deep Learning and
Transfer Learning Framework for RUL Prediction in
Multi-Component Structures**

Vlachopoulou Evangelia
1093648

Supervising Professor: Loutas Theodoros

Diploma Thesis submitted to Department of Mechanical Engineering & Aeronautics at
University of Patras

Diploma Thesis
PATRA, JUNE 2026

University of Patras, Department of Mechanical Engineering & Aeronautics

Vlachopoulou Evangelia

© 2026 – All rights reserved



**UNIVERSITY OF PATRAS
DEPARTMENT OF MECHANICAL ENGINEERING &
AERONAUTICS
DIVISION OF APPLIED MECHANICS,
TECHNOLOGY OF
MATERIALS & BIOMECHANICS
LABORATORY OF APPLIED MECHANICS &
VIBRATIONS**

The present diploma thesis was presented
by

Vlachopoulou Evangelia

1093648

on the 6th of July 2026

The approval of the diploma thesis does not imply acceptance of the author's views. During the writing, the principles of academic ethics were observed.

ABSTRACT

From Coupon to System: A Deep Learning and Transfer Learning Framework for RUL Prediction in Multi-Component Structures

Vlachopoulou Evangelia

Prognostics health management (PHM) of metallic structures relies increasingly on data-driven methods for Remaining Useful Life (RUL) prediction yet the transition from coupon-level to system-level prognostics remains an open challenge, particularly under limited system-level data availability. Existing approaches typically address single-component degradation, yet a methodological gap remains in methods that can effectively leverage coupon-level knowledge to perform uncertainty-aware RUL prediction for interconnected multi-component systems. This paper proposes a transfer learning framework for system-level RUL prediction and uncertainty quantification of a three-coupon aluminum system, validated on experimental fatigue data comprising five coupons and six triplet specimens. A Genetic Algorithm (GA) is first employed to construct a health indicator from strain sensor measurements, providing a monotonic and informative input representation for the prognostic model. A Long Short-Term Memory (LSTM) network is then pre-trained on single-coupon fatigue data and subsequently adapted to the system level via Low-Rank Adaptation (LoRA), enabling parameter-efficient knowledge transfer under scarce system-level training data. System-level RUL (SRUL) prediction is evaluated under two failure criteria: first-coupon failure, representing a conservative operational limit, and last-coupon failure corresponding to total system collapse. Uncertainty quantification is performed through a dual decomposition strategy, separating epistemic and aleatoric uncertainty. The proposed framework is evaluated against two baselines: a Zero-Shot Baseline, which estimates system RUL as the minimum (or maximum) of coupon-level RULs at each time step without any system-level training, and a Standard Transfer Learning model that uses the pretrained LSTM on system-level data without LoRA adaptation. The LoRA-enhanced framework consistently outperforms both baselines, demonstrating that LoRA-based transfer learning from coupon- to system-level prognostics is an effective and data-efficient strategy for structural PHM, offering a principled path toward uncertainty-aware life prediction in interconnected structural systems.

Keywords: Remaining Useful Life, Transfer Learning, Low-Rank Adaptation, Prognostics Health Management, Uncertainty Quantification, Fatigue Degradation, LSTM,

System-Level Prognostics, Health Indicator, Genetic Algorithm

ΠΕΡΙΛΗΨΗ

Από το Δοκίμιο στο Σύστημα: Ένα Πλαίσιο Βαθιάς Μάθησης και Μεταφοράς Γνώσης για την Πρόβλεψη της Εναπομένουσας Ωφέλιμης Ζωής σε Δομές Πολλαπλών Στοιχείων

Βλαχοπούλου Ευαγγελία

Η διαχείριση προγνωστικής υγείας (PHM) μεταλλικών κατασκευών βασίζεται ολοένα και περισσότερο σε μεθόδους μηχανικής μάθησης για την πρόβλεψη της Εναπομένουσας Ωφέλιμης Ζωής (RUL) — ωστόσο η μετάβαση από την πρόβλεψη σε επίπεδο δοκιμίου προς την πρόβλεψη σε επίπεδο συστήματος παραμένει ανοιχτή πρόκληση, ιδιαίτερα υπό περιορισμένη διαθεσιμότητα δεδομένων συστήματος. Οι υπάρχουσες προσεγγίσεις αντιμετωπίζουν συνήθως την υποβάθμιση μεμονωμένων συνιστωσών, αφήνοντας κενό στις μεθόδους που μπορούν να αξιοποιήσουν αποτελεσματικά τη γνώση επιπέδου δοκιμίου για την πρόβλεψη RUL με ποσοτικοποίηση αβεβαιότητας σε διασυνδεδεμένα πολυσυνιστωσιακά συστήματα. Η παρούσα εργασία προτείνει ένα πλαίσιο μεταφοράς γνώσης για την πρόβλεψη RUL σε επίπεδο συστήματος και την ποσοτικοποίηση αβεβαιότητας σε σύστημα τριών αλουμινένιων δοκιμίων, επικυρωμένο σε πειραματικά δεδομένα κόπωσης που περιλαμβάνουν πέντε μεμονωμένα δοκίμια και έξι τριπλέτες. Αρχικά εφαρμόζεται Γενετικός Αλγόριθμος (GA) για την κατασκευή δείκτη υγείας από μετρήσεις αισθητήρων παραμόρφωσης, παρέχοντας μια μονότονη και πληροφοριακή αναπαράσταση εισόδου για το προγνωστικό μοντέλο. Στη συνέχεια, ένα δίκτυο μνήμης μακράς βραχυπρόθεσμης κατάστασης (LSTM) εκπαιδεύεται εκ των προτέρων σε δεδομένα κόπωσης μεμονωμένων δοκιμίων και στη συνέχεια προσαρμόζεται στο επίπεδο συστήματος μέσω Προσαρμογής Χαμηλής Τάξης (LoRA), επιτρέποντας αποδοτική μεταφορά γνώσης υπό σπάνια δεδομένα συστήματος. Η πρόβλεψη SRUL αξιολογείται υπό δύο κριτήρια αστοχίας: αστοχία πρώτου δοκιμίου, που αντιπροσωπεύει ένα συντηρητικό λειτουργικό όριο, και αστοχία τελευταίου δοκιμίου που αντιστοιχεί στην ολική κατάρρευση του συστήματος. Η ποσοτικοποίηση αβεβαιότητας πραγματοποιείται μέσω στρατηγικής διπλής αποσύνθεσης, διαχωρίζοντας επιστημική και αλεατορική αβεβαιότητα. Το προτεινόμενο πλαίσιο αξιολογείται έναντι δύο μοντέλων αναφοράς: ενός μοντέλου Μηδενικής Εκπαίδευσης, το οποίο εκτιμά το RUL συστήματος ως το ελάχιστο ή μέγιστο

των RUL των δοκιμίων χωρίς καμία εκπαίδευση σε επίπεδο συστήματος, και ενός τυπικού μοντέλου Μεταφοράς Γνώσης χωρίς προσαρμογή LoRA. Το πλαίσιο με LoRA υπερτερεί σταθερά έναντι και των δύο μοντέλων αναφοράς, αποδεικνύοντας ότι η μεταφορά γνώσης βασισμένη σε LoRA από το επίπεδο δοκιμίου στο επίπεδο συστήματος αποτελεί αποτελεσματική και αποδοτική ως προς τα δεδομένα στρατηγική για την PHM κατασκευών.

Λέξεις Κλειδιά: Εναπομένουσα Ωφέλιμη Ζωή, Μεταφορά Γνώσης, Προσαρμογή Χαμηλής Τάξης, Διαχείριση Προγνωστικής Υγείας, Ποσοτικοποίηση Αβεβαιότητας, Υποβάθμιση Κόπωσης, LSTM, Προγνωστική Επιπέδου Συστήματος, Δείκτης Υγείας, Γενετικός Αλγόριθμος

List of Tables

6.1	Optimized hyperparameter configuration for the coupon-level LSTM. . .	42
6.2	Architectural configurations for the three frameworks.	43
6.3	Optimized hyperparameter configuration for the proposed LoRA framework (MAX criterion, five training histories).	44
6.4	Predictive performance of the Zero-Shot Baseline under the Min criterion.	45
6.5	Comparative predictive performance under the Min criterion with a single system-level training history.	46
6.6	Comparative predictive performance under the Min criterion with five system-level training histories.	48
6.7	Predictive performance of the Zero-Shot Baseline under the Max criterion.	49
6.8	Comparative predictive performance under the Max criterion with a single system-level training history.	50
6.9	Comparative predictive performance under the Max criterion with five system-level training histories.	52
6.10	Ablation study results under the Min criterion, comparing the proposed LoRA framework against unconstrained full fine-tuning using a single system-level training history.	56
6.11	Ablation study results under the Max criterion, comparing the proposed LoRA framework against unconstrained full fine-tuning using a single system-level training history.	56

List of Figures

2.1	Structure of the LSTM cell.	7
2.2	Sigmoid activation function.	8
2.3	Hyperbolic tangent activation function.	8
2.4	A typical LSTM architecture for RUL estimation.	10
2.5	Bias-Variance Trade-off.	12
2.6	Training Loss and Validation Loss versus Model Complexity.	13
2.7	Holdout Data Split.	13
2.8	Structure of Leave-One-Out Cross-Validation (LOOCV).	14
2.9	Structure of Nested Leave-One-Out Cross-Validation.	14
2.10	Conceptual overview of transfer learning.	15
2.11	Transfer learning via frozen and trainable layers.	16
3.1	Overview of the proposed five-stage framework for system-level RUL prediction.	23
3.2	Architecture of the LoRA-based system-level adaptation framework.	26
4.1	Architectural pipeline of the Zero-Shot Baseline framework.	31
4.2	Architectural pipeline of the Standard Transfer Learning Baseline, showing the frozen component branches and the trainable MLP aggregator.	33
5.1	Overview of the applied loading conditions and experimental profile of the fatigue machine.	37
5.2	Location of the extracted strain data of the specimens.	38
5.3	Extracted strain data from DIC measurement, specimen 5.	38
5.4	Coupon/Component	39
5.5	Triplet/System	39
6.1	Coupon-level raw strain data and optimised Health Indicator signals derived via Genetic Algorithm.	40
6.2	RUL predictions with uncertainty quantification for the five coupons.	42
6.3	RUL predictions of the Zero-Shot Baseline under the Min criterion.	45

6.4	RUL predictions under the Min criterion with a single system-level training history: Comparison between the Standard Transfer Learning Baseline (red) and the LoRA framework (blue).	46
6.5	RUL predictions under the Min criterion with 5 system-level training histories: Comparison between the Standard Transfer Learning Baseline (red) and the LoRA framework (blue).	47
6.6	RUL predictions of the Zero-Shot Baseline under the Max criterion. . . .	49
6.7	RUL predictions under the Maximum criterion with a single system-level training history: Comparison between the Standard Transfer Learning Baseline (red) and the LoRA framework (blue).	50
6.8	RUL predictions under the Maximum criterion with 5 system-level training histories: Comparison between the Standard Transfer Learning Baseline (red) and the LoRA framework (blue).	51
6.9	Impact of the number of system-level training histories on MAE (left) and CRPS (right).	54
6.10	Impact of the number of system-level training histories on MAE (left) and CRPS (right).	55

List of Abbreviations

CI	Confidence Interval
CRPS	Continuous Ranked Probability Score
DIC	Digital Image Correlation
GA	Genetic Algorithm
HI	Health Indicator
HSMM	Hidden Semi-Markov Model
HMM	Hidden Markov Model
ICML	International Conference on Machine Learning
ICLR	International Conference on Learning Representations
LLM	Large Language Model
LoRA	Low-Rank Adaptation
LOOCV	Leave-One-Out Cross-Validation
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MC	Monte Carlo
MLP	Multi-Layer Perceptron
NLL	Negative Log-Likelihood
NLP	Natural Language Processing
PHM	Prognostics and Health Management
PF	Particle Filter
PNN	Probabilistic Neural Network
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SLP	System-Level Prognostics
SRUL	System Remaining Useful Life
TL	Transfer Learning
TTT	Test-Time Training
UQ	Uncertainty Quantification

Contents

ABSTRACT	iv
ΠΕΡΙΛΗΨΗ	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CONTENTS	xi
1 Introduction	1
2 Theoretical Background	5
2.1 Long Short-Term Memory (LSTM) Networks	5
2.1.1 From RNNs to LSTMs: Motivation	5
2.1.2 Model Evaluation: Bias-Variance Trade-off and Data Splitting . .	12
2.2 Transfer Learning	15
2.2.1 Motivation and Core Concept	15
2.2.2 Feature Extraction via Frozen and Trainable Layers	16
2.2.3 Catastrophic Forgetting and the Limitations of Full Fine-Tuning .	16
2.2.4 Transfer Learning in Prognostics and Health Management	17
2.3 Health Indicators and Genetic Algorithms	17
2.3.1 Health Indicators in Structural Prognostics	17
2.3.2 Genetic Algorithms for Health Indicator Optimisation	19
2.4 Uncertainty Quantification	19
2.4.1 Sources of Predictive Uncertainty	19
2.4.2 Decomposition and Estimation	20
2.4.3 Practical Interpretation in Structural PHM	21
3 Methodology	22
3.1 Health Indicator Construction via Genetic Algorithm	23
3.2 LSTM Pre-training at Component Level	24

3.3	LoRA-Based System-Level Adaptation	24
3.3.1	Mathematical Formulation of LoRA	25
3.3.2	System-Level Architecture and Fine-tuning	26
3.4	Uncertainty Quantification	27
3.4.1	Aleatoric Uncertainty via Probabilistic Neural Network	27
3.4.2	Epistemic Uncertainty via Monte Carlo Dropout	27
3.4.3	Total Predictive Uncertainty and Confidence Intervals	28
3.5	System-Level RUL Aggregation	28
4	Baseline Models and Metrics	30
4.1	Zero-Shot Baseline	30
4.2	Standard Transfer Learning Baseline	31
4.3	Full Fine-Tuning Baseline	33
4.4	Evaluation Metrics	33
5	Case Study	36
5.1	Experimental Campaign	36
5.2	Dataset Description	37
5.2.1	Coupon-Level Data	38
5.2.2	System-Level Data	38
6	Results and Discussion	40
6.1	Coupon-Level RUL Prediction	40
6.2	System Level RUL Prediction Performance	43
6.2.1	Min Criterion	44
6.2.2	Max Criterion	48
6.3	Effect of the Number of System-Level Training Histories	53
6.3.1	Min Criterion	53
6.3.2	Max Criterion	54
6.3.3	Impact of the Low-Rank Constraint: LoRA vs. Full Fine-Tuning	55
7	Limitations and Future Work	57
8	Conclusions	59

ACKNOWLEDGEMENTS

Upon the completion of this diploma thesis, I would like to express my sincere gratitude to all those who supported and guided me throughout this work.

First and foremost, I wish to thank my supervising Professor, Theodoros Loutas of the University of Patras, for entrusting me with this research topic and for his continuous support and encouragement throughout the course of this work. His guidance and scientific insight were invaluable in shaping both the direction and the quality of this thesis.

I would also like to extend my sincere thanks to PhD candidate Giannis Stamatelatos, whose day-to-day guidance, technical expertise, and genuine willingness to help made a profound difference at every stage of this work. His patience, availability, and enthusiasm for the subject were a constant source of motivation and inspiration.

I am deeply grateful to Dr. Nick Eleftheroglou and Dr. Dimitrios Zarouchas of Delft University of Technology (TU Delft) for providing the in-house experimental dataset that forms the foundation of this study, and for their invaluable scientific guidance and constructive feedback throughout the research process. I would also like to thank them for the warm welcome they extended to me during my time at TU Delft, where part of this thesis was carried out. From the very first day, they made me feel like a true member of their research group, creating an environment of openness and collaboration that made the experience both professionally enriching and personally memorable.

Beyond the individual contributions, I feel truly fortunate to have had the opportunity to collaborate with such a dedicated and knowledgeable research team. This experience not only allowed me to explore a topic of significant importance in the field of structural health monitoring and engineering prognostics, but also opened new pathways of knowledge that I had not previously encountered. The collaborative spirit and shared passion for research that characterised this group were, in themselves, among the most valuable aspects of this academic journey.

I would also like to extend my sincere thanks to my fellow student Staikos Konstantinos, with whom I shared the same experimental dataset while pursuing different methodological approaches. Our discussions and ongoing exchange of ideas on the subject contributed greatly to a deeper and more complete understanding of the topic for both of us, and I am grateful for that collaborative spirit.

Finally, I would like to express my deepest gratitude to my family and friends for their unconditional support, patience, and encouragement throughout my years of study. Their belief in me was a continuous source of strength, and this work would not have been

possible without them.

Chapter 1

Introduction

PHM has addressed this necessity primarily through the estimation of RUL, treating it as a probabilistic variable to account for the inherent uncertainties in material degradation [1, 2]. While significant progress has been made in quantifying RUL for isolated components, such as bearings, brakes, batteries, and composite components, extending these predictions to the system level remains a critical but underexplored challenge, as highlighted by recent reviews of system-level prognostics (SLP) [3, 4, 5]. In practice, most industrial systems consist of multiple interlinked components, and maintenance decisions are made at the system level, not at the level of individual elements [3]. A prediction of coupon-level RUL, however accurate, does not directly translate into actionable guidance unless it is integrated into a coherent system-level estimate.

Prognostic methodologies are generally classified into model-based, data-driven, or hybrid approaches. Model-based methods leverage physical laws to describe degradation, such as the use of Paris' Law for crack growth estimation in open-hole metallic structures [6]. While these offer high interpretability and require minimal failure data, they depend on strong domain knowledge and the availability of an accurate physical law. In complex, interconnected aerospace systems, however, the exact underlying physics governing coupled degradation and multi-axial stress states are often not fully known or are too complex to model analytically. Hybrid approaches attempt to combine the strengths of both worlds, yet they often inherit the complexities of physical modeling. Consequently, in complex structural applications, data-driven methods, which handle non-linear degradation patterns without explicit physical equations, are often the only viable option. However, their performance is strictly tied to data availability, making the scarcity of representative training data a critical bottleneck for their application.

In complex aerospace structures, degradation is seldom an isolated phenomenon. Instead, it is governed by the coupled behaviour and load redistribution between interconnected structural elements [7]. SLP aims to capture these interdependencies to provide a holistic estimate of the SRUL [4]. However, a fundamental barrier to SLP is the extreme scarcity of system-level failure data. Due to the prohibitive cost and time required for

full-scale fatigue testing [8], structural certification programs allocate only a handful of full-scale tests across the entire qualification process, standard data-driven models struggle to generalise at the structural level. Prior work has demonstrated the feasibility of upscaling prognostic methodologies from simpler structural elements to more complex assemblies [9], yet such approaches still rely on data-rich lower-level observations and do not directly address the case of severely scarce system-level failure data. This creates a “scale gap” between the high-volume data available from coupon-level testing and the sparse data context of integrated structural assemblies.

Existing literature in SLP often circumvents this challenge by treating systems as collections of independent components or by assuming that the dependency relationships between subsystems are known a priori and fixed [3, 4]. While some advanced methods have been proposed to model inter-component interactions, including GCN approaches [10], physics-based or model-driven ensemble strategies [9], and the graph neural network framework of Stamatelatos et al. [11] for spatially distributed strain sensor assemblies, they typically remain data-intensive and computationally demanding, limiting their practical application in scenarios where system-level observations are nearly non-existent. There is, therefore, a pressing need for a framework that can bridge this gap by transferring knowledge from the data-rich coupon domain to the data-scarce system domain.

