SYSTEM INFORMATICS AND DATA ANALYTICS FOR SMART AND CONNECTED SYSTEMS - GOING BEYOND ACCURACY -

By

Minhee Kim

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

Doctor of Philosophy

(Industrial and Systems Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2022

Date of final oral examination: 04/26/2022

The dissertation is approved by the following members of the Final Oral Committee:

Dr. Kaibo Liu (Department of Industrial and Systems Engineering, *University of Wisconsin-Madison*)

Dr. Todd Allen (Department of Nuclear Engineering and Radiological Sciences, *University of Michigan*)

Dr. Jeffrey Linderoth (Department of Industrial and Systems Engineering, *University of Wisconsin-Madison*)

Dr. Po-ling Loh (Department of Pure Mathematics and Mathematical Statistics, *University of Cambridge*)

Dr. Wenxiao Pan (Department of Mechanical Engineering, *University of Wisconsin-Madison*)

Acknowledgement

I would like to express my deepest gratitude to my academic advisor, Dr. Kaibo Liu, for his constant encouragement, insight and tremendous opportunities he has provided throughout my doctoral studies at the University of Wisconsin-Madison. He has motivated me to become an independent researcher and thinker.

I would like to express my special thanks to the committee members, Dr. Todd Allen, Dr. Jeffrey Linderoth, Dr. Po-ling Loh, and Dr. Wenxiao Pan for their time, advice, support and contribution to this work.

I was fortunate to study with people who are more than just labmates. They are also great friends. I can't thank each of them enough for helping me get through this period of my life in the most positive way.

Lastly, my eternal thanks also go to my family for their unconditional love, support, and encouragement. In particular, I appreciate my parents for always believing in me even when I did not and always being there for me. Without their support and love, I could not have come this far in my life.

Table of Contents

List of Tables

List of Figures

Abstract

Sensor signal information is essential to accurately monitor the degradation status of a unit and predict its failure time. Recent advances in sensor technologies bring unprecedented opportunities for new developments in degradation modeling and prognostics of smart and connected systems. However, two major challenges need to be addressed to explore novel applications in degradation modeling and prognostics. The first challenge is to handle the fundamental complexity of systems, such as multiple sensors, multiple failure modes, multiple operational conditions, or heterogeneity among units. The second challenge is to provide insights and guidance for further data-driven decision-making. Many state-of-the-art approaches have been focusing solely on improving modeling and prognostic accuracy. Yet, a lack of practical interpretability limits their wide usage in real-world applications. For instance, identifying informative sensors and covariates or quantifying the uncertainty of the model prediction results are as important as predicting unit failure time for system redesign, subsequent risk analysis, and maintenance decision making.

This thesis focuses on modeling the degradation processes and conducting failure time prediction of complex systems by bridging the gaps between advanced statistics, machine learning, and engineering domain knowledge. The proposed methodologies enable (i) the multisensor signal fusion to infer the unobserved degradation status, (ii) accurate and reliable prognostics with interval estimations of failure time of units involving multiple failure modes and multiple operational conditions, (iii) efficient handling of heterogeneous units where each unit has its distinct individual-level characteristics, (iv) selecting informative degradation sensors, and (v) incorporation of desired general degradation characteristics from engineering domain knowledge.

The first chapter introduces the background and elaborates on the challenges in degradation modeling and prognostics for complex systems. The objective of this thesis is also highlighted.