Transfer learning (TL) has been increasingly adopted in PHM to overcome data scarcity by leveraging knowledge acquired in a source domain to improve performance in a related target domain [12, 13]. Applied to RUL prediction, TL-based LSTM frameworks have demonstrated the ability to adapt degradation patterns across different operating conditions and component types [12]. However, existing approaches typically rely on full fine-tuning of pre-trained networks requires sufficient target-domain data to avoid overfitting. LoRA [14] addresses this limitation by freezing the pre-trained model weights and injecting trainable low-rank decomposition matrices, drastically reducing the number of parameters that need to be updated during adaptation. Originally developed for large language models (LLMs), LoRA has since been applied to time series foundation models [15], demonstrating that parameter-efficient fine-tuning can be extended beyond Natural Language Processing (NLP). Motivated by this limitation, this paper proposes a novel transfer learning framework for system-level RUL prediction using LoRA in interconnected aluminum structures. The core novelty lies in the application of LoRA to a pre-trained LSTM network. By pre-training the model on individual coupon degradation histories and subsequently adapting it to a three-coupon “triplet” system using parameter-efficient fine-tuning, we enable accurate SRUL estimation even under severe system-level data scarcity. This approach allows the model to learn complex structural interdependencies without requiring the extensive system-level datasets typically demanded by deep learning architectures.

To ensure the reliability of the proposed framework, the system-level predictions are integrated into an uncertainty-aware pipeline. An optimised Health Indicator (HI) is first constructed from raw strain measurements using a Genetic Algorithm (GA) [16], targeting desirable prognostic properties, monotonicity and prognosability, following established practice in structural PHM [17, 18]. Following the established formulation of Kendall and Gal [19], predictive uncertainty is decomposed into epistemic and aleatoric components, with epistemic uncertainty captured via MC Dropout [20] and aleatoric uncertainty modelled through a Gaussian likelihood head trained with the negative log-likelihood loss [21]. While this dual decomposition is well established in deep learning, its systematic application and reporting in structural PHM remains uncommon: prognostic studies frequently report only total predictive variance [19] or quantify a single uncertainty source, obscuring whether residual spread reflects irreducible fatigue scatter or model ignorance under data scarcity. In the present framework, the two components are tracked separately and propagated consistently through both the coupon-level pre-training and the LoRA-based system-level adaptation, enabling a transparent assessment of how parameter-efficient fine-tuning influences each source of uncertainty as system-level data is incrementally introduced. This disentanglement provides actionable information for maintenance decision-making: epistemic uncertainty signals that additional data collection would reduce model uncertainty, while aleatoric uncertainty reflects the irreducible variability of the fatigue process itself [20, 19, 2].

The methodology is validated using a unique in-house experimental dataset of aluminum coupons and triplets, evaluating system failure under both conservative (first-coupon failure) and non-conservative (last-coupon failure) criteria.

The main contributions of this work are summarised as follows:

- A parameter-efficient transfer learning framework (LoRA-LSTM) for coupon-to-system RUL adaptation, specifically designed for structural fatigue applications with limited system-level data.
- Demonstration of structural knowledge transfer, showing how degradation patterns learned at the coupon scale can be leveraged to capture the coupled degradation behaviour of multi-component assemblies.
- An uncertainty-aware prognostic pipeline integrating GA-optimised feature extraction and dual uncertainty quantification to support robust operational decision-making at the system level.
- A comparative empirical analysis under two system failure criteria first-to-fail and last-to-fail providing complementary perspectives on system-level life prediction for structurally coupled assemblies.

The remainder of this paper is organised as follows. Chapter 2 provides the theoretical background. Chapter 3 details the methodology, including the LoRA-based adaptation and the uncertainty propagation strategy. Chapter 4 outlines the baseline models used for comparison. Chapter 5 describes the experimental campaign and data collection. Chapter 6 presents and discusses the results. Chapter 7 outlines the limitations and future work, and Chapter 8 presents the conclusions.

Chapter 2

Theoretical Background

This chapter provides the theoretical foundations underlying the proposed framework. The concepts and methods reviewed here were selected because they collectively motivate and inform the design choices made in the methodology presented in Chapter 3. Specifically, Section 2.1 introduces LSTM networks and the principles of model evaluation that govern the coupon-level training strategy. Section 2.2 reviews transfer learning and its limitations, motivating the adoption of parameter-efficient fine-tuning. Section 2.3 covers Health Indicators and Genetic Algorithms, establishing the degradation descriptor used as input throughout the framework. Finally, Section 2.4 introduces the sources of predictive uncertainty and their formal decomposition. All content in this chapter is presented at a general level, independent of any specific implementation; the application of these concepts to the present dataset and experimental setup is detailed in Chapter 3.

2.1 Long Short-Term Memory (LSTM) Networks

2.1.1 From RNNs to LSTMs: Motivation

This section introduces the concept and motivation behind the use of LSTM networks for modeling temporal data, followed by an overview of their internal architecture and main computational principles.

When dealing with time-dependent data, it is crucial to employ models capable of learning how current observations relate to past information. Recurrent Neural Networks (RNNs) represent a class of neural networks specifically designed for processing temporal or sequential data, as they possess internal memory that allows them to retain information from previous time steps and model temporal dependencies. However, RNNs suffer from a limitation called the vanishing/exploding gradient problem, which hinders their ability to learn long-term dependencies, leading to loss of relevant information as the sequence length increases.

To overcome this issue, the LSTM architecture was introduced, offering a significant

improvement over standard RNNs by enabling the learning of long-term dependencies without suffering from gradient instability [22].

LSTM Cell Structure

The internal structure of the LSTM model defines how information is stored, updated, and propagated through time. Its operation relies on a combination of memory units and control gates, as described below. The core idea of LSTM lies in the introduction of a memory cell, which stores information over extended periods, and three gates that regulate the flow of information within the network:

- The forget gate, which determines which parts of the previous information should be discarded,
- The input gate, which decides what new information should be added to the memory cell, and
- The output gate, which controls how much of the stored information is used to produce the current output.

These gates are governed by sigmoid and hyperbolic tangent activation functions, which allow the network to smoothly regulate the amount of information that is remembered or forgotten at each time step. The mathematical representation of the LSTM mechanism is given by the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.3)$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \quad (2.4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.5)$$

$$h_t = o_t \otimes \tanh(C_t) \quad (2.6)$$

where x_t represents the input vector at time step t , h_t is the hidden state, and C_t is the internal cell state. $\tilde{C}_t$ is the candidate memory cell within the LSTM architecture, containing the new information extracted from the current input and previous hidden

state, which is then selectively integrated, through the input and forget gates, into the final memory state C_t that preserves the network's long-term knowledge. The matrices W and U correspond to trainable weight parameters, while b denotes the bias terms. The symbol $\otimes$ denotes element-wise (Hadamard) multiplication, meaning that the corresponding elements of two vectors are multiplied together [18].

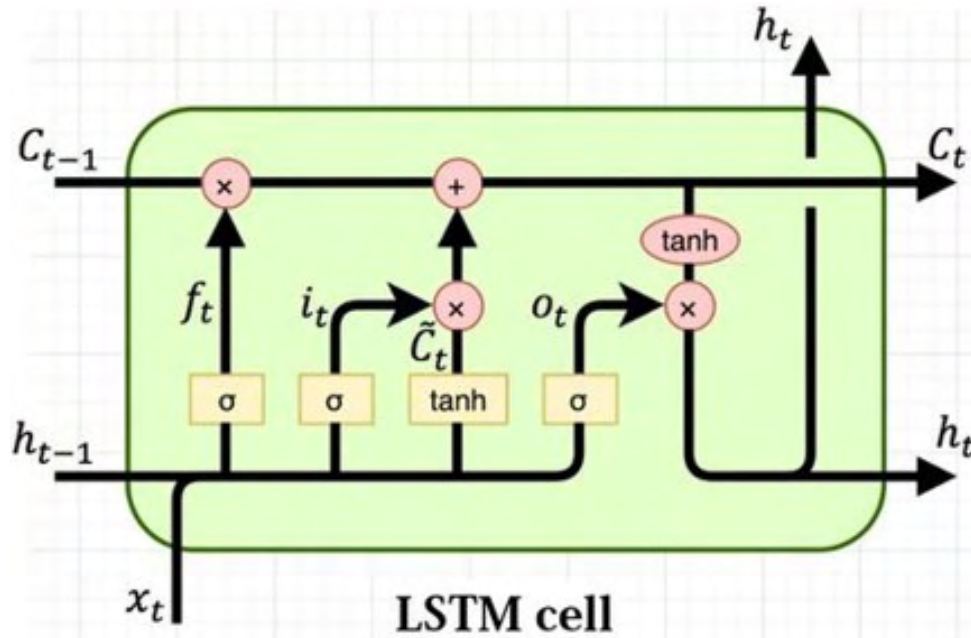


Figure 2.1: Structure of the LSTM cell.

In LSTM networks, two activation functions are used: the sigmoid and the hyperbolic tangent (tanh). The sigmoid activation function is applied to the input, forget, and output gates to regulate information flow by producing values within $[0, 1]$, effectively controlling how much of the current input and previous memory contributes to the new state. Mathematically, the sigmoid function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.7)$$

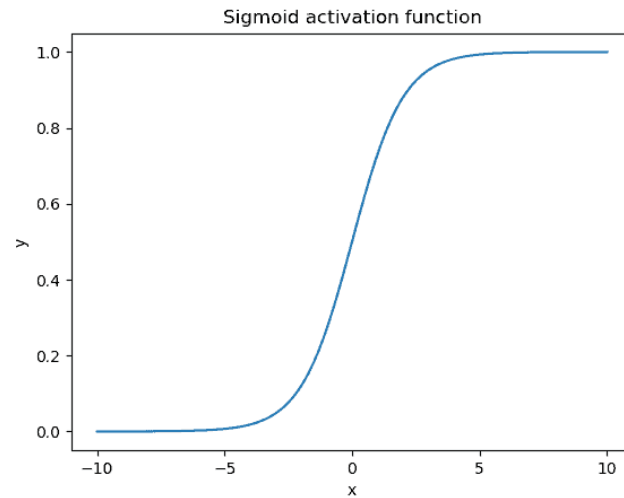


Figure 2.2: Sigmoid activation function.

The tanh activation function is utilized for computing the candidate cell state and for shaping the final hidden state output, enabling representation of both positive and negative values in the range $[-1, 1]$. Mathematically, the tanh function is defined as:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.8)$$

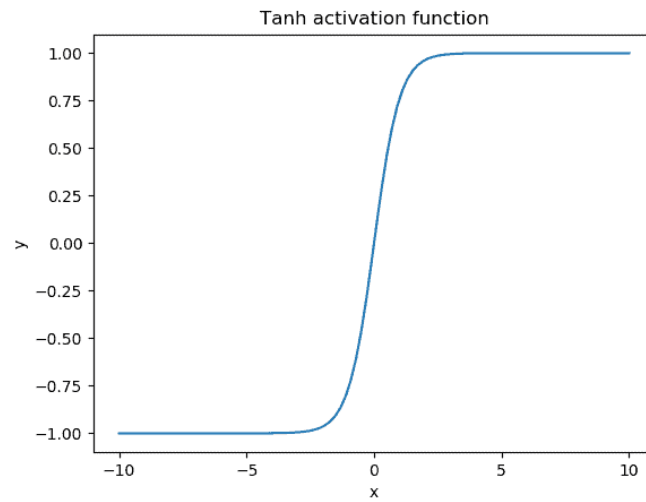


Figure 2.3: Hyperbolic tangent activation function.

Advanced LSTM Configurations

The standard LSTM configuration, often referred to as the Vanilla LSTM, serves as the foundation for more advanced architectures. In practice, stacking multiple LSTM layers can lead to improved learning of higher-order temporal representations, forming what is known as a Deep LSTM (DLSTM) network.

The Vanilla LSTM architecture typically consists of an input layer, one or more LSTM layers, and an output layer. Each LSTM cell maintains information across time, enabling the model to effectively capture complex temporal patterns. This makes LSTMs particularly suitable for RUL estimation, where it is essential to model the temporal degradation behavior of materials or components.

In more advanced designs, multiple LSTM layers can be stacked to form a DLSTM architecture. In such networks, the hidden state output of each LSTM layer serves as the input to the next layer. This hierarchical structure allows the model to extract temporal features at different levels of abstraction, since the lower layers capture short-term signal variations, while higher layers learn more abstract and long-term dependencies. As a result, DLSTM networks can represent complex nonlinear relationships within the data more effectively than shallow architectures.

Moreover, DLSTMs tend to exhibit improved generalization performance, especially when trained on large datasets with complex temporal behavior. However, their increased depth also introduces challenges, such as longer training times and the need for proper regularization to prevent overfitting. To address these issues, techniques such as dropout are commonly applied during training, randomly deactivating a subset of neurons to improve robustness. In addition, the inclusion of a masking layer enables the model to process sequences of varying lengths, ensuring flexibility when dealing with temporal data.

Due to their robustness, flexibility, and high predictive accuracy, LSTM-based and particularly DLSTM architectures have become fundamental tools in time-series forecasting and PHM. They have demonstrated strong performance in identifying hidden degradation patterns, predicting system evolution, and estimating the RUL of engineering components.

A representative example of a typical LSTM architecture is illustrated in Figure 2.4, showing a many-to-one configuration consisting of an input layer, one or more LSTM layers, a fully connected hidden layer, and a linear output layer.

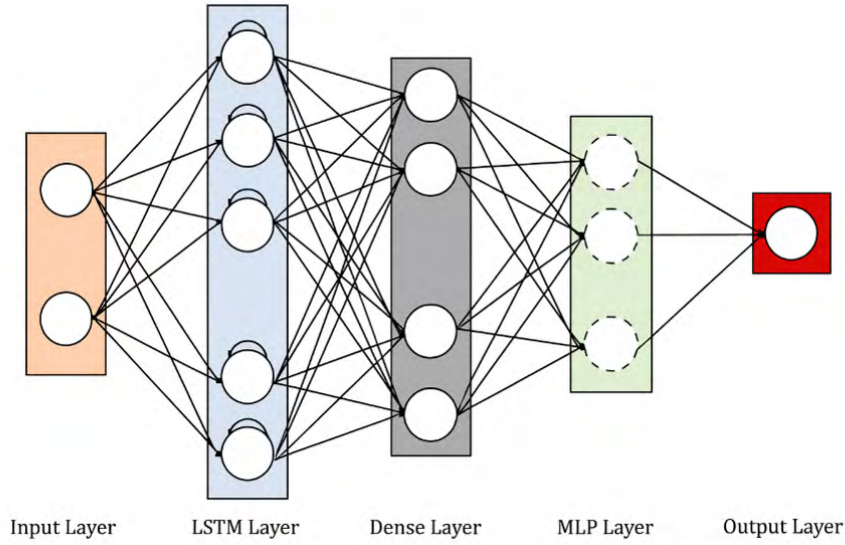


Figure 2.4: A typical LSTM architecture for RUL estimation.

Training: Gradient Descent and Backpropagation Through Time

LSTM networks are trained using Gradient Descent (GD) optimization, which iteratively adjusts the model parameters to minimize the prediction error. Given that LSTMs contain recurrent connections, gradients are computed through Backpropagation Through Time (BPTT), a process that unfolds the temporal layers of the network to capture long-range dependencies in sequential data. To quantify the prediction error, the Mean Squared Error (MSE) loss function is commonly employed:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.9)$$

where $\hat{y}_i$ denotes the predicted value at time step i , y_i is the corresponding ground-truth value, and N is the number of training samples. Common choices for regression tasks include the MSE, the Mean Absolute Error (MAE), and the Huber loss, while probabilistic formulations may employ the Negative Log-Likelihood (NLL) [23]. During training, GD updates all trainable parameters θ of the network (input weights, recurrent weights, and bias terms) according to:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} L(\theta_t) \quad (2.10)$$

where η is the learning rate controlling the step size of parameter updates. The gradients $\nabla_{\theta} L$ are propagated through the LSTM gates using the chain rule, allowing the network to learn jointly from past and present information.

Hyperparameter Tuning Strategies

Beyond the physical architecture of an LSTM network, a second group of hyperparameters governs how the network updates its internal weights and processes input sequences during training. The most consequential of these are the learning rate, the batch size, and the input window size.

Learning Rate. The learning rate controls the step size taken during each iteration of gradient descent. A value that is too high causes the optimisation to overshoot the minimum or diverge entirely, while a value that is too low results in slow convergence and increases the risk of becoming trapped in suboptimal local minima [23].

Batch Size. The batch size defines the number of training samples processed before each weight update. Smaller batches introduce stochastic noise into the gradient estimate, which can act as an implicit regulariser and help the network escape local minima, at the cost of more update iterations. Larger batches provide more accurate gradient estimates and faster wall-clock training, but may reduce generalisation performance on unseen data.

Window Size. In sequential time-series modelling, the window size determines how many past observations are provided to the network at each time step. For structural health monitoring applications, selecting an appropriate window is critical: a window that is too short fails to capture the long-term history of damage accumulation, while an excessively long window introduces unnecessary computational complexity and may dilute the most informative recent observations.

Because the interaction between hyperparameters and model performance is highly non-linear, manual tuning is generally impractical. Contemporary research therefore relies on automated optimisation strategies [24]:

- **Grid Search** exhaustively evaluates every combination across a predefined discrete grid. While guaranteed to find the best combination within the specified grid, it scales poorly with the number of hyperparameters and becomes prohibitively expensive in high-dimensional search spaces.
- **Random Search** samples combinations randomly from defined statistical distributions. By exploring a more diverse region of the parameter space per unit of compute, it typically locates competitive configurations in a fraction of the time required by grid search.
- **Bayesian Optimisation** constructs a probabilistic surrogate model of the objective function and uses it to select the most promising configurations to evaluate next, balancing exploration of unvisited regions with exploitation of regions already known to perform well. This approach significantly reduces the number of training runs required and is the method of choice in contemporary PHM literature [12].

2.1.2 Model Evaluation: Bias-Variance Trade-off and Data Splitting

Bias-Variance Trade-off

A fundamental consideration in supervised learning is the decomposition of prediction error into two competing sources: bias and variance [25, 26]. Bias refers to the error introduced by approximating a complex relationship with an insufficiently flexible model, leading to underfitting. Variance refers to the model's sensitivity to fluctuations in the training data, leading to overfitting. As model complexity increases, bias decreases while variance grows, producing a U-shaped total error curve with an optimum at intermediate complexity, as illustrated in Figure 2.5.

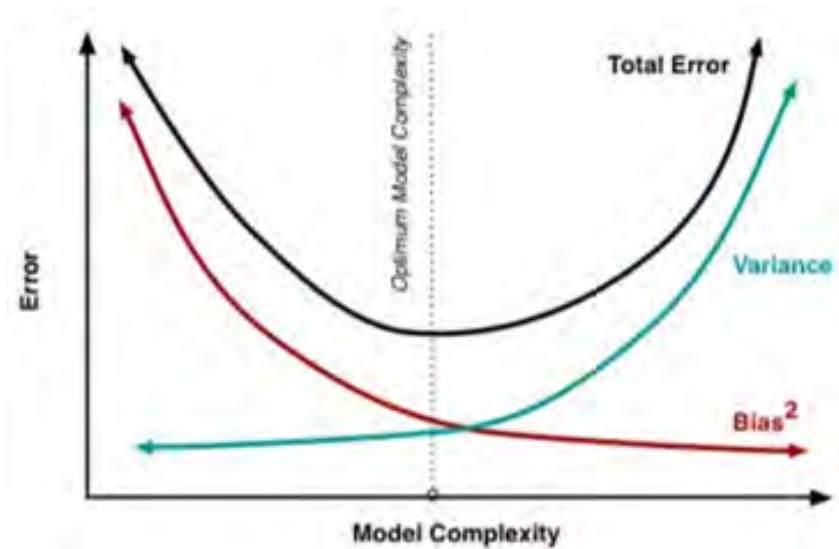


Figure 2.5: Bias-Variance Trade-off.

In the context of LSTM networks, this trade-off is governed by architectural choices such as the number of units and layers, the dropout rate, and the training duration, as depicted in Figure 2.6. All of these are treated as hyperparameters and optimised via Bayesian optimisation in the present framework (Section 3.2).

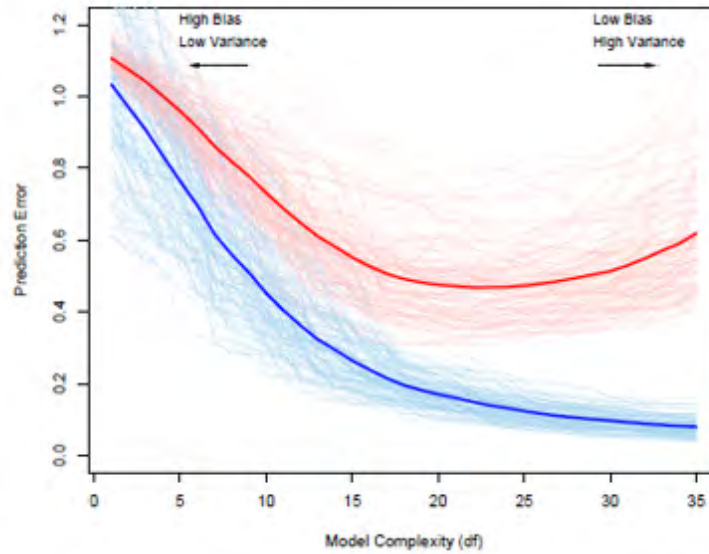


Figure 2.6: Training Loss and Validation Loss versus Model Complexity.

Data Splitting Strategies

The choice of data splitting strategy directly affects the reliability of the performance estimate, particularly under limited data availability. Three approaches of increasing sophistication are considered here:

1. Holdout Method. The simplest approach partitions the dataset into fixed training, validation, and test subsets, as illustrated in Figure 2.7. While computationally efficient, this method produces estimates that are highly sensitive to the particular split chosen and is therefore unsuitable when the dataset is small.

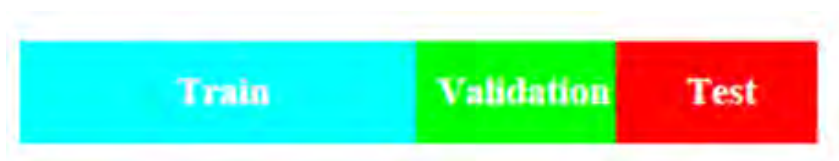


Figure 2.7: Holdout Data Split.

2. Leave-One-Out Cross-Validation (LOOCV). LOOCV is a special case of K -fold cross-validation in which $K = N$, where N is the total number of samples. In each iteration, a single sample is held out as the test set while the model is trained on the remaining $N - 1$ samples. This procedure is repeated N times so that each sample serves as the test set exactly once, as depicted in Figure 2.8. By maximizing the amount of training data available at each fold, LOOCV produces nearly unbiased estimates of generalization performance, making it particularly well-suited for small datasets. The final performance is reported as the mean μ and standard deviation σ of the error metrics across folds:

$$\text{MAE}_{\text{final}} = \mu \pm \sigma \quad (2.11)$$

$$\text{RMSE}_{\text{final}} = \mu \pm \sigma \quad (2.12)$$



Figure 2.8: Structure of Leave-One-Out Cross-Validation (LOOCV).

3. Nested LOOCV. When both model selection and performance evaluation must be performed on a limited dataset, a nested cross-validation scheme is required to avoid optimistic bias. In Nested LOOCV, an outer LOOCV loop is responsible for performance estimation, while an inner LOOCV loop handles hyperparameter tuning or model selection independently for each outer fold. In each outer iteration, one sample is held out as the test set; within the remaining samples, an inner LOOCV is performed to select the optimal hyperparameters, and the model retrained on all inner samples is then evaluated on the outer test sample. This two-loop structure ensures a strict separation between the model selection and evaluation phases, preventing information leakage and yielding an unbiased estimate of generalization performance, as illustrated in Figure 2.9.

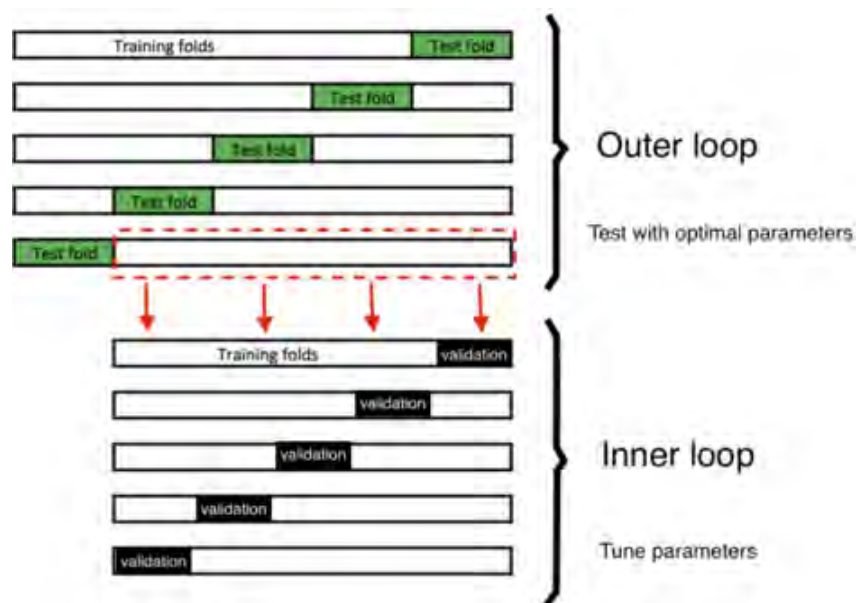


Figure 2.9: Structure of Nested Leave-One-Out Cross-Validation.

LOOCV is preferred over the holdout method and Nested LOOCV because it maximises the use of available training data at each fold while providing a statistically robust and unbiased estimate of generalisation performance across unseen specimens. This is a critical requirement given the scarcity of coupon-level failure data in structural PHM applications.

2.2 Transfer Learning

2.2.1 Motivation and Core Concept

Transfer learning (TL) is a machine learning paradigm in which knowledge acquired while solving one problem is leveraged to improve performance on a related but distinct problem. Rather than training a model from scratch on a target task, TL initialises the learning process from a pre-trained model that has already captured relevant patterns from a source domain, as illustrated in Figure 2.10. This is particularly valuable in engineering applications where labelled failure data are scarce, expensive to obtain, or require destructive testing to collect.

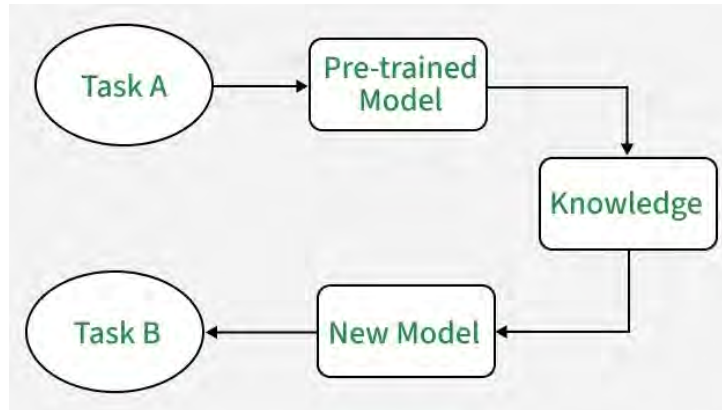


Figure 2.10: Conceptual overview of transfer learning.

Formally, let $\mathcal{D}_S = \{\mathcal{X}_S, P(X_S)\}$ denote the source domain with feature space $\mathcal{X}_S$ and marginal distribution $P(X_S)$, and let $\mathcal{T}_S$ denote the corresponding source task. Similarly, let $\mathcal{D}_T$ and $\mathcal{T}_T$ denote the target domain and task respectively. Transfer learning aims to improve the learning of a target predictive function $f_T(\cdot)$ using knowledge from $\mathcal{D}_S$ and $\mathcal{T}_S$, where in general $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

In the context of the present work, the source domain consists of coupon-level fatigue degradation sequences, while the target domain corresponds to system-level (triplet) degradation under mechanical coupling. Although both domains share the same physical material and loading mechanism, the target domain introduces structural interdependencies and load redistribution effects that are absent at the coupon level, making it a related but non-identical learning problem.

2.2.2 Feature Extraction via Frozen and Trainable Layers

A common transfer learning strategy is feature extraction, in which the weights of a pre-trained model are frozen and only a small number of task-specific layers are trained on the target data. As illustrated in Figure 2.11, the frozen layers act as a fixed feature extractor that maps inputs into a learned representation space, while the trainable layers adapt this representation to the target task.

Transfer Learning: Frozen and Trainable Layers

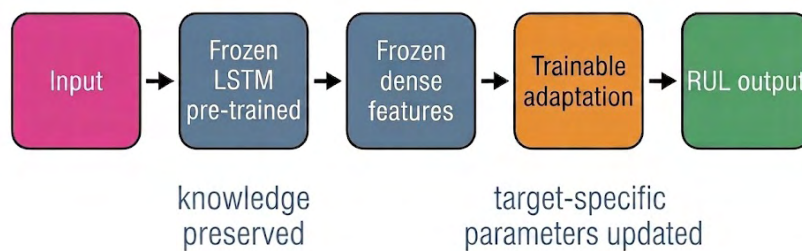


Figure 2.11: Transfer learning via frozen and trainable layers.

This approach is particularly well-suited to scenarios where the target dataset is small, as it prevents overfitting by drastically reducing the number of parameters that need to be estimated from limited data. The decision of which layers to freeze and which to train depends primarily on the similarity between the source and target domains. When the two domains are closely related, freezing the majority of the pre-trained weights and training only the final task-specific layers is typically sufficient. In the present framework, the pre-trained LSTM backbone, trained on coupon-level health indicator sequences, serves as a frozen feature extractor, and only the system-level aggregator is trained on the available triplet data.

2.2.3 Catastrophic Forgetting and the Limitations of Full Fine-Tuning

An alternative to feature extraction is full fine-tuning, in which all model weights are updated during adaptation to the target domain. While this approach offers greater flexibility, it is prone to a phenomenon known as catastrophic forgetting: when a neural network is trained sequentially on two tasks, updating all weights on the new task tends to overwrite the representations learned during pre-training, erasing the knowledge acquired from the source domain [22].

In the context of structural prognostics, catastrophic forgetting is particularly damaging. The coupon-level pre-training encodes the temporal dynamics of fatigue degradation, a physically grounded, slowly evolving process, into the LSTM weights. If these weights are fully updated on only a handful of system-level failure histories, the model risks losing this degradation knowledge and overfitting to the limited target data. The challenge is therefore to adapt the model to the system level while preserving the degradation physics captured during pre-training. This motivates the use of parameter-efficient fine-tuning strategies, such as Low-Rank Adaptation (LoRA), which are discussed in the following section.

2.2.4 Transfer Learning in Prognostics and Health Management

Transfer learning has been increasingly adopted in PHM to address the persistent challenge of data scarcity. In practice, run-to-failure datasets are difficult and costly to obtain, particularly at the system level where full-scale fatigue testing is prohibitively expensive. TL-based approaches mitigate this bottleneck by leveraging degradation knowledge from related components or operating conditions to improve prediction in a data-scarce target setting [12, 13].

Applied to RUL prediction, TL frameworks typically pre-train a model on a data-rich source domain, such as a well-instrumented component type or a specific loading condition and subsequently adapt it to a target domain where failure observations are sparse. In the structural fatigue context, this corresponds to transferring knowledge from the abundant coupon-level testing programme to the scarce system-level triplet dataset, bridging what has been referred to as the scale gap between component and assembly testing.

2.3 Health Indicators and Genetic Algorithms

2.3.1 Health Indicators in Structural Prognostics

A Health Indicator (HI) is a scalar quantity derived from sensor measurements that tracks the evolution of damage in a structural component from its initial undamaged state through to final failure. In the context of data-driven prognostics, the HI serves as the primary input to the predictive model, and its quality directly determines the reliability of the resulting RUL estimates. Raw sensor signals, such as strain measurements from surface-mounted gauges, are typically high-dimensional, noisy, and do not inherently reflect the monotonic accumulation of damage. A well-constructed HI must therefore distil this information into a single, physically meaningful signal that captures the progressive degradation of the monitored component [27].

The construction of a suitable HI from raw sensor data can follow two broad approaches. Physics-based methods derive the HI from theoretical models of the underlying damage mechanism, ensuring interpretability and physical consistency. Data-driven methods, by contrast, learn a mapping from raw features to a scalar HI directly from experimental data, without requiring an explicit physical model. In complex structural applications where the governing damage mechanics are not fully known, data-driven feature construction is often the only viable option [27].

Regardless of the approach adopted, the suitability of a candidate HI for prognostic purposes is evaluated through two quantitative criteria: monotonicity and prognosability.

Monotonicity reflects the physically motivated requirement that damage accumulates irreversibly over the specimen life. A well-constructed HI should increase (or decrease) consistently with time, without reversals unrelated to actual damage progression. For a population of N specimens, the average monotonicity is defined as:

$$Mo = \frac{1}{N} \sum_{j=1}^N M_j \quad (2.13)$$

where M_j is the monotonicity of specimen j , computed from the signs of the first-order differences of the HI sequence:

$$M_j = \left| \frac{n_j^+ - n_j^-}{n_j - 1} \right| \quad (2.14)$$

with n_j^+ and n_j^- denoting the number of positive and negative increments respectively, and n_j the total number of observations.

Prognosability quantifies the consistency of the HI value at end-of-life across the specimen population. A high prognosability indicates that different specimens reach a similar HI level at failure, which is essential for reliable threshold-based life prediction:

$$Pr = \exp \left(- \frac{\text{std}(\text{HI}_{\text{eol}})}{|\overline{\text{HI}_{\text{eol}}} - \overline{\text{HI}_0}|} \right) \quad (2.15)$$

where HI_{eol} and HI_0 denote the HI values at end-of-life and at the initial cycle across all specimens respectively, $\text{std}(\cdot)$ is the sample standard deviation, and the overline denotes the sample mean.

Both criteria are normalised to $[0, 1]$, where unity corresponds to ideal behaviour. They are combined into a scalar fitness function:

$$F(\alpha) = m \cdot \text{Monotonicity} + n \cdot \text{Prognosability} \quad (2.16)$$

where m and n are scalar weights reflecting the relative importance of each criterion.

2.3.2 Genetic Algorithms for Health Indicator Optimisation

The construction of the HI from raw sensor data requires determining the optimal combination of features extracted from the measurements. This is formulated as an optimisation problem: find the feature weights that maximise the fitness function $F(\alpha)$ defined above. Given the non-convex and potentially multimodal nature of this search space, classical gradient-based optimisation is not well-suited. Instead, a Genetic Algorithm (GA) is employed [16].

Genetic Algorithms are a class of evolutionary optimisation methods inspired by the principles of natural selection and biological evolution. The algorithm operates on a population of candidate solutions, referred to as individuals, each encoding a set of parameters to be optimised. Over successive generations, individuals are selected, combined, and modified according to their fitness, driving the population toward progressively better solutions. The main stages of a GA are:

- Initialisation: a population of candidate solutions is generated randomly within the search space.
- Selection: individuals with higher fitness values are preferentially chosen to reproduce, favouring solutions that better satisfy the optimisation objective.
- Crossover: pairs of selected individuals exchange portions of their parameter vectors to produce offspring, combining favourable characteristics from both parents.
- Mutation: small random perturbations are applied to individual parameters, maintaining diversity within the population and preventing premature convergence to local optima.
- Termination: the process repeats until a stopping criterion is met, such as a maximum number of generations or convergence of the fitness value.

2.4 Uncertainty Quantification

2.4.1 Sources of Predictive Uncertainty

In data-driven prognostics, a model's prediction is never perfectly certain. Predictive uncertainty arises from two fundamentally distinct sources, and distinguishing between them is essential for informed maintenance decision-making [19].

Aleatoric uncertainty reflects the irreducible randomness inherent in the data-generating process itself. In the context of structural fatigue, this corresponds to the natural scatter in crack initiation sites, microstructural variability, and load fluctuations across nominally identical specimens. Because it stems from the physical process rather than the model,

aleatoric uncertainty cannot be reduced by collecting additional data. Mathematically, it is modelled as the variance of the residual noise around the predicted mean:

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_a^2) \quad (2.17)$$

where σ_a^2 is the aleatoric variance, treated as a learnable output of the network.

Epistemic uncertainty, by contrast, reflects the model's lack of knowledge due to limited or unrepresentative training data. It is therefore reducible: as more system-level observations become available, the model's parameter distribution narrows and epistemic uncertainty decreases. This distinction is particularly relevant in the present framework, where system-level failure data are severely scarce and epistemic uncertainty is expected to dominate early in the adaptation process.

2.4.2 Decomposition and Estimation

Following the formulation of Kendall and Gal [19], the total predictive variance can be decomposed into its two components:

$$\underbrace{\text{Var}(\hat{y})}_{\text{Total}} = \underbrace{\mathbb{E}[\sigma_a^2]}_{\text{Aleatoric}} + \underbrace{\text{Var}[\mu_\theta]}_{\text{Epistemic}} \quad (2.18)$$

where the expectation and variance are taken over the approximate posterior distribution of the model parameters θ .

Aleatoric uncertainty is captured by augmenting the network with a Gaussian likelihood head: rather than predicting only a point estimate $\hat{y}$, the model outputs both a mean $\mu_\theta(x)$ and a log-variance $\log \sigma_a^2(x)$. The network is then trained by minimising the Negative Log-Likelihood (NLL) loss:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{N} \sum_{i=1}^N \left[\frac{(y_i - \mu_\theta(x_i))^2}{2\sigma_a^2(x_i)} + \frac{1}{2} \log \sigma_a^2(x_i) \right] \quad (2.19)$$

This formulation encourages the model to report higher aleatoric uncertainty in regions where the targets are inherently noisy, without requiring explicit noise labels.

Epistemic uncertainty is estimated via Monte Carlo (MC) Dropout [20]. Dropout layers, retained active at inference time, induce stochasticity in the forward pass, approximating sampling from the posterior distribution over model parameters [20]. Given T stochastic forward passes, the epistemic variance is estimated as:

$$\sigma_e^2 \approx \frac{1}{T} \sum_{t=1}^T \mu_{\theta_t}(x)^2 - \left(\frac{1}{T} \sum_{t=1}^T \mu_{\theta_t}(x) \right)^2 \quad (2.20)$$

The mean aleatoric variance is then recovered by averaging $\sigma_{a,t}^2$ across the T samples,

yielding a consistent decomposition of total predictive uncertainty.

2.4.3 Practical Interpretation in Structural PHM

The dual decomposition provides actionable information beyond a single confidence interval. High epistemic uncertainty signals that model predictions would benefit from additional system-level data, motivating targeted experimental campaigns or sensor augmentation. High aleatoric uncertainty, by contrast, reflects genuine fatigue scatter that no amount of additional data can eliminate, and should instead inform conservative safety margins in maintenance scheduling.