Chapter 2 focuses on cases where multiple sensors monitor a single unit under a single failure mode and a single operational condition. This chapter proposes a novel data fusion method that constructs a 1-D health index via automatically selecting and combining multiple sensor signals to better characterize the degradation process. In particular, this methodology develops a new latent linear model that constructs the health index and selects informative sensors in a unified manner. Chapter 3 handles the degradation processes of more complex systems that may involve multiple sensor signals, multiple failure modes, and multiple operational conditions. It may not be feasible to explicitly formulate the mathematical relationships between the collected degradation signals and the underlying degradation processes or the remaining useful lifetimes. To address this issue, we propose a novel Bayesian deep learning framework that incorporates general characteristics of degradation processes and provides interval estimations of remaining useful life. Chapter 4 focuses on individualized degradation modeling and prognostics for a heterogeneous group, where each individual unit shows a distinct degradation process. Existing degradation modeling methodologies usually treat each unit separately and do not fully utilize the distinct characteristics of each individual. In this chapter, we propose a generic framework to handle the heterogeneity across units by effectively leveraging the intrinsic covariate information, which is closely related to the unit's degradation process. The degradation processes of units can be viewed as functional data. In Chapter 5, we will introduce the covariate-dependent sparse functional data analysis. Unlike most of the existing methods, the proposed method can handle high-dimensional covariates with the informative covariate identification procedure, and sparse and irregularly spaced measurements, i.e., does not require complete or dense observations. The main innovation of the proposed method is that we model the variation coming from covariates and the variation left conditioned on covariates, such that the functional principal component analysis and Gaussian

process can be conducted in a unified manner. In Chapter 6, we summarize the main contributions of the thesis.

In summary, this thesis contributes to developing stochastic degradation modeling, diagnostic and prognostic analysis methodologies explicitly designed for smart and connected systems. The research possesses great potential for applications in manufacturing, health care, and energy facilities, which will contribute to optimizing economic performance and minimizing safety risks.

Chapter 1 Introduction

1.1 Motivation and Overview

In many modern high-reliability applications, the performance of a system degrades gradually over time and eventually leads to failure. The failure time of a system is often defined as the time when the degradation status reaches a predefined threshold level. Thus, by modeling the progression of the degradation status and estimating the time it first hits the failure threshold, we will be able to predict the remaining useful lifetime (RUL) of the system. One fundamental challenge here is that the underlying degradation status of systems is unobservable. To overcome this, we monitor one or multiple sensor signals which are directly or indirectly related to the degradation status of a system to infer the underlying degradation status. These signals are also called degradation signals or condition monitoring (CM) signals. For example, as a bearing degrades, it exhibits larger vibration. By modeling and estimating the amplitude of a bearing's vibration, we can estimate the failure time.

Existing literature on RUL prediction often assumes that a single degradation signal can fully characterize the degradation process, and all systems have the same characteristics (homogeneity) and experience a single failure mode under a single operational condition. Under these assumptions, several popular methods have been developed, such as the general path models [1], [2], stochastic process models [3], and machine learning approaches [4]. We will present a state-of-the-art review of the existing literature on degradation modeling and prognostics in Section 1.3.

In recent years, the rapid advancements of the Internet of Things (IoT)-based sensing technology, communication networks, and computing power have emerged as powerful tools to enhance the modeling, inference, and prognostics of degradation processes. We are now able to

collect information about the system from various sources beyond a single degradation signal and efficiently transmit or store the data. These trends present major opportunities and challenges to develop novel statistical methods for degradation modeling and prognostics of modern complex systems.

First, recent developments in sensor technologies have dramatically accelerated the use of multiple sensors to monitor the degradation process of a system. In this way, various aspects of the degradation process can be captured [5]. However, different sensor signals usually have different levels of relevance to the degradation process. In practice, some sensors may be unrelated to the underlying degradation process. Thus, there are two challenges involved in the multisensor degradation modeling: (i) how to identify informative sensors, and (ii) how to properly combine the information from the selected sensor signals to accurately estimate RULs.

Second, the degradation processes are stochastic in nature. There are multiple sources of uncertainties in degradation modeling and prognostics including the system-to-system variability or the sensor measurement errors. Thus, it impossible to provide point estimations of RULs with absolute certainty in many applications. It is crucial to obtain interval estimations of RULs or estimations of RUL distributions for subsequent risk analysis and maintenance decision making [6].

Third, in many applications, whereas all systems in a group share some similarities, each system has its own distinct individual-level characteristics, such as the manufacturing design information of engineering systems and the risk factors of patients. These individual-level characteristics can provide valuable information about the degradation process in addition to the degradation signals. However, many traditional approaches in the literature have assumed homogeneous systems and constructed a group-level degradation model that all systems share [7]. Proper modeling of the

individual-level characteristics will improve the overall RUL prediction accuracy and enable us to address long-standing challenges in the existing literature such as the cold start case where the system of interest is newly launched and has not yet been collected any degradation signals.