Chapter 3

Methodology

The proposed framework addresses the problem of system RUL prediction under scarce system-level data through a sequential five-stage pipeline, illustrated in Figure 3.1. In the first stage, a GA is employed to construct a HI from raw sensor measurements for each component, yielding a monotonic and prognostically informative scalar signal that serves as the common input representation across all subsequent stages. In the second stage, a LSTM network is pre-trained on component HI sequences, learning the temporal dynamics of fatigue degradation at the component scale; component-level RUL predictions are produced together with associated confidence intervals through a Negative Log-Likelihood (NLL) training objective. In the third stage, the pre-trained LSTM is adapted to the system level via LoRA: the pre-trained model weights are frozen and trainable low-rank decomposition matrices are injected, enabling parameter-efficient knowledge transfer from the component domain to the three-component triplet system without requiring extensive system-level failure data. System-level HI sequences, constructed from the component-level HIs of the interconnected specimens, are used as input at this stage. In the fourth stage, a dual uncertainty decomposition is applied: MC Dropout quantifies epistemic uncertainty arising from model parameter variability, while a PNN captures aleatoric uncertainty stemming from the inherent stochasticity of the fatigue process; the two components are combined to form the total predictive uncertainty. Finally, in the fifth stage, the system-level RUL is aggregated under two failure criteria, first-component failure (*Min criterion*), corresponding to the minimum component RUL at each time step, and last-component failure (*Max criterion*), corresponding to the maximum, providing complementary conservative and operational perspectives on structural life prediction.

The following subsections detail each stage of the framework.

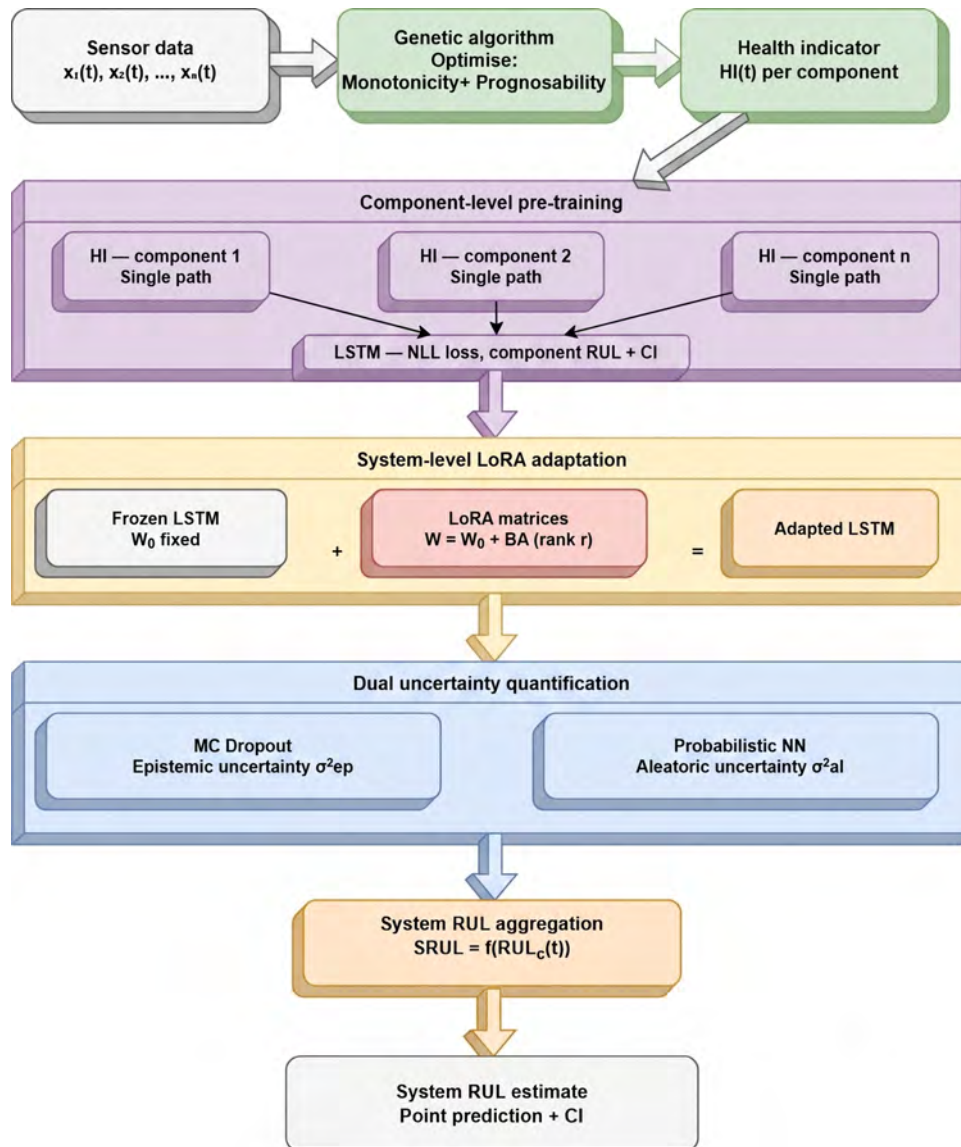


Figure 3.1: Overview of the proposed five-stage framework for system-level RUL prediction.

3.1 Health Indicator Construction via Genetic Algorithm

As discussed in Section 2.3, reliable RUL estimation requires a monotonic and prognostically consistent degradation descriptor rather than raw sensor signals, which are high-dimensional, noisy, and do not inherently reflect damage accumulation [28]. In the present framework, this descriptor is constructed in the form of a HI via a GA.

The candidate HI is expressed as a weighted linear combination of $P = 5$ strain-derived features $\{f_p\}_{p=1}^P$:

$$\text{HI}(\alpha) = \sum_{p=1}^P \alpha_p f_p \quad (3.1)$$

where $\alpha = [\alpha_1, \dots, \alpha_P]^\top \in \mathbb{R}^P$ is the parameter vector to be optimised. In the present work, equal weighting is adopted ($m = n = 1$) in the fitness function (Equation 2.16), as both monotonicity and prognosability are considered equally critical for reliable RUL estimation. The GA searches the real-valued weight space $\mathbb{R}^5$ for the combination α^* that maximises the fitness function $F(\alpha)$, yielding a HI that is both monotonic and prognostically consistent across the coupon population. A detailed description of the feature extraction procedure and the GA configuration is provided in [17, 18].

3.2 LSTM Pre-training at Component Level

Fatigue degradation is an inherently sequential process, where the current damage state depends on the entire loading history. To capture these temporal dependencies, a LSTM architecture is employed [22], as introduced in Section 2.1. LSTMs are specifically designed to mitigate the vanishing gradient problem of standard RNNs through a gating mechanism, making them ideal for modeling long-term fatigue trends.

The network is pre-trained exclusively on single-component HI sequences to encode the underlying degradation dynamics. At each time step t , the model processes a sliding window of L consecutive HI values, $\mathbf{x}_t = [\text{HI}(t - L + 1), \dots, \text{HI}(t)]^\top$. Rather than producing a deterministic estimate, the network operates as a PNN: the final hidden state $\mathbf{h}_t$ is mapped to a predicted mean $\hat{\mu}_t$ and variance $\hat{\sigma}_t^2$, parametrising a Gaussian distribution over the RUL. Training is performed by minimising the NLL loss (Eq. 2.19). The adopted architecture consists of a single LSTM layer followed by a fully connected dense layer, with the exact configuration determined through Bayesian hyperparameter optimisation.

The resulting weights serve as the foundation for the LoRA-based system-level adaptation described in Section 3.3.

3.3 LoRA-Based System-Level Adaptation

The transition from component to system-level prognostics requires an adaptation strategy that can handle the increased structural complexity without discarding the degradation physics encoded during pre-training. Conventional full fine-tuning risks catastrophic forgetting and overfitting under scarce system-level data, as discussed in Subsection 2.2.3. LoRA addresses both failure modes simultaneously: the pre-trained weights are kept frozen, and a compact, trainable perturbation, expressed as the product

of two low-rank matrices, is injected in parallel with each frozen weight matrix. The resulting update is constrained to a low-dimensional subspace, acting as a strong implicit regulariser that preserves the learned degradation features while allowing the model to capture the coupling behaviour of the interconnected coupon assembly.

In the present implementation, a single shared LoRA adapter is applied across all three coupon branches. Sharing a single adapter enforces a common adaptation subspace grounded in the coupon-level degradation physics, and further reduces the number of trainable parameters, a critical design choice under the severe data scarcity of the system-level regime, while still maintaining the ability to process each coupon's degradation sequence independently.

3.3.1 Mathematical Formulation of LoRA

Let $W_0 \in \mathbb{R}^{d \times k}$ denote a pre-trained weight matrix of the dense layers following the LSTM backbone. Instead of updating W_0 directly, LoRA constrains the weight update ΔW to a low-rank decomposition:

$$W = W_0 + \Delta W = W_0 + \frac{\alpha}{r}BA, \quad (3.2)$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and the rank $r \ll \min(d, k)$ is a hyperparameter that governs the capacity of the adaptation. The scaling constant α is fixed to 16 in the present implementation, and the term α/r facilitates tuning of the effective learning rate relative to the rank. The matrices are initialised such that $\Delta W = 0$ at the start of fine-tuning: A is initialised using Glorot uniform initialisation to ensure stable gradient flow, and B is set to zero, so that the adapted model is identical to the pre-trained model at initialisation. Only A and B are updated during fine-tuning; W_0 remains frozen throughout. This reduces the number of trainable parameters per adapted matrix from $|\theta_{\text{full}}| = dk$ to $|\theta_{\text{LoRA}}| = r(d + k)$.

A LoRA adapter is applied specifically to the intermediate dense layer that follows the LSTM backbone, shared across all three coupon branches, refining the extracted temporal features for system-level estimation. This design choice is motivated by the distinct roles of the two layers: the LSTM recurrent weights encode the temporal dynamics of fatigue degradation and are expected to remain valid across both the coupon and system domains, as the underlying physics of aluminium fatigue are shared. The dense layer, by contrast, performs a task-specific mapping from temporal features to RUL estimates, a mapping that must account for the structural interdependencies and load redistribution effects present at the system level but absent at the coupon level. Confining adaptation to this layer therefore preserves the domain-invariant degradation representations encoded in the LSTM while allowing the model to recalibrate its output projection for the system-level

task.

3.3.2 System-Level Architecture and Fine-tuning

The overall architecture for system-level adaptation is illustrated in Figure 3.2. Each of the three system components is processed by a dedicated input branch. All three branches share the same frozen pre-trained weights W_0 and the same set of LoRA adaptation matrices (A , B) through a single shared backbone. This design preserves a common adaptation subspace grounded in the coupon-level degradation physics encoded in W_0 , while the distinct input sequences from each structural position allow the model to produce differentiated probabilistic outputs.

Each branch produces a probabilistic output $(\hat{\mu}_c, \hat{\sigma}_c^2)$ for component $c \in \{1, 2, 3\}$, parametrising a Gaussian distribution over the individual component RUL. The three outputs are then concatenated and passed through a trainable system aggregator, a feed-forward network, that maps the joint component-level predictions to a system-level Gaussian output $(\hat{\mu}_{\text{sys}}, \hat{\sigma}_{\text{sys}}^2)$. The network's weights are trained by minimising the NLL loss defined in Eq. (2.19), where the ground-truth labels y_i are defined according to the failure criterion in use (Min or Max, as described in Section 3.5). Training is performed with the Adam optimiser; the rank r and the learning rate are treated as hyperparameters and selected through a Bayesian optimisation process.

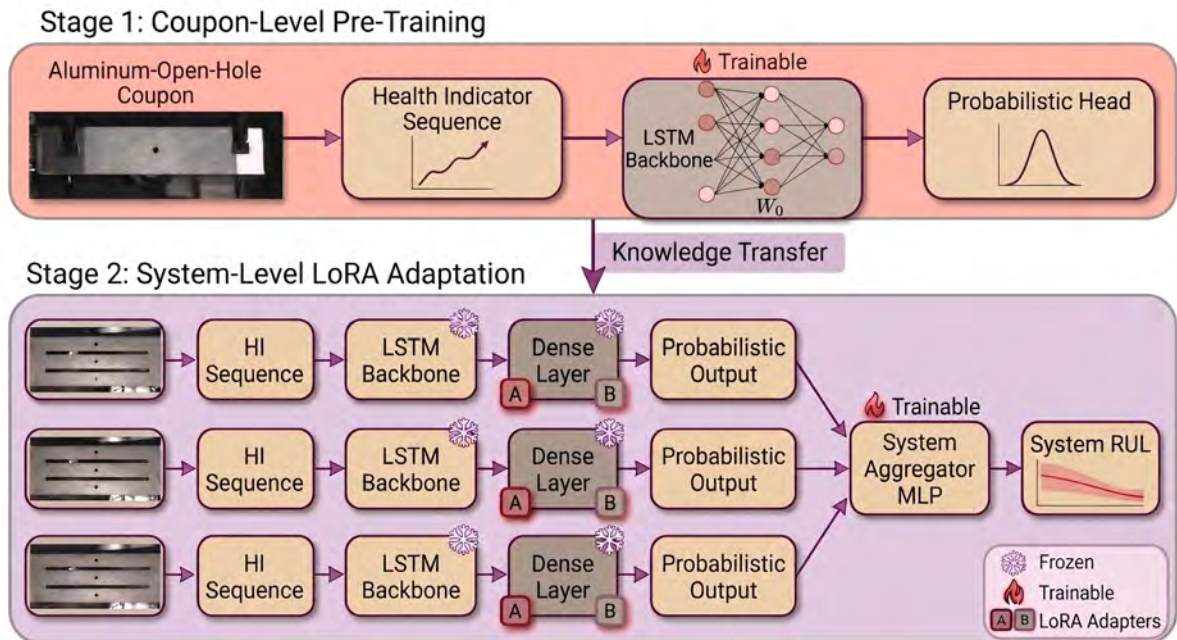


Figure 3.2: Architecture of the LoRA-based system-level adaptation framework.

3.4 Uncertainty Quantification

Reliable prognostics demand not only accurate point estimates of the Remaining Useful Life, but also a principled quantification of the confidence associated with each prediction. In safety-critical structural applications, an overconfident model that underestimates its own uncertainty can be more dangerous than a less accurate one. The present framework decomposes predictive uncertainty into aleatoric and epistemic components, as formally introduced in Section 2.4 and propagates both through to the final system-level RUL estimate. The same strategy is applied consistently at both the coupon and system levels.

Tracking both components separately, across coupon pre-training and LoRA-based system-level adaptation, enables a transparent assessment of how parameter-efficient fine-tuning influences each source of uncertainty as system-level observations are incrementally introduced.

3.4.1 Aleatoric Uncertainty via Probabilistic Neural Network

Aleatoric uncertainty, as formally defined in Section 2.4, reflects irreducible variability in the data-generating process and cannot be reduced by collecting more data.

To capture this source of uncertainty, a PNN is adopted in place of a conventional point-prediction head [21]. Rather than producing a single scalar RUL estimate, the network outputs the parameters of a Gaussian distribution over the RUL, the mean $\hat{\mu}$ and the variance $\hat{\sigma}^2$ and is trained to minimise the NLL loss defined in Eq. (2.19). The NLL loss consists of two competing terms: the first penalises prediction error scaled by the inverse variance, while the second penalises overconfidence by preventing the model from inflating $\hat{\sigma}^2$ arbitrarily. The model is therefore incentivised to report high uncertainty precisely where its predictions are unreliable, and the predicted variance $\hat{\sigma}^2$ thus provides a data-driven, input-dependent estimate of aleatoric uncertainty, capturing the heteroscedastic nature of structural degradation.

3.4.2 Epistemic Uncertainty via Monte Carlo Dropout

Epistemic uncertainty, as defined in Section 2.4, reflects model ignorance due to limited training data and is reducible as more observations become available — particularly relevant here given the severe system-level data scarcity.

Epistemic uncertainty is estimated through MC Dropout, a computationally efficient approximation to Bayesian inference, as introduced in Section 2.4. Concretely, for a given input, $T = 100$ stochastic forward passes are performed with active Dropout, yielding a set of mean and variance predictions $\{(\hat{\mu}^{(t)}, \hat{\sigma}^{2(t)})\}_{t=1}^T$. The epistemic uncertainty is then estimated following Eq. (2.20), where the Monte Carlo mean $\bar{\mu} = \frac{1}{T} \sum_{t=1}^T \hat{\mu}^{(t)}$ also serves

as the final point prediction.

3.4.3 Total Predictive Uncertainty and Confidence Intervals

The aleatoric and epistemic components are recombined under the law of total variance, yielding a single calibrated measure of predictive confidence [29]. In general form, for an aggregation operator $f(\cdot)$ applied across C component-level Gaussian predictions, the total predictive variance is decomposed as:

$$\sigma_{\text{total}}^2 = \underbrace{\mathbb{E}[\sigma_a^2]}_{\text{Aleatoric}} + \underbrace{\text{Var}[\mu_\theta]}_{\text{Epistemic}} \quad (3.3)$$

This decomposition provides actionable insight into the source of uncertainty at each point in the degradation trajectory: high epistemic uncertainty signals that additional system-level data would reduce model ignorance, whereas high aleatoric uncertainty reflects irreducible fatigue scatter that no amount of additional data can eliminate, an important distinction for maintenance decision-making. This formulation is later instantiated in its component-specific form as Eq. (4.3) (Section 4.1).

The total predictive standard deviation $\hat{\sigma}_{\text{total}}$ is obtained from Eq. (3.3), and a symmetric 95% predictive interval is constructed as:

$$\text{lower} = \hat{\mu}_{\text{total}} - 1.96 \cdot \hat{\sigma}_{\text{total}} \quad (3.4)$$

$$\text{upper} = \hat{\mu}_{\text{total}} + 1.96 \cdot \hat{\sigma}_{\text{total}} \quad (3.5)$$

following the same general form later specialised as Eq. (4.5) (Section 4.1). This procedure is applied identically at both the coupon and system levels, ensuring that the uncertainty estimates reported for the SRUL reflect the full propagation of predictive confidence through the shared backbone, the LoRA adaptation, and the system aggregator.

It should be noted that, since the framework employs a Gaussian NLL loss, the resulting predictive distribution is defined over the entire real line. Consequently, as the predicted RUL approaches zero near end-of-life, the lower bound of the predictive interval may extend into negative values, which are physically inadmissible. This is a known limitation of Gaussian-based probabilistic regression in the context of RUL estimation, where the target variable is strictly non-negative by definition.

3.5 System-Level RUL Aggregation

The definition of system failure is a design choice that depends on the operational context and safety philosophy. In general, the SRUL is derived by applying an aggregation operator $f(\cdot)$ to the predicted mean RULs of the individual components, $\hat{\mu}(t) =$

$[\hat{\mu}_1(t), \dots, \hat{\mu}_C(t)]$, where C is the total number of components:

$$\widehat{\text{SRUL}}(t) = f(\hat{\mu}_1(t), \dots, \hat{\mu}_C(t)). \quad (3.6)$$

While $f(\cdot)$ can represent various statistical functions such as the mean or median, two complementary criteria are adopted in this work depending on the assumed system architecture [30]: the *minimum* operator (Min criterion), applied when any single component failure compromises the entire assembly (series configuration), and the *maximum* operator (Max criterion), applied when the system remains operational until all components have failed (parallel or redundant configuration).

The associated system-level predictive interval is constructed by applying the general procedure defined in Eq. (3.5) to the aggregated system output $(\hat{\mu}_{\text{sys}}, \hat{\sigma}_{\text{sys}}^2)$ produced by the system aggregator, ensuring that the uncertainty propagated through the shared backbone and the LoRA adaptation layers is consistently reflected in the final SRUL estimate.

Chapter 4

Baseline Models and Metrics

To evaluate the effectiveness of the proposed LoRA-based system aggregator, three baseline models are considered. These benchmarks represent non-adaptive, standard transfer learning, and unconstrained fine-tuning approaches [31, 32], providing lower and upper bounds and allowing performance isolation for the proposed lightweight adaptation strategy. The predictive performance of all considered frameworks is assessed across a comprehensive set of deterministic, probabilistic, and uncertainty-oriented metrics, described at the end of this section.

4.1 Zero-Shot Baseline

The Zero-Shot Baseline evaluates SRUL prediction without any system-level training or parameter adaptation on the system domain. Instead, it directly uses the component-level degradation knowledge captured by the pre-trained networks.

As shown in Fig. 4.1, the health index (HI) sequence of each component $i \in \{1, 2, 3\}$ is processed independently by a pre-trained branch consisting of an LSTM layer followed by a Dense layer. Each branch produces a probabilistic estimate of the component RUL, defined by a mean μ_i and a standard deviation σ_i :

$$\text{Out}_i = [\mu_i, \sigma_i] \quad (4.1)$$

These component-level predictions are then passed to an aggregation block. To combine them into a SRUL estimate without further training, a non-parametric aggregation function f is applied to the component means:

$$\mu_{\text{total}} = f(\mu_1, \mu_2, \mu_3), \quad (4.2)$$

where the choice of f is not fixed but depends on the assumed system topology and is left to the practitioner. For a *series* configuration, where failure of any single component

leads to system failure, $f = \min$ is the appropriate choice. For a *parallel* (fully redundant) configuration, where the system remains operational as long as at least one component is functional, $f = \max$ is used instead. This design decision should reflect the physical architecture of the monitored system.

The total predictive variance is decomposed into an aleatoric and an epistemic component:

$$\sigma_{\text{total}}^2 = \underbrace{\text{Var}(f(\mathbf{S}))}_{\text{Aleatoric}} + \underbrace{f(\sigma_1^2, \sigma_2^2, \sigma_3^2)}_{\text{Epistemic}}, \quad (4.3)$$

where $\mathbf{S}$ denotes samples drawn from the component Gaussian distributions, used to estimate the variance of f applied to their realisations. The aleatoric term captures the variability introduced by aggregating uncertain component predictions, while the epistemic term is conservatively set to the largest individual component uncertainty, reflecting the worst-case model confidence among the three branches. Consistently with the aggregation rule, f in the aleatoric term is instantiated as min or max according to the chosen system topology.

Finally, the aggregated mean and standard deviation are used to construct the 95% predictive confidence interval:

$$\text{lower} = \mu_{\text{total}} - 1.96 \cdot \sigma_{\text{total}} \quad (4.4)$$

$$\text{upper} = \mu_{\text{total}} + 1.96 \cdot \sigma_{\text{total}} \quad (4.5)$$

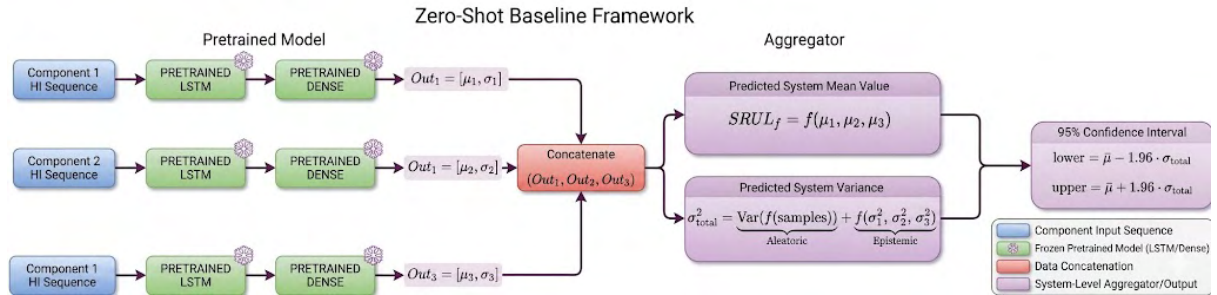


Figure 4.1: Architectural pipeline of the Zero-Shot Baseline framework.

4.2 Standard Transfer Learning Baseline

To evaluate the contribution of the proposed LoRA mechanism, a standard transfer learning baseline is introduced. This setup follows a typical feature-extraction approach [33, 34, 35], where the pre-trained component-level network is reused without updating its weights, and a trainable multi-layer perceptron (MLP) aggregator is added

on top.

Let the pre-trained component model be denoted as $f(x; \Theta_{pre})$, with parameters:

$$\Theta_{pre} = \{\Theta_{LSTM}, \Theta_{Dense}, \Theta_{\mu}, \Theta_{var}\} \quad (4.6)$$

To ensure a fair comparison with the proposed model, the same dual Monte Carlo (MC) Dropout scheme is used, applied after the LSTM layer and the intermediate dense layer. During system-level training, all backbone parameters remain frozen:

$$\Theta_{pre} \leftarrow \text{Frozen} \quad (4.7)$$

As a result, no gradients are propagated through the pre-trained weights:

$$\frac{\partial \mathcal{L}}{\partial \Theta_{pre}} = 0 \quad (4.8)$$

where $\mathcal{L}$ is the Gaussian negative log-likelihood loss.

As illustrated in Fig. 4.2, the frozen backbone processes each component's health index sequence independently, producing a mean and variance for each component.

The component outputs are then combined into a feature vector z , which serves as input to the aggregator. The structure of z depends on the assumed failure mode. For a series system, the minimum predicted mean is included as an additional feature to highlight the weakest component. For a parallel system, the maximum mean is used instead:

$$\mu_{criterion} = \begin{cases} \min(\mu_1, \mu_2, \mu_3) & \text{series assumption} \\ \max(\mu_1, \mu_2, \mu_3) & \text{parallel assumption} \end{cases} \quad (4.9)$$

$$z = [o_1, o_2, o_3, \mu_{criterion}] \quad (4.10)$$

where $o_i = [\mu_i, \sigma_i^2]$ is the probabilistic output of the i -th component.

Adaptation to the system-level task is handled only by the MLP aggregator $g(z; \Phi_{agg})$, which consists of dense layers with Dropout for regularization. The final system-level prediction is:

$$[\mu_{sys}, \sigma_{sys}^2] = g(z; \Phi_{agg}) \quad (4.11)$$

where Φ_{agg} are the only trainable parameters. This baseline allows us to assess whether a system-level model can learn the overall behavior using fixed component representations.

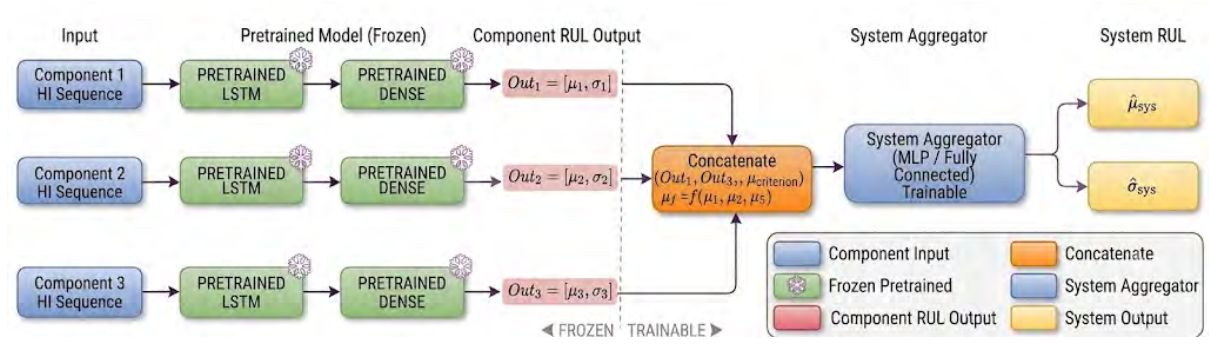


Figure 4.2: Architectural pipeline of the Standard Transfer Learning Baseline, showing the frozen component branches and the trainable MLP aggregator.

4.3 Full Fine-Tuning Baseline

To isolate the contribution of the low-rank adapter constraint introduced by LoRA, a full fine-tuning baseline is considered. In contrast to the proposed LoRA framework, this approach unfreezes all pre-trained model weights during system-level adaptation, allowing the LSTM layer, the intermediate Dense layer, and the probabilistic output heads to be updated without any low-rank constraints [34]. The number of trainable parameters is therefore maximal, equal to the full parameter count of the pre-trained backbone plus the system aggregator.

The architecture mirrors that of the LoRA framework: each of the three component branches processes its HI sequence through the shared backbone, producing probabilistic outputs that are subsequently passed to a trainable MLP aggregator. Training is performed using the same NLL loss, Adam optimiser, and Bayesian hyperparameter optimisation procedure described in Subsection 3.3.2, ensuring a fair and controlled comparison. The same dual MC Dropout uncertainty decomposition is applied at inference time across all branches.

This baseline is designed to answer a specific question: under severe system-level data scarcity, does the low-rank constraint of LoRA act as a meaningful implicit regulariser, or is the performance gain of the proposed framework simply attributable to the introduction of additional trainable parameters? The comparison between this baseline and the LoRA framework is presented in Subsection 6.3.3.

4.4 Evaluation Metrics

To evaluate the predictive performance of the considered frameworks, the models are assessed across a comprehensive set of deterministic, probabilistic, and uncertainty-oriented metrics.

Point-estimate metrics. Point-estimate accuracy is quantified using the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE), defined respectively as:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|, \quad \text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}, \quad (4.12)$$

where $\hat{y}_i$ is the predicted RUL and y_i the corresponding ground truth for the i -th test sample. MAE treats all errors equally regardless of magnitude, while RMSE penalises large deviations more heavily due to its quadratic nature, making it particularly sensitive to outliers. Both metrics are reported in normalised form as well as in operational cycles to facilitate physical interpretation.

Continuous Ranked Probability Score (CRPS). Unlike MAE and RMSE, which evaluate only point predictions, the CRPS assesses the entire predictive distribution. Intuitively, it measures how well the predicted probability distribution covers the true outcome: a distribution that is both sharp (narrow) and well-centred around the true value will achieve a low CRPS, while a distribution that is overconfident, underconfident, or systematically biased will be penalised.

Formally, for a probabilistic forecast expressed as a cumulative distribution function F and a scalar observation y , the CRPS is defined as:

$$\text{CRPS}(F, y) = \int_{-\infty}^{+\infty} (F(x) - \mathbf{1}[x \geq y])^2 dx, \quad (4.13)$$

where $\mathbf{1}[x \geq y]$ is an indicator function equal to 1 if $x \geq y$ and 0 otherwise. Geometrically, this corresponds to the squared area enclosed between two CDF curves: the predicted distribution $F(x)$ and the empirical CDF of the observation, which is a Heaviside step function that jumps from 0 to 1 at $x = y$. When the prediction is perfect, the two curves coincide and the CRPS equals zero. The further the predicted distribution is from the true value — either due to bias or excessive spread — the larger the enclosed area and thus the higher the CRPS. A key property of the CRPS is that it is a proper scoring rule: it is minimised in expectation only when F is the true data-generating distribution, ensuring that reliable probabilistic forecasting is always incentivised [36].

In this work, the predictive CDF F is not a single Gaussian but a *Gaussian mixture*, constructed by averaging the CDFs of S stochastic forward passes obtained via Monte Carlo (MC) Dropout:

$$F(x) = \frac{1}{S} \sum_{s=1}^S \Phi\left(\frac{x - \mu_s}{\sigma_s}\right), \quad (4.14)$$

where μ_s and σ_s are the predicted mean and standard deviation of the s -th forward pass, and Φ denotes the standard normal CDF. This formulation is richer than a single Gaus-

sian: the spread of the individual σ_s values captures *aleatoric* uncertainty (irreducible noise in the data), while the disagreement among the μ_s samples captures *epistemic* uncertainty (model uncertainty due to limited data). Since no closed form exists for the CRPS of a Gaussian mixture, it is evaluated numerically over a fine grid spanning the normalised RUL domain $[-0.1, 1.1]$. As with the point-estimate metrics, the CRPS is reported in both normalised form and in operational cycles.

Coverage probability. The coverage of the 95% predictive confidence interval is reported to assess calibration independently of sharpness. For each test sample, the interval is constructed as $[\mu_{\text{sys}} - 1.96 \sigma_{\text{sys}}, \mu_{\text{sys}} + 1.96 \sigma_{\text{sys}}]$, where μ_{sys} and σ_{sys} are the aggregated system-level mean and standard deviation. Coverage is then defined as the fraction of test samples whose true RUL falls within this interval:

$$\text{Coverage} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}[y_i \in [\mu_i - 1.96 \sigma_i, \mu_i + 1.96 \sigma_i]]. \quad (4.15)$$

Ideal calibration corresponds to a coverage of 0.95. Values below this threshold indicate overconfident predictions (intervals too narrow), while values above it indicate underdispersed uncertainty estimates (intervals too wide).

Weighted Mean Width of Uncertainty (WSU). While coverage ensures that true RUL values fall within the predicted bounds, it can be trivially satisfied by a model that outputs arbitrarily wide intervals. The Weighted Mean Width of Uncertainty (WSU) complements coverage by measuring the average width of the predicted confidence intervals across the degradation trajectory, penalising overly conservative bounds and rewarding sharp, informative predictions. Despite its intuitive appeal, the WSU was not adopted as an evaluation metric in the present framework. The primary reason is that sharpness and calibration are already jointly assessed by the CRPS, which is a proper scoring rule that simultaneously penalises both distributional bias and excessive spread. Reporting the WSU alongside the CRPS would therefore introduce redundancy without additional discriminative power. Furthermore, under the Gaussian likelihood formulation adopted here, the interval width is a direct function of the predicted standard deviation $\hat{\sigma}$, which is already reflected in the CRPS value. For these reasons, the CRPS is preferred as the single probabilistic metric that subsumes the information conveyed by both coverage and sharpness in a theoretically rigorous and non-redundant manner [36].

Chapter 5

Case Study

To maintain terminological consistency throughout this chapter, the following conventions are adopted: the triplet or specimen, comprising three interconnected coupons, is treated as the system, while each individual coupon is treated as a component. This mapping is preserved across all subsequent sections.

5.1 Experimental Campaign

The validation of the proposed framework is based on fatigue data obtained from aluminum 7075-T6 open-hole specimens, a material selected for its extensive use in aerospace structural applications. The experimental campaign was designed following a hierarchical approach to evaluate the transition from component-level to system-level degradation. The coupon level dataset consists of five single open-hole coupons, while the system level set comprises six multi-coupon specimens, each formed by three identical interconnected coupons. These higher-level assemblies were designed to introduce mechanical coupling and load-sharing effects, representing a simplified redundant structural system.

All fatigue experiments were conducted using an MTS servo-hydraulic testing machine with a capacity of 60 kN. Testing was performed under tension–tension load control with a load ratio of 0.1 and a frequency of 5 Hz. To maintain consistent stress levels across different scales, the maximum cyclic load was set at 9 kN for the single coupons and proportionally increased to 27 kN for the multi-coupon specimens. The loading sequence was organized in repeated blocks of 250 cycles, followed by a brief hold period at the maximum load to facilitate high-resolution data acquisition and surface imaging. Failure for all tests was defined by a global displacement threshold of 13 mm, ensuring a consistent end-of-life criterion across the dataset.

Condition monitoring was primarily conducted through Digital Image Correlation (DIC), providing full-field surface strain measurements. The specimens were prepared with a matte white base and a black speckle pattern, with images captured by 5 MP

industrial cameras during the hold phases of the loading cycle. For the purposes of this study, the axial strain component along the vertical direction was selected as the principal health indicator. This choice is justified by the direct correlation between localized strain concentration around the central hole and the progression of fatigue cracks.

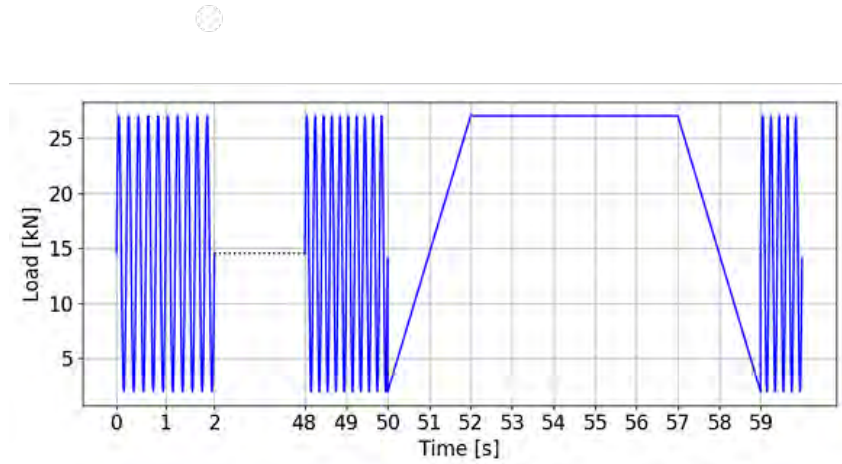


Figure 5.1: Overview of the applied loading conditions and experimental profile of the fatigue machine.

5.2 Dataset Description

The data processing phase focused on extracting the axial strain profiles in the vicinity of the stress concentrators. Out of the eight manufactured specimens, six yielded valid datasets and were included in the analysis (Specimens 3–8), while two (Specimens 1 and 2) were excluded due to signal irregularities that violated the monotonicity assumption required for reliable health indicator construction. The extracted strain histories provide a monotonic proxy for damage accumulation, serving as the primary input for the RUL prediction models. By focusing on the localized mechanical response, the framework captures the critical degradation features necessary for reliable prognostic performance at both the coupon and system levels.

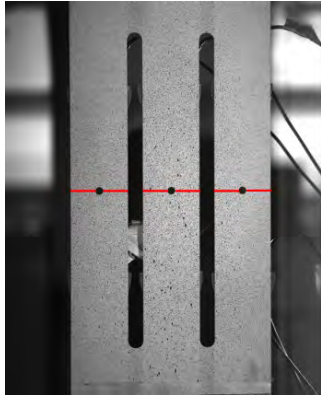


Figure 5.2: Location of the extracted strain data of the specimens.

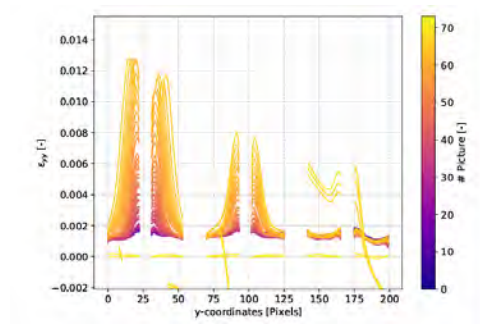


Figure 5.3: Extracted strain data from DIC measurement, specimen 5.

5.2.1 Coupon-Level Data

The coupon-level dataset comprises five single open-hole coupons, providing the foundational training data for the component-level prognostic model, as depicted in Figure 5.4. For each coupon, the axial strain history was extracted from the DIC measurements in the vicinity of the central hole, capturing the progressive strain accumulation from crack initiation through to final failure. The resulting strain profiles exhibit a monotonically increasing trend, reflecting the underlying fatigue damage accumulation process and serving as a reliable health indicator for RUL estimation at the component scale.

5.2.2 System-Level Data

The system-level dataset comprises the six valid specimens, each consisting of three interconnected single open-hole coupons, as depicted in Figure 5.5. For each specimen, the DIC-derived axial strain history of every constituent coupon was extracted individually, yielding three parallel component-level health indicator sequences per system test. This structure directly reflects the hierarchical nature of the proposed framework, where system-level degradation is inferred from the concurrent evolution of its constituent component health states. The strain histories across the three coupons within each specimen exhibit correlated yet distinct degradation trajectories, a consequence of the mechanical coupling and load redistribution inherent to the multi-coupon configuration.