1.2 Objectives

The objectives of this research are:

(i) developing a novel health index-based method which selects informative sensors and combines multiple sensor signals to better understand the degradation process.

(ii) proposing a novel Bayesian deep learning framework that performs degradation modeling and prognostics in complex systems involving multiple sensors, multiple failure modes and multiple operational conditions with probabilistic interpretability.

(iii) developing individualized degradation modeling and prognostics of a heterogeneous group which encodes the available information about intrinsic covariates into the random-effect coefficients and quantifies the similarities between different units.

(iv) identifying important intrinsic covariates when each system has a small number of longitudinal measurements and there is no physical knowledge available for a functional form of the degradation process.

1.3 State-of-the-art

Existing degradation modeling and prognostic approaches can be broadly classified into two main categories: model-based approaches and data-driven approaches. In model-based methods, a mathematical model is developed to represent the underlying physics of the system degradation. For instance, different variants of the Paris-Erdogan law have been used to model the crack growth in a gear [8], [9]. Data-driven methods build models based on the historical data, such as

degradation signals or covariates, with less required knowledge of the inherent system degradation physics.

Popular data-driven methods include general path models [1], [2], stochastic process models [3], [10]–[14], and machine learning approaches [15]–[17]. General path models, first introduced in [1], formulate the sensor signal using a random-effect model, where the random-effect parameters are used to capture the unit-to-unit variability. Stochastic process models use different stochastic processes to represent the evolution of a sensor signal, e.g., Wiener process [3], [10], [11], inverse Gaussian process [12], and gamma process [13], [14]. Machine learning approaches include support vector machine (SVM) [15], k-nearest neighbors [16], decision trees [17], and neural networks [18], [19].

Most of the existing general path models and stochastic process models assume that a single sensor can fully characterize the underlying degradation status, and thus model the signal as a mixed effect model or a stochastic process. However, in practice, multiple sensors are often installed on a single system to capture the various aspects of complex degradation mechanisms. To address this issue, several recent approaches have been proposed in multisensor degradation modeling using data fusion. For instance, there are principal component analysis (PCA)-based methods to select the informative sensors and extract features from multiple sensor signals for prognostics. Unfortunately, extracted features are quite difficult to interpret in practice. Alternatively, health index-based methods provide better interpretability and practicality. The health index-based method constructs a one-dimensional index by directly combining multiple sensor signals to represent the underlying degradation process. By visualizing one-dimensional health index, it is much more straightforward to monitor the current degradation process and predict the RUL. Recent health index-based approaches [20]–[22] define the desired properties of

a good degradation signal, e.g., monotonic trends, and construct the health index by optimizing those properties. However, these methods are heuristic and do not guarantee finding the optimal combination of sensor signals. Moreover, a system may consist of multiple and mutually interactive subsystems or components, involve multiple failure mechanisms [23], [24] or operate under multiple operational conditions to meet certain task requirements [25]. In such cases, there may not exist a one-dimensional index that can well represent the underlying degradation process.

Machine learning approaches tackle such difficulties by taking the most recent multiple sensor signals as inputs and RULs as outputs. In other words, the approaches let the model find the relations between multiple sensor signals and corresponding RULs based on the historical data. The main focus of the existing machine learning prognostic approaches is on how to design the model, e.g., what type of neural network to use or how to concatenate operational variables and multiple sensor signals as one input. For instance, [26] used a recurrent neural network to estimate the degradation percentage of bearings based on the features extracted from the vibration signals. [18] recently proposed a neural network for RUL prediction, which is also based on one of the variants of the recurrent neural network. While these methods enjoy great flexibility, one major drawback is that the model works in a black-box manner which means we cannot interpret why the model outputs the certain RUL prediction results or how much uncertainty is involved in the results. Moreover, most of the existing machine learning approaches do not consider the general degradation characteristics, such as the stochastic nature of degradation processes.