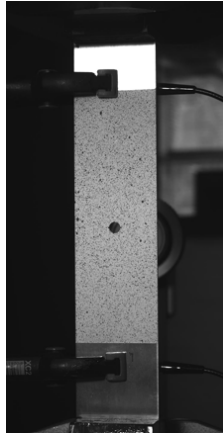


Figure 5.4: Coupon/Component



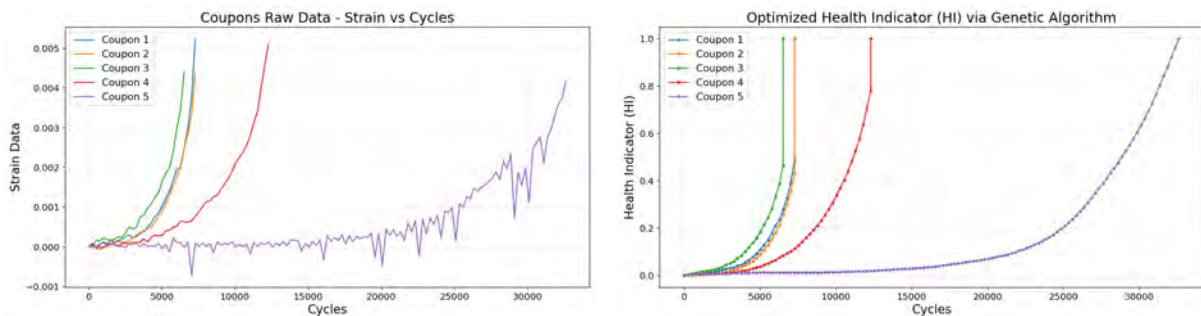
Figure 5.5: Triplet/System

Chapter 6

Results and Discussion

6.1 Coupon-Level RUL Prediction

The coupon-level prognostic model is formulated using a LSTM architecture with probabilistic outputs, designed to predict the RUL of individual open-hole coupons. The model operates on the constructed Health Indicator (HI), which serves as a compact representation of the degradation process and is used as the input time series to the network. The extracted axial strain histories and the resulting HI profiles for all coupon specimens are illustrated in Figures 6.1a and 6.1b respectively. Instead of producing deterministic outputs, the model predicts the parameters of a Gaussian distribution at each time step, namely the mean and variance. Training is performed by minimizing the Gaussian NLL loss, enabling the model to learn accurate point estimates and associated predictive uncertainty.



(a) Strain signals for the five coupons.

(b) Constructed Health Indicator signals for the five coupons.

Figure 6.1: Coupon-level raw strain data and optimised Health Indicator signals derived via Genetic Algorithm.

The HI used as input to the LSTM was constructed via the GA-based fusion procedure described in Section 3.1. Specifically, the candidate HI was formed as a weighted linear combination of $P = 5$ strain-derived features, each computed from the raw axial strain

signal up to the current cycle:

- **Range:** the cumulative peak-to-peak amplitude of the strain signal, $\max(\varepsilon_{1:k}) - \min(\varepsilon_{1:k})$, capturing the progressive widening of strain excursions as damage accumulates.
- **RMS:** the root-mean-square value of the strain history up to cycle k , reflecting the cumulative energy content of the signal.
- **Mean:** the running average strain, tracking the gradual shift in the baseline strain level.
- **Cumulative sum:** the running integral of the strain signal, $\sum_{i=1}^k \varepsilon_i$, representing the total accumulated strain exposure.
- **Quantile slope:** the interquartile range (90th–10th percentile) of the strain history normalised by the cycle interval, capturing the rate of spread of the strain distribution.

These five features were selected because, individually, each exhibited a favourable trend with respect to monotonicity and prognosability across the coupon population, and their combination via the GA (Section 3.1) was optimised to jointly maximise these two criteria, yielding the fitness-maximising weight vector α used to fuse them into the final HI. The GA-optimised weights were determined via 200 generations of evolutionary search over a population of 50 candidate solutions.

Model evaluation is conducted using a Leave-One-Out Cross-Validation (LOOCV) scheme. This approach provides a statistically robust and unbiased estimate of generalization performance.

Uncertainty quantification is achieved via Monte Carlo Dropout inference. Multiple stochastic forward passes through the network are performed at test time, yielding an ensemble of predictions. This ensemble is used to decompose the predictive uncertainty into aleatoric (data-related) and epistemic (model-related) components, providing additional insight into the reliability of the predictions. The resulting RUL predictions with associated uncertainty bounds for all coupon folds are shown in Figure 6.2.

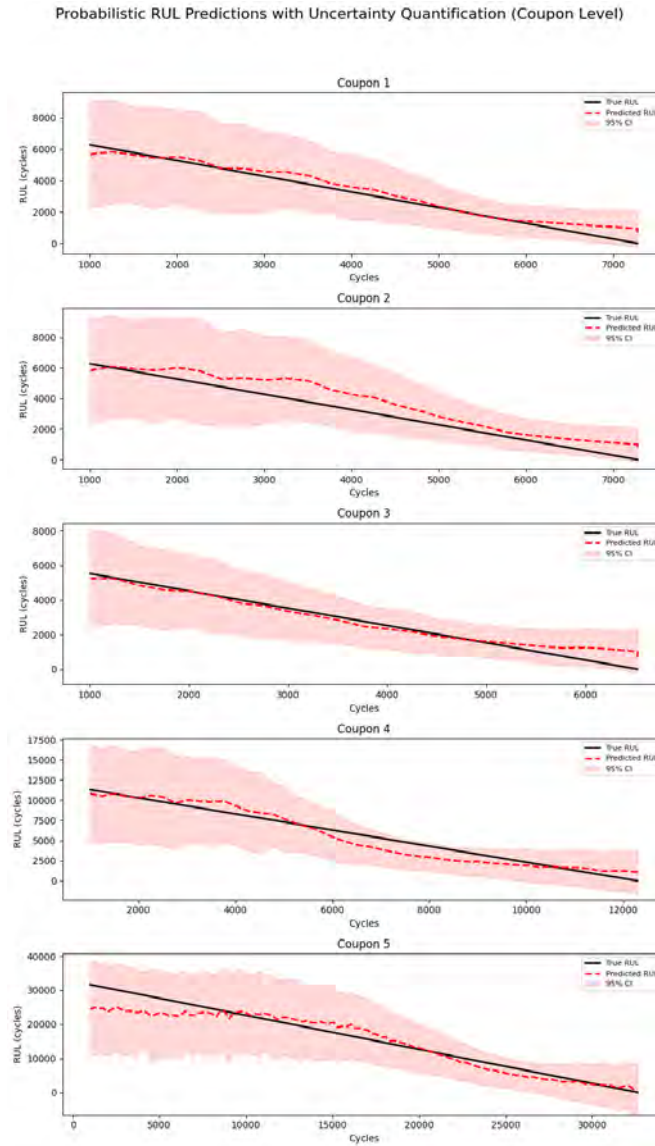


Figure 6.2: RUL predictions with uncertainty quantification for the five coupons.

The hyperparameters of the LSTM framework were optimized using Bayesian optimization through the Optuna framework, with CRPS as the minimization objective. Considering the relatively small dataset, a shallow architecture was adopted to reduce the risk of overfitting. The network consists of a single LSTM layer with 16 units, followed by a Dense layer with 8 neurons for feature transformation. The selected hyperparameters are summarized in Table 6.1.

Table 6.1: Optimized hyperparameter configuration for the coupon-level LSTM.

Hyperparameter	Value
LSTM Units	16
Dense Units	8
Dropout Rate(2)	0.3
Learning Rate	2.49×10^{-5}

Following the completion of the LOOCV evaluation, a final backbone model is trained on the full coupon-level dataset. This backbone model serves as a pre-trained feature extractor for subsequent system-level transfer learning experiments.

6.2 System Level RUL Prediction Performance

The predictive performance of the three approaches, the Zero-Shot Baseline, the Standard Transfer Learning Baseline, and the proposed LoRA-based framework, is evaluated under two training regimes: a low-data regime using a single system-level training history, and a moderate-data regime using five system-level training histories. The Zero-Shot Baseline does not utilize any system-level training data and serves as a fixed reference, while both the Standard Transfer Learning Baseline and the LoRA framework are trained on the available system-level histories. To ensure a fair comparison, all three frameworks share the exact same probabilistic LSTM backbone pre-trained on the coupon-level dataset (architecture detailed in Subsection 6.1 and Table 6.1). MC Dropout remains active at test time across all frameworks, with 100 stochastic forward passes.

Model development followed a two-stage optimization procedure. In the first stage, the architectural configuration of each framework was determined via Bayesian Optimization through Optuna, covering layer sizes, dropout rates, and aggregator structure. A key distinction among the frameworks is how the failure criterion is applied: in the Zero-Shot Baseline it is computed post-inference via a simple `min/max` on the predictions, while in the Standard Transfer Learning and LoRA frameworks it is embedded within the computational graph, allowing the aggregator to account for the system-level failure definition during training. The resulting architectural configurations are summarized in Table 6.2.

Table 6.2: Architectural configurations for the three frameworks.

Hyperparameter	Zero-Shot	Standard TL	LoRA
Frozen Core Units	LSTM (16), Dense (8)	LSTM (16), Dense (8)	LSTM (16), Dense (8)
Dropout Rate ($\times 2$)	0.3	0.3	0.3
Failure Logic	Post-inference	Aggregator input	Aggregator input
Aggregator Dense Layers	–	$64 \rightarrow 32 \rightarrow 16$	$64 \rightarrow 32 \rightarrow 16$
Aggregator Dropout	–	0.2	0.2
Trainable Parameters	0	Φ_{agg} only	$\Phi_{agg} + \text{LoRA } (A, B)$

In the second stage, the training-related hyperparameters, namely learning rate, batch size, lookback window, and LoRA rank in the case of the LoRA-based framework, were optimized separately for each combination of framework, failure criterion, and training regime, again using Optuna with CRPS as the objective. The hyperparameter values reported in the following table are provided for the LoRA framework, corresponding to the five training histories regime under the Max criterion. Results for both the single and

five training histories regimes, and for both the Min and Max criteria, are presented in the subsequent subsections for direct comparison.

Table 6.3: Optimized hyperparameter configuration for the proposed LoRA framework (MAX criterion, five training histories).

Hyperparameter	Value
Learning Rate	2.49×10^{-5}
Batch Size	16
Lookback Window	5
LoRA Rank	24
LoRA Alpha (α)	16

6.2.1 Min Criterion

Under the Min Criterion, overall system failure is triggered as soon as the first individual component undergoes critical degradation. Consequently, the true RUL of the system is strictly bounded by the shortest lifetime among the triplet components. Figure 6.3 illustrates the RUL predictions obtained by the Zero-Shot Baseline under the Min criterion, and the corresponding performance metrics are reported in Table 6.4. Without any system-level training, the baseline achieves a mean MAE of 0.1301 and a mean CRPS of 0.0955, with a coverage of 1.000, indicating that while the uncertainty intervals are sufficiently wide to capture all true RUL values, the point estimate accuracy remains limited. This behaviour is expected, as the Zero-Shot Baseline applies the Min criterion post-inference without any adaptation to the system-level degradation patterns, and therefore serves as a lower-bound reference for the trained frameworks.

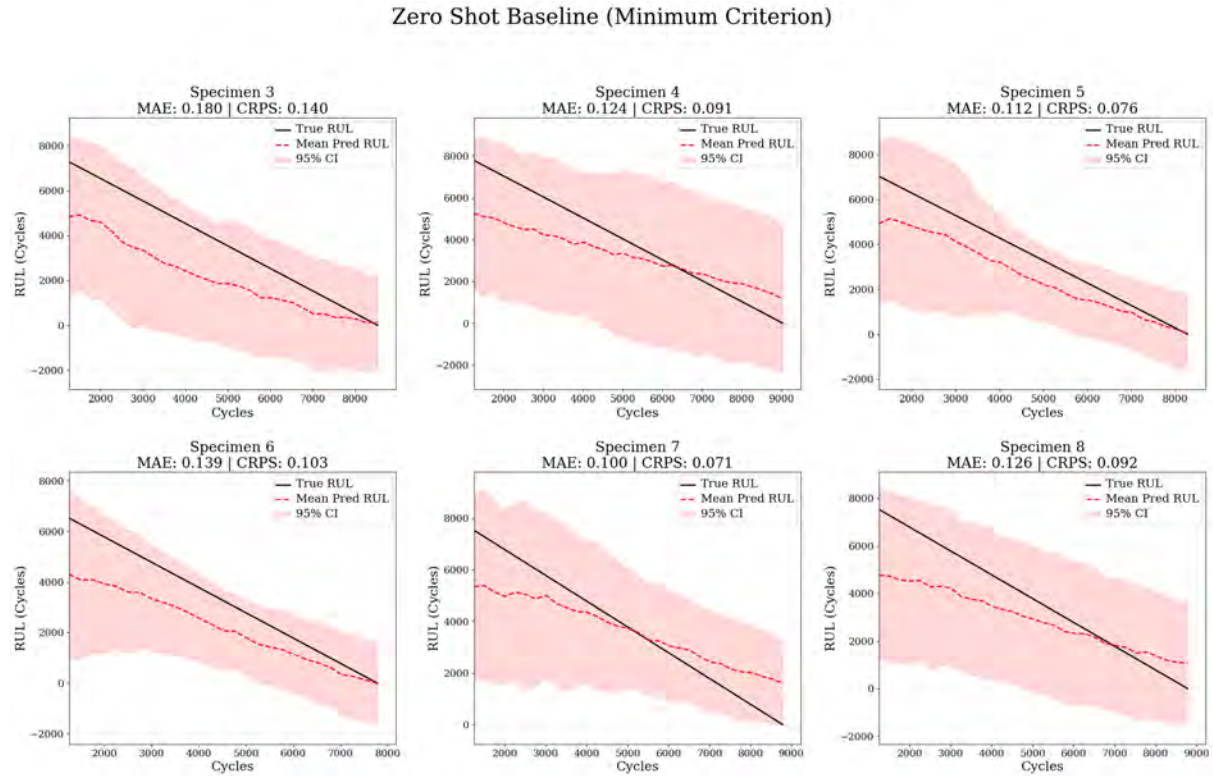


Figure 6.3: RUL predictions of the Zero-Shot Baseline under the Min criterion.

Table 6.4: Predictive performance of the Zero-Shot Baseline under the Min criterion.

Metric	Mean	Std
MAE	0.1301	0.0254
MAE (cycles)	1107.8	212.3
RMSE	0.1511	0.0252
RMSE (cycles)	1287.6	215.9
CRPS	0.0955	0.0226
CRPS (cycles)	813.0	189.4
Coverage	1.0000	0.0000
Aleatoric	0.02108	0.00873
Epistemic	0.00726	0.00074

Figures 6.4 and 6.5 present the RUL predictions of the Standard Transfer Learning Baseline and the LoRA framework respectively, both trained on a single system-level training history, with the corresponding metrics reported in Table 6.5. With a single training history, both frameworks improve upon the Zero-Shot Baseline. The Standard Transfer Learning Baseline achieves a mean MAE of 0.1283 and CRPS of 0.0925, offering a marginal improvement over the zero-shot configuration. The LoRA framework, however, achieves a mean MAE of 0.1078 and CRPS of 0.0778, representing reductions of approximately 17% and 19% respectively relative to the Zero-Shot Baseline, and approximately 16% and 16% relative to the Standard Transfer Learning Baseline. Notably,

the LoRA framework also exhibits substantially lower standard deviations across all metrics, suggesting more consistent performance across the evaluation folds. Within the LoRA framework, the aleatoric uncertainty (0.03225) substantially exceeds the epistemic component (0.00883), consistent with the pattern observed under the Min criterion (Section 6.2.1), and reflecting the inherent variability of the parallel failure process.

Comparison: Non-LoRA vs. LoRA Framework — Trained on Specimen 4 (Minimum Criterion)

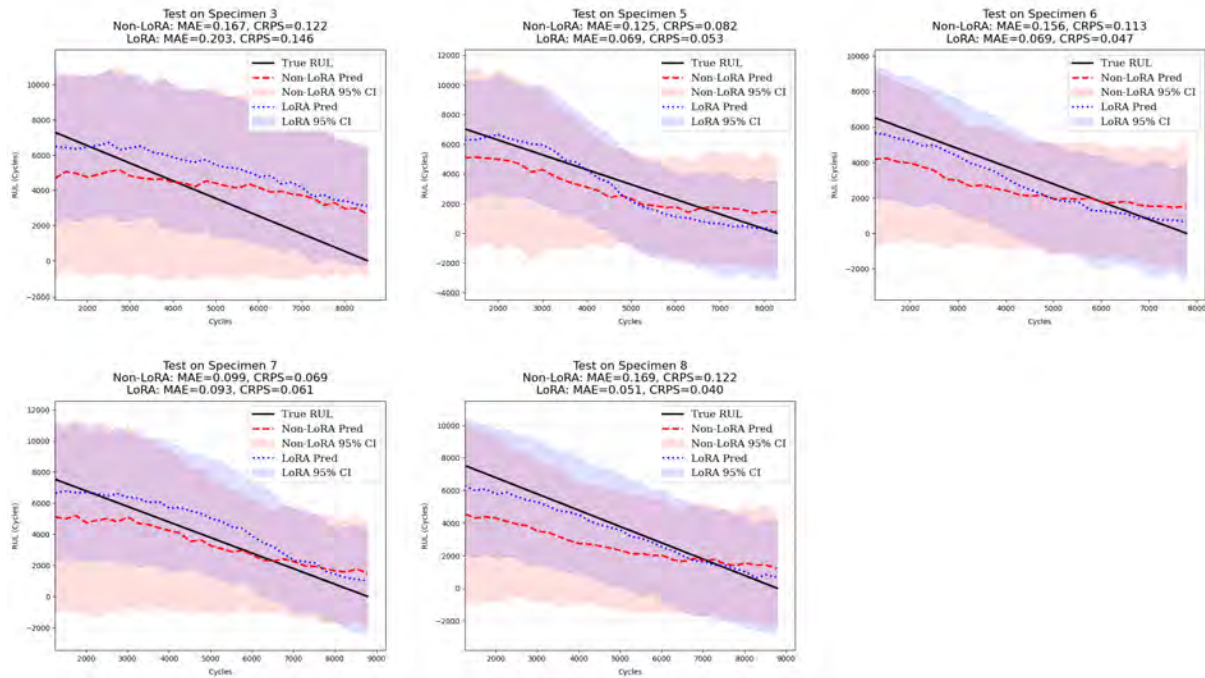


Figure 6.4: RUL predictions under the Min criterion with a single system-level training history: Comparison between the Standard Transfer Learning Baseline (red) and the LoRA framework (blue).

Table 6.5: Comparative predictive performance under the Min criterion with a single system-level training history.

Metric	Standard TL		LoRA	
	Mean	Std	Mean	Std
MAE	0.1283	0.0210	0.1078	0.0107
MAE (cycles)	1099.4	188.6	928.2	104.6
RMSE	0.1503	0.0262	0.1250	0.0138
RMSE (cycles)	1288.6	236.0	1076.6	135.0
CRPS	0.0925	0.0173	0.0778	0.0089
CRPS (cycles)	793.7	155.5	669.9	86.9
Coverage	1.0000	0.0000	0.9778	0.0052
Aleatoric	0.04948	0.00831	0.02774	0.00916
Epistemic	0.01756	0.00213	0.01326	0.00235

When the number of available system-level training histories is increased to five,

both frameworks show further improvement, as reported in Table 6.6 and illustrated in Figure 6.5. The Standard Transfer Learning Baseline reduces its MAE to 0.1118 and CRPS to 0.0779, demonstrating that even a frozen backbone can benefit from additional system-level training data when the aggregator has more examples to learn from. The LoRA framework, however, achieves substantially stronger gains, reaching a mean MAE of 0.0640 and CRPS of 0.0463, reductions of approximately 51% and 51% respectively relative to the Zero-Shot Baseline, and approximately 43% and 41% relative to the Standard Transfer Learning Baseline under the same regime. The aleatoric and epistemic uncertainties are both reduced considerably when moving from one to five training histories. The epistemic component drops from 0.01326 to 0.00507, reflecting the improved model confidence resulting from richer system-level adaptation, a direct consequence of the LoRA layers having access to a more representative sample of system-level degradation behaviour. The aleatoric component similarly decreases from 0.02774 to 0.01243, suggesting that the model has learned a more precise characterisation of the inherent fatigue variability at the system level. Notably, aleatoric uncertainty remains the dominant component in both regimes (see Subsection 2.4.3), with the epistemic component dropping from 0.01326 to 0.00507 as system-level coverage increases. Coverage remains at 1.000 for both frameworks, indicating that the predictive intervals reliably contain the true RUL throughout the specimen lifetime.

Comparison: Non-LoRA vs. LoRA Framework — Trained on 5 System Level Training Histories (Minimum Criterion)

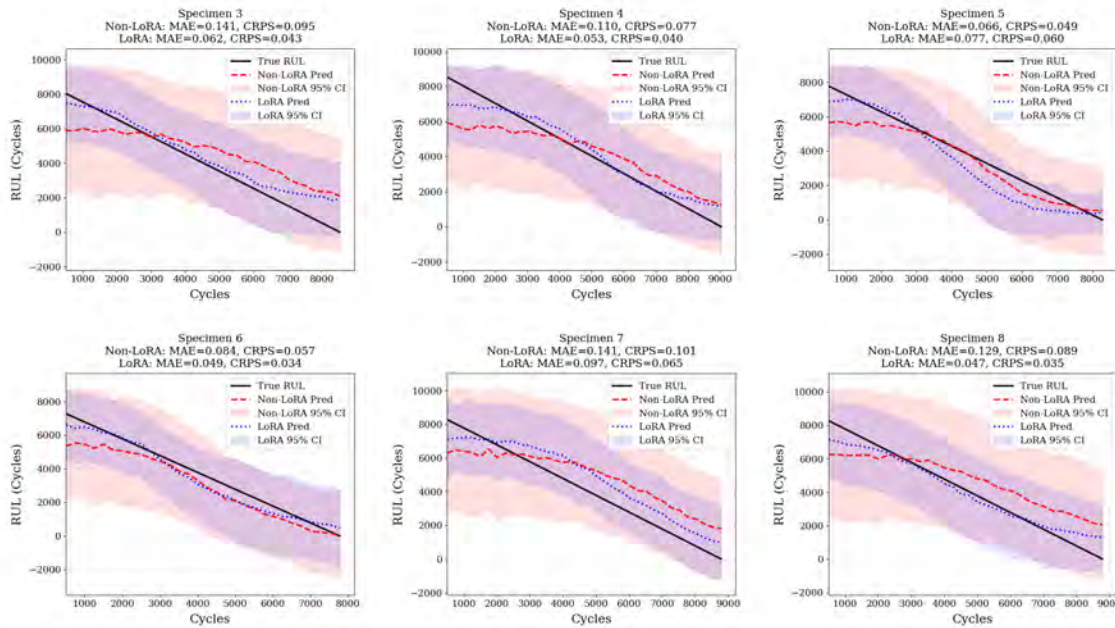


Figure 6.5: RUL predictions under the Min criterion with 5 system-level training histories: Comparison between the Standard Transfer Learning Baseline (red) and the LoRA framework (blue).

Table 6.6: Comparative predictive performance under the Min criterion with five system-level training histories.

Metric	Standard TL		LoRA	
	Mean	Std	Mean	Std
MAE	0.1118	0.0284	0.0640	0.0177
MAE (cycles)	961.7	267.9	547.3	158.3
RMSE	0.1294	0.0250	0.0782	0.0170
RMSE (cycles)	1111.6	246.8	668.8	152.5
CRPS	0.0779	0.0191	0.0463	0.0120
CRPS (cycles)	670.1	182.2	395.6	106.8
Coverage	0.9970	0.0043	1.0000	0.0000
Aleatoric	0.02710	0.00320	0.01243	0.00285
Epistemic	0.01140	0.00250	0.00507	0.00173

Overall, under the Min criterion the LoRA framework consistently outperforms both baselines across all evaluated metrics and training regimes. These results confirm that the LoRA framework is capable of reliable system-level prognostics even under severe data scarcity, while scaling favourably as more training data becomes available.

6.2.2 Max Criterion

Under the Max Criterion, structural integrity is maintained as long as at least one component remains operational. System failure occurs only when all three individual open-hole coupons have completely failed, and the system-level true RUL is thus dictated by the maximum operating lifetime recorded within the triplet. Figure 6.6 illustrates the RUL predictions of the Zero-Shot Baseline under the Max criterion, with metrics reported in Table 6.7. Compared to the Min criterion, the Zero-Shot Baseline exhibits noticeably higher errors under the Max criterion, with a mean MAE of 0.1825 and CRPS of 0.1353. This is consistent with the greater difficulty of the Max criterion task. Predicting the RUL of the last surviving component requires the model to capture the upper tail of the component lifetime distribution, which is inherently more variable and harder to estimate from static coupon-level representations alone. More importantly, in a physical parallel configuration, the failure of an individual coupon triggers an immediate load redistribution, diverting the stress onto the remaining operational components and accelerating their structural degradation. Coverage drops to 0.8977, further indicating that despite the substantial width of the uncertainty intervals, they fail to reliably contain the true system RUL. This discrepancy stems from the baseline model’s inability to accurately approximate the underlying curvature and trajectory of the true RUL degradation path. Consequently, the prediction intervals become severely misaligned with the actual target trend, even though they remain wide; this structural bias prevents them from capturing the true RUL effectively.

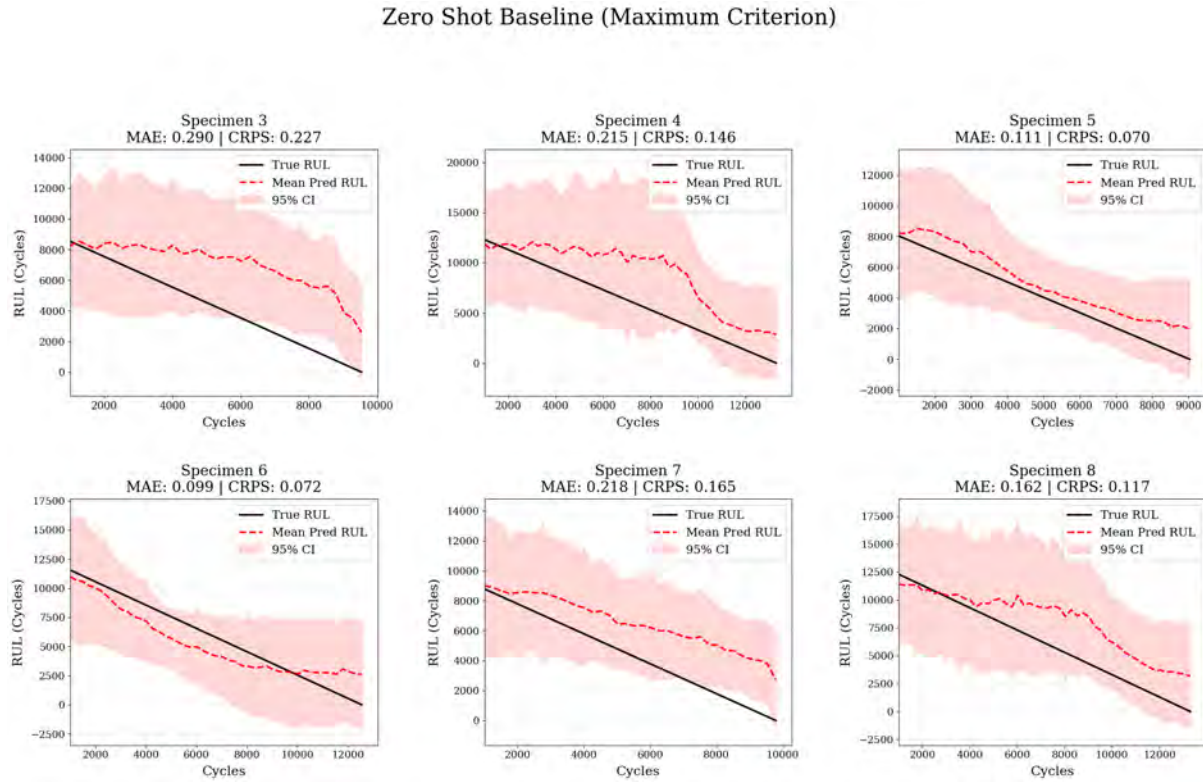


Figure 6.6: RUL predictions of the Zero-Shot Baseline under the Max criterion.

Table 6.7: Predictive performance of the Zero-Shot Baseline under the Max criterion.

Metric	Mean	Std
MAE	0.1825	0.0664
MAE (cycles)	2026.4	699.2
RMSE	0.2044	0.0741
RMSE (cycles)	2276.2	799.3
CRPS	0.1353	0.0547
CRPS (cycles)	1501.8	541.1
Coverage	0.8977	0.0031
Aleatoric	0.01846	0.00310
Epistemic	0.01992	0.00637

With a single system-level training history, both trained frameworks improve substantially over the Zero-Shot Baseline, as reported in Table 6.8 and illustrated in Figure 6.7. The Standard Transfer Learning Baseline reduces MAE to 0.1511 and CRPS to 0.1119, a reduction of approximately 17% and 16% respectively relative to the Zero-Shot Baseline. The LoRA framework further improves upon this, achieving a mean MAE of 0.1180 and CRPS of 0.0886, representing reductions of approximately 35% and 33% relative to the Zero-Shot Baseline, and approximately 22% and 21% relative to the Standard Transfer Learning Baseline. Coverage remains high for both frameworks, with the LoRA framework maintaining 0.9698. Within the LoRA framework, the aleatoric uncertainty

(0.03225) substantially exceeds the epistemic component (0.00883), consistent with the pattern observed under the Min criterion (Subsection 6.2.1), and reflecting the inherent variability of the parallel failure process.

Comparison: Non-LoRA vs. LoRA Framework — Trained on Specimen 4 (Maximum Criterion)

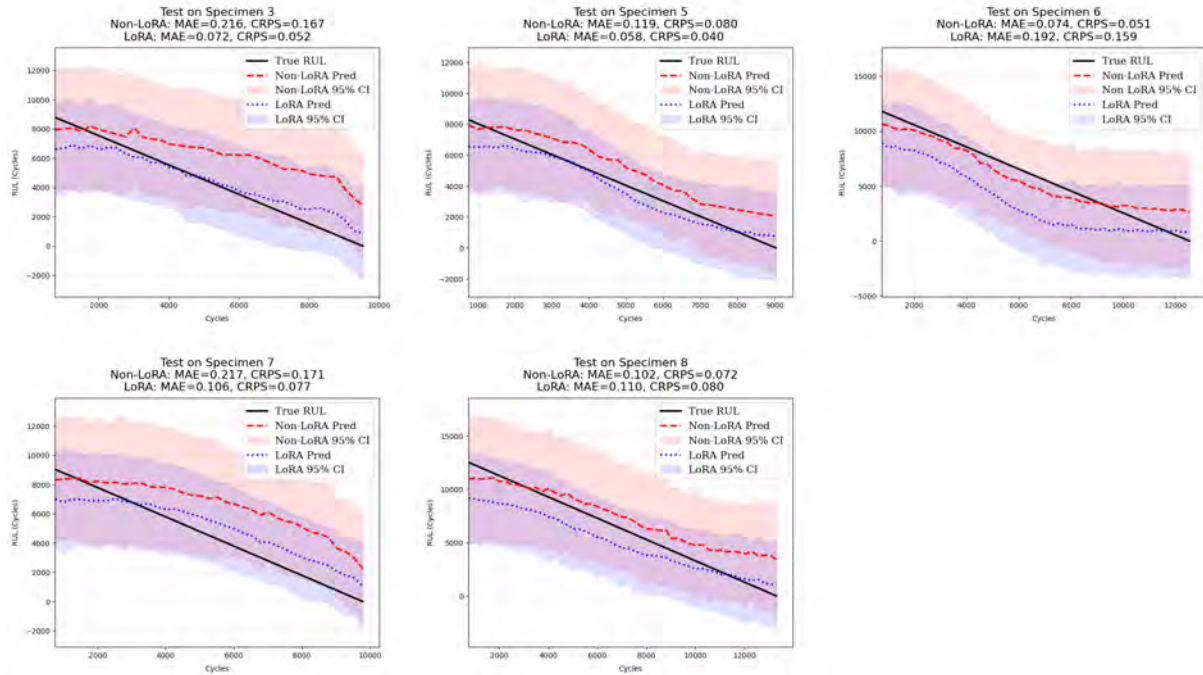


Figure 6.7: RUL predictions under the Maximum criterion with a single system-level training history: Comparison between the Standard Transfer Learning Baseline (red) and the LoRA framework (blue).

Table 6.8: Comparative predictive performance under the Max criterion with a single system-level training history.

Metric	Standard TL		LoRA	
	Mean	Std	Mean	Std
MAE	0.1511	0.0197	0.1180	0.0158
MAE (cycles)	1657.2	237.2	1334.3	182.4
RMSE	0.1732	0.0230	0.1344	0.0167
RMSE (cycles)	1899.1	278.3	1517.2	188.1
CRPS	0.1119	0.0180	0.0886	0.0119
CRPS (cycles)	1219.6	210.2	999.0	129.8
Coverage	0.9928	0.0115	0.9698	0.0417
Aleatoric	0.04163	0.01598	0.03225	0.02192
Epistemic	0.01288	0.00138	0.00883	0.00285

When five system-level training histories are available, both frameworks continue to improve, as reported in Table 6.9 and illustrated in Figure 6.8. The Standard Transfer

Learning Baseline achieves a mean MAE of 0.1250 and CRPS of 0.0910, while the LoRA framework reaches 0.1080 and 0.0854 respectively. Although the absolute gains of the LoRA framework over the Standard Transfer Learning Baseline are more modest under the Max criterion compared to the Min criterion, they remain consistent across all metrics. Notably, the Standard Transfer Learning Baseline achieves perfect coverage (1.000) under this regime, which must be interpreted with caution. The elevated aleatoric uncertainty (0.03232) relative to the LoRA framework (0.02020) suggests that the frozen backbone produces less precise distributional estimates, resulting in inflated predictive intervals that achieve nominal coverage by width rather than by accuracy. The LoRA framework achieves a coverage of 0.9163, marginally below the nominal 95% level. This is attributed to the sharper predictive intervals produced by the LoRA adaptation, as reflected in the lower aleatoric uncertainty (0.02020 vs 0.03232 for the Standard TL baseline): a more precise distributional estimate naturally yields a narrower confidence interval, which in turn reduces coverage.

Comparison: Non-LoRA vs. LoRA Framework — Trained on 5 System Level Training Histories (Maximum Criterion)

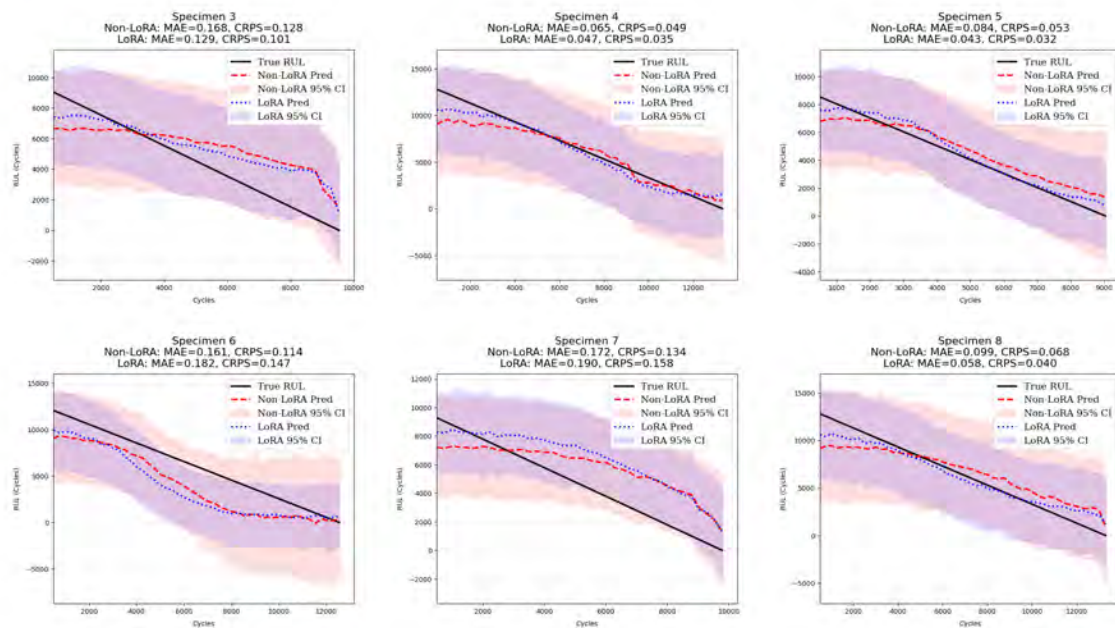


Figure 6.8: RUL predictions under the Maximum criterion with 5 system-level training histories: Comparison between the Standard Transfer Learning Baseline (red) and the LoRA framework (blue).

Table 6.9: Comparative predictive performance under the Max criterion with five system-level training histories.

Metric	Standard TL		LoRA	
	Mean	Std	Mean	Std
MAE	0.1250	0.0435	0.1080	0.0621
MAE (cycles)	1376.4	450.0	1191.6	682.1
RMSE	0.1435	0.0441	0.1268	0.0666
RMSE (cycles)	1588.6	453.1	1398.9	725.5
CRPS	0.0910	0.0356	0.0854	0.0531
CRPS (cycles)	999.7	351.9	939.1	580.1
Coverage	1.0000	0.0000	0.9163	0.1186
Aleatoric	0.03232	0.00471	0.02020	0.00333
Epistemic	0.00954	0.00228	0.00605	0.00054

Overall, under the Max criterion the LoRA framework consistently outperforms both baselines in terms of point estimate accuracy and uncertainty calibration. The more moderate absolute improvements compared to the Min criterion are consistent with the inherently greater difficulty of the parallel failure assumption, where system lifetime is governed by the last surviving component and is therefore more sensitive to inter-specimen variability. Nevertheless, the LoRA framework demonstrates robust and reliable performance across both training regimes, confirming its suitability for system-level prognostics under different structural assumptions.

The proposed LoRA framework is compared against the RUL Inference via RUL Inoperabilities Model (RIM) approach of [37], which was evaluated on the same in-house fatigue dataset under a parallel (Max) failure assumption using two base predictors: a Hidden Semi-Markov Model (HSMM) and a Particle Filter (PF). Both studies operate under a single system-level training history regime; however, an important methodological distinction exists: in [37] a fixed specimen was selected as the training history, whereas the proposed framework employs a leave-one-out cross-validation scheme in which every available specimen cyclically occupies the training role, yielding a more statistically robust and unbiased performance estimate. This distinction limits the directness of the comparison, and the following results should be interpreted accordingly. In terms of point estimate accuracy, the RIM method reports mean RMSE values of 6.82 kcycles (HSMM) and 7.32 kcycles (PF), whereas the proposed LoRA framework achieves a mean RMSE of 1.517 kcycles under the Max criterion with a single training history. A similar trend is observed in probabilistic calibration: RIM achieves mean CRPS values of 4.52 kcycles (HSMM) and 4.06 kcycles (PF), while the LoRA framework reaches 0.999 kcycles. Regarding coverage, RIM reports mean values of 0.93 and 0.65 for the two base predictors respectively, while the LoRA framework achieves 0.9698, comparable to the HSMM-based RIM while maintaining considerably sharper predictive intervals. Overall, despite the methodological differences, the LoRA framework demonstrates competitive

predictive performance across all evaluated metrics.

6.3 Effect of the Number of System-Level Training Histories

To investigate the data efficiency of the proposed LoRA framework, a systematic sensitivity analysis is conducted on the impact of the number of available system-level training histories on predictive performance. The analysis spans configurations from one to five training histories, evaluated in terms of MAE and CRPS. This investigation focuses exclusively on the LoRA framework, as it demonstrated superior performance over both baselines under both criteria in Section 6.2. A consistent monotonically decreasing trend is observed across both criteria as the number of training histories increases, confirming that the LoRA framework is capable of effectively exploiting additional system-level data to improve both point estimate accuracy and probabilistic calibration.

6.3.1 Min Criterion

Under the Min criterion, both MAE and CRPS exhibit a consistent decreasing trend as the number of system-level training histories increases from one to five, as illustrated in Figure 6.9. The most notable improvement occurs between four and five training histories, where both metrics undergo a pronounced drop, suggesting that the LoRA adaptation mechanism derives particular benefit from the additional system-level context in this regime. Specifically, MAE decreases from approximately 0.105 at a single training history to 0.064 at five histories, representing a relative reduction of approximately 39%. Similarly, CRPS decreases from 0.076 to 0.046 over the same range, a reduction of approximately 40%. These results indicate that the LoRA framework is not only capable of operating under severe data scarcity, but scales favourably as more system-level information becomes available. The steeper decline observed in the final training step further suggests that the model has not yet reached saturation, and performance may continue to improve with additional training data beyond the five histories available in this study.

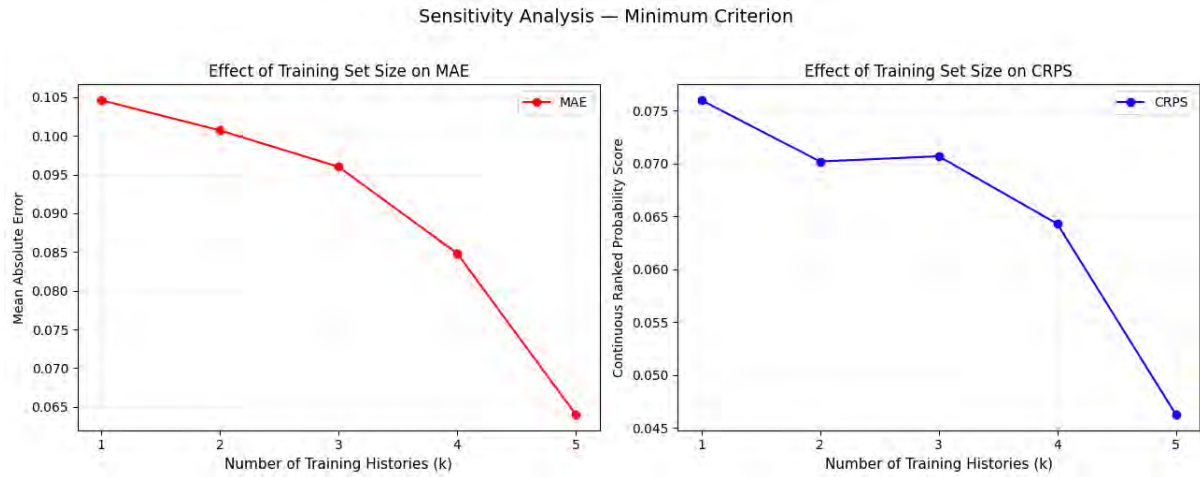


Figure 6.9: Impact of the number of system-level training histories on MAE (left) and CRPS (right).

6.3.2 Max Criterion

Under the Max criterion, the sensitivity analysis reveals a similarly decreasing trend in MAE as the number of training histories increases, as shown in Figure 6.10. MAE decreases monotonically from approximately 0.118 at a single training history to 0.108 at five histories, a relative reduction of approximately 8%. While the overall magnitude of improvement is more moderate compared to the Min criterion, the trend remains consistent, reflecting a steady accumulation of system-level knowledge by the LoRA adaptation layers.

The CRPS curve, however, exhibits a non-monotonic behaviour: it decreases from 0.089 at one history to a minimum of 0.084 at four histories, before rising slightly to 0.085 at five histories. This minor uptick at the final step may reflect variability in the leave-one-out evaluation folds rather than a genuine degradation in probabilistic calibration, given the small dataset size. Overall, the CRPS at five histories remains substantially lower than at one history, confirming a net improvement in uncertainty quantification as more system-level data is incorporated. The comparatively smaller gains under the Max criterion relative to the Min criterion suggest that the parallel failure assumption presents a more challenging learning problem, likely due to the greater variability in system lifetime dictated by the last-surviving component.

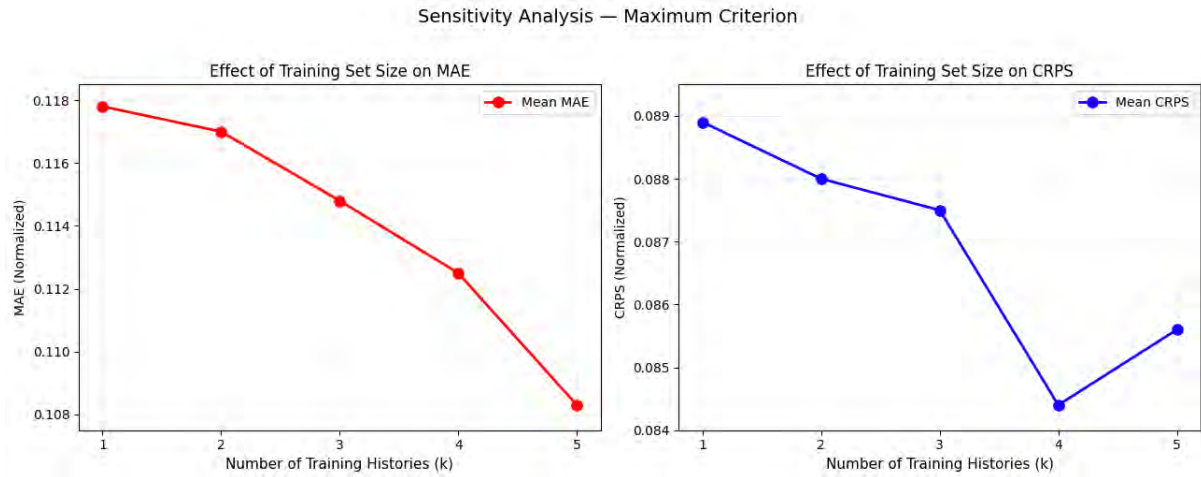


Figure 6.10: Impact of the number of system-level training histories on MAE (left) and CRPS (right).

6.3.3 Impact of the Low-Rank Constraint: LoRA vs. Full Fine-Tuning

To isolate the contribution of the low-rank adapter constraint, an ablation study was conducted under the single system-level training history regime. The proposed parameter-efficient design is compared directly against a traditional full fine-tuning strategy (*Fine-Tune*), where all weights of the pretrained model (LSTM and dense layers) are updated without any low-rank constraints. The quantitative comparisons are summarized in Table 6.10 for the Min criterion and Table 6.11 for the Max criterion.

Under the Min criterion (Table 6.10), the proposed single-adapter framework achieves a CRPS of 0.0774 (665.7 cycles) and an RMSE of 0.1251 (1074.8 cycles), substantially outperforming the full fine-tuning approach, which yields a CRPS of 0.1002 (859.3 cycles) and an RMSE of 0.1585 (1359.0 cycles). A similar performance trend is observed under the Max criterion (Table 6.11); the LoRA framework reaches a CRPS of 0.0886 (999.0 cycles) and an RMSE of 0.1344 (1517.2 cycles), whereas the fully fine-tuned model degrades to a CRPS of 0.1110 (1203.6 cycles) and an RMSE of 0.1674 (1823.9 cycles).

Furthermore, the uncertainty decomposition reveals that full fine-tuning results in higher aleatoric uncertainty estimates under both criteria (0.03565 for Min and 0.04928 for Max), indicating that the unconstrained dense layer produces less precise distributional predictions. Conversely, the proposed LoRA configuration maintains tighter, well-calibrated prediction intervals, achieving a superior coverage probability of 1.0000 under the Min criterion and 0.9698 under the Max criterion.

These results conclusively demonstrate that the low-rank constraint itself, rather than merely a reduction in the number of trainable parameters, acts as a critical implicit regularizer. Under severe system-level data scarcity, unconstrained optimization leads to over-

fitting and severe calibration degradation, while the proposed shared LoRA framework successfully balances model adaptability with robust structural knowledge preservation.

Table 6.10: Ablation study results under the Min criterion, comparing the proposed LoRA framework against unconstrained full fine-tuning using a single system-level training history.

Metric	Fine-Tune	LoRA Framework
MAE	0.1345	0.1078
MAE (cycles)	1153.1	928.2
RMSE	0.1585	0.1251
RMSE (cycles)	1359.0	1074.8
CRPS	0.1002	0.0774
CRPS (cycles)	859.3	665.7
Coverage	0.9776	0.9778
Aleatoric	0.03565	0.02774
Epistemic	0.01204	0.01326

Table 6.11: Ablation study results under the Max criterion, comparing the proposed LoRA framework against unconstrained full fine-tuning using a single system-level training history.

Metric	Fine-Tune	LoRA Framework
MAE	0.1468	0.1180
MAE (cycles)	1597.9	1334.3
RMSE	0.1674	0.1344
RMSE (cycles)	1823.9	1517.2
CRPS	0.1110	0.0886
CRPS (cycles)	1203.6	999.0
Coverage	0.9665	0.9698
Aleatoric	0.04928	0.03225
Epistemic	0.01389	0.00883

Chapter 7

Limitations and Future Work

The proposed LoRA-based transfer learning framework demonstrates promising performance for system-level RUL prediction under data-scarce conditions. Nevertheless, several limitations of the current study should be acknowledged, and directions for future work are identified accordingly.

A natural extension of the present framework is the adoption of a Bayesian low-rank adaptation strategy [38], which would propagate parameter uncertainty through the adaptation matrices themselves, providing a more principled treatment of model uncertainty during system-level fine-tuning.

The present framework operates under an offline training paradigm, in which the LoRA adaptation layers are trained on a fixed set of system-level histories prior to deployment. In practice, condition monitoring data arrive sequentially as a continuous stream, and the degradation state of the system evolves over time. The current architecture does not exploit this temporal structure during inference, and is therefore unable to update its parameters as new observations become available. A natural extension of this work is the development of an online adaptation mechanism, in which the model continuously refines its predictions as the data stream progresses. This aligns with the paradigm of Test-Time Training (TTT) [39, 40], where the model is allowed to update a subset of its parameters at inference time using the incoming test data, without requiring access to the original training set. Given the low-rank structure of the LoRA layers, such online updates would involve only a small number of parameters, making this extension computationally feasible even under real-time operational constraints.

The comparison between the LoRA-based framework and the Standard Transfer Learning baseline is subject to a parameter imbalance: the inclusion of the LoRA adaptation matrices introduces additional trainable parameters not present in the baseline, meaning that part of the observed performance gap could be attributable to increased model capacity rather than exclusively to the inductive bias of low-rank adaptation. A controlled ablation study matching parameter counts across both configurations would provide a more rigorous attribution of the performance gains. Furthermore, the current

study considers a simplified system configuration comprising identical and mechanically equivalent components; extending the framework to heterogeneous component configurations would require modifications to the aggregator architecture and the failure criterion embedding, and represents an important step toward industrial applicability. Both of these directions are left for future work.

A further limitation of the current framework concerns the choice of output distribution. Since a Gaussian negative log-likelihood loss is employed, the predictive distribution is defined over the entire real line, and the lower bound of the predictive interval may extend into negative values as the true RUL approaches zero near end-of-life. This is physically inadmissible, as remaining useful life is strictly non-negative by definition. Addressing this constraint formally, for instance through the adoption of a strictly positive output distribution such as a log-normal or a Weibull formulation, represents a natural direction for future work.

Chapter 8

Conclusions

This work introduced a novel LoRA-based transfer learning framework designed for system-level RUL prediction under limited data availability. By leveraging parameter-efficient fine-tuning, the proposed approach effectively bridges the gap between coupon-level degradation modeling and full system-level prognostics, requiring minimal target-system data for adaptation. The framework addresses a critical obstacle in PHM by offering a scalable, computationally efficient alternative to traditional full-scale models, which are often non-viable in industrial settings due to high data demands.

The methodology was thoroughly validated through an extensive benchmarking campaign utilizing experimental fatigue data from aluminum 7075-T6 multi-coupon specimens, representing a simplified system-level configuration with mechanical coupling. When evaluated against standard baselines, specifically a non-parametric Zero-Shot approach and a Standard Transfer Learning model, the proposed LoRA-based framework demonstrated superior performance across both low-data and moderate-data training regimes. It consistently enhanced SRUL prediction accuracy and generated more reliable probabilistic profiles, even when adapted using only a single system-level history. The low-rank adaptation layers proved highly capable of capturing complex load-sharing and degradation dynamics while maintaining a remarkably low parameter footprint, effectively mitigating the risks of overfitting that typically compromise standard transfer learning approaches under data scarcity.

These findings underscore the practical utility of parameter-efficient transfer learning for safety-critical sectors like aerospace, where full-scale experimental testing is economically and logistically constrained. By providing high-fidelity predictions with minimal training overhead, the framework establishes a robust, interpretable, and data-efficient pathway toward practical, uncertainty-aware, and system-level prognostics in modern engineering applications.

Bibliography

- [1] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel. Evaluating prognostics performance for algorithms incorporating uncertainty estimates. In *IEEE Aerospace Conference*, pages 1–11. IEEE, 2010.
- [2] M. Salinas-Camus, K. Goebel, and N. Eleftheroglou. A comprehensive review and evaluation framework for data-driven prognostics: Uncertainty, robustness, interpretability, and feasibility. *Mechanical Systems and Signal Processing*, 237:113015, 2025.
- [3] S. Kim, J.-H. Choi, and N. H. Kim. Challenges and opportunities of system-level prognostics. *Sensors*, 21(22):7655, 2021.
- [4] F. Tamssaouet, K. T. P. Nguyen, K. Medjaher, and M. E. Orchard. System-level failure prognostics: Literature review and main challenges. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 237(3), 2022.
- [5] M.-H. Laves et al. Ensemble neural networks for remaining useful life (RUL) prediction. *arXiv preprint arXiv:2309.12445*, 2023.
- [6] F. Pourkamali-Anaraki, J. F. Hussein, and S. E. Stapleton. Probabilistic neural networks (PNNs) for modeling aleatoric uncertainty in scientific machine learning. *arXiv preprint arXiv:2402.13945*, 2024.
- [7] T. Loutas, A. Oikonomou, N. Eleftheroglou, F. Freeman, and D. Zarouchas. Remaining useful life prognosis of aircraft brakes. *International Journal of Prognostics and Health Management*, 13(1), 2022.
- [8] R. Sunder. Full-scale fatigue testing. In N. Eswara Prasad and R. J. H. Wanhill, editors, *Aerospace Materials and Material Technologies*. Springer, 2016.
- [9] G. Galanopoulos, E. Fytsilis, N. Yue, A. Broer, D. Milanoski, D. Zarouchas, and T. Loutas. A data-driven methodology for upscaling remaining useful life predictions: From single- to multi-stiffened composite panels. *Composites Part C: Open Access*, 11:100366, 2023.

- [10] H. Li, W. Zhao, Y. Lu, and J. Liao. Multi-sensor information fusion based on graph convolutional network for system-level remaining useful life prediction. *Reliability Engineering & System Safety*, 223:108496, 2022.
- [11] G. Stamatelatos, G. Galanopoulos, D. Zarouchas, and T. Loutas. Graph neural networks for SHM: exploiting spatial interdependencies of strain data for diagnostics and prognostics. *Structural Health Monitoring*, 0(0), 2025.
- [12] J. Chen, H. Jing, Y. Chang, and Q. Liu. Long short-term memory neural network with transfer learning and ensemble learning for remaining useful life prediction. *Sensors*, 22(15):5744, 2022.
- [13] S. Schwendemann and A. Sikora. Transfer-learning-based estimation of the remaining useful life of heterogeneous bearing types using low-frequency accelerometers. *Journal of Imaging*, 9(2):34, 2023.
- [14] E. J. Hu et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR 2022)*, 2022.
- [15] Y. Li and W. Zhu. TRACE: Time series parameter efficient fine-tuning. *arXiv preprint arXiv:2503.16991*, 2025.
- [16] J. Coble and J. W. Hines. Identifying optimal prognostic parameters from data: A genetic algorithms approach. In *Annual Conference of the PHM Society*, volume 1, 2009.
- [17] G. Galanopoulos, N. Eleftheroglou, T. Loutas, and D. Zarouchas. A novel strain-based health indicator for the remaining useful life estimation of degrading composite structures. *Composite Structures*, 306:116579, 2023.
- [18] G. Galanopoulos, N. Eleftheroglou, T. Loutas, and D. Zarouchas. Health monitoring of aerospace structures utilizing novel health indicators extracted from complex strain and acoustic emission data. *Sensors*, 21(17):5701, 2021.
- [19] A. Kendall and Y. Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- [20] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, volume 48, pages 1050–1059, 2016.
- [21] D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN)*, pages 55–60, 1994.

- [22] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. Rusu, K. Milan, J. Quan, T. Ramalho, and A. Grabska-Barwinska. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521–3526, 2017.
- [23] Juan Terven, Diana M. Cordova-Esparza, Alfonso Ramirez-Pedraza, Edgar A. Chavez-Urbiola, and Julio A. Romero-Gonzalez. Loss functions and metrics in deep learning. *Artificial Intelligence Review*, 2025.
- [24] J. Ilemobayo et al. Hyperparameter optimisation in machine learning: A review of methods and applications. *Artificial Intelligence Review*, 2024.
- [25] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.
- [26] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.
- [27] T. Bublil, R. Cohen, R. S. Kenett, and J. Bortman. Machine health indicators and digital twins. *Sensors*, 25:2246, 2025.
- [28] N. Eleftheroglou, S. S. Mansouri, T. Loutas, P. Karvelis, G. Georgoulas, G. Nikolakopoulos, and D. Zarouchas. Intelligent data-driven prognostic methodologies for the real-time remaining useful life until the end-of-discharge estimation of the lithium-polymer batteries of unmanned aerial vehicles with uncertainty quantification. *Applied Energy*, 254:113677, 2019.
- [29] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
- [30] A. Lisnianski and G. Levitin. *Multi-State System Reliability: Assessment, Optimization and Applications*. World Scientific, 2003.
- [31] K. Goebel, B. Saha, and A. Saxena. A comparison of data-driven and model-based prediction techniques for remaining useful life estimation. In *Annual Conference of the PHM Society*, volume 1, pages 1–8, 2009.
- [32] W. Wang, M. Zhao, J. Wang, and J. Xu. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, 69:401–412, 2019.
- [33] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109:43–76, 2021.

- [34] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, 2014.
- [35] Z. Chen, M. Gryllias, and W. Li. Mechanical fault diagnosis using convolutional neural networks and extreme learning machine. *Mechanical Systems and Signal Processing*, 133:106272, 2019.
- [36] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- [37] M. Salinas-Camus, O. Carpentier, and N. Eleftheroglou. Probabilistic system-level prognostics in aerospace applications. Technical report, Intelligent System Prognostics Group, Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, 2024.
- [38] A. X. Yang, M. Robeyns, X. Wang, and L. Aitchison. Bayesian low-rank adaptation for large language models. *arXiv preprint arXiv:2308.13111*, 2024.
- [39] Y. Sun, X. Li, K. Dalal, J. Xu, A. Vikram, G. Zhang, Y. Dubois, X. Chen, X. Wang, S. Koyejo, T. Hashimoto, and C. Guestrin. Learning to (Learn at Test Time): RNNs with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.
- [40] T. Zhang, S. Bi, Y. Hong, K. Zhang, F. Luan, S. Yang, K. Sunkavalli, W. T. Freeman, and H. Tan. Test-time training done right. *arXiv preprint arXiv:2407.04599*, 2024.