The incorporation of covariate information is another crucial factor for modern degradation modeling and prognostics. It is important to note how degradation signals, extrinsic covariates and intrinsic covariates are different. Degradation signals are taken over time reflecting the underlying degradation status. Extrinsic covariates are external factors that represent environmental stresses,

such as temperature or pressure. It is common to assume that the extrinsic covariates have additive effects on the degradation status, i.e., it accelerates or decelerates the degradation process. Recently, in [27], the cumulative exposure model describes the effect of extrinsic covariates on the parametric failure time distribution assuming the dynamic environmental covariates such as temperature and humidity can increase or decrease the degradation rate. Intrinsic covariates represent the basic nature of a system and often do not change over time. Examples include the genetic information of patients or manufacturing information of engineering systems. Unlike extrinsic covariates, the effects of intrinsic covariates on degradation processes cannot be modeled additively. Another widely used approach that incorporates covariates for degradation modeling is through the Cox proportional hazards model (PHM) [28]. Yet, existing PHM-based prognostic methods often directly use observed degradation signals or their features as covariates [29], [30]. For instance, in [30], the features extracted from vibration signals of bearings are used as covariates. Thus, the concept of covariates used in PHM-based approaches are closer to the degradation signals we define in this thesis. Although some efforts have focused on degradation modeling and prognostics of a heterogeneous group [31], little research has targeted the incorporation of intrinsic covariate information and covariate selection procedure.

1.4 Outline of the Thesis

The remainder of the thesis is organized as follows. Chapter 2 proposes a generic health index approach which performs degradation modeling, prognostics, and sensor selection in a unified manner for applications where multiple sensors monitor a single unit under a single failure mode and a single operational condition. A novel latent linear model is constructed to accurately characterize the unobservable underlying degradation status of the unit. By applying a variable

selection algorithm to this linear model, we identify informative sensors with very high computational efficiency.

Chapter 3 handles more complex systems that have not only multiple sensor signals, but also multiple failure modes or multiple operational conditions. To address this issue, we propose a novel Bayesian deep learning framework which provides interval estimations of RULs. In particular, this method systematically quantifies two types of uncertainties embedded in degradation modeling and prognostics: the uncertainties resulting from the unknown model parameters and those stemming from the stochastic nature of degradation processes.

Chapter 4 focuses on individualized degradation modeling and prognostics for a heterogeneous group. In this chapter, we handle the heterogeneity across units by effectively leveraging the intrinsic covariate information. A multivariate Gaussian process (GP) nonparametrically establishes the relation between the covariate information and degradation processes. Through modeling the similarities between different systems based on their covariates, efficient information transfer among systems is enabled. In other words, new collected degradation signals from one system can be shared with the entire heterogeneous group according to their covariate similarities for better degradation modeling and prognostics.

Chapter 5 discusses the covariate-dependent sparse functional data analysis. A degradation signal can be viewed as a special kind of functional data. The future work targets applications where each unit has multiple intrinsic covariates similar to the method described in Chapter 4, yet each unit now has only sparse and irregular signal measurements. The motivational application of this study is void swelling. Void swelling is a nuclear-specific material degradation mechanism that causes an increase in volume of components exposed to high-energy neutrons at high temperatures [32]. Void swelling is affected by many intrinsic covariates such as alloy composition

and material structure. It is important to identify crucial covariates affecting the swelling processes and model the effects of these covariates to mitigate the effect of swelling and ensure safe operation. Existing studies often do not use covariate information or require dense measurements. In practice, however, there are only sparse measurements available for void swelling due to the high cost of data acquisition. To address this issue, we plan to use the functional PCA to capture the similarities across different systems and the GP to model the covariate effects. We will also investigate different kernel designs of the GP to identify significant covariates. Finally, Chapter 6 summarizes the contributions of the thesis.

Chapter 2 A Generic Health Index Approach for Multisensor Degradation Modeling and Sensor Selection

2.1 Introduction

Degradation is quite common in engineering systems and will eventually lead to failures. Unexpected failures can cause production downtime, poor customer satisfaction, safety issues, etc. To avoid such losses, sensors have been widely used to monitor the degradation process of a unit. The collected sensor signals contain useful information about the degradation status of the unit, which if properly used, can lead to accurate prediction of the remaining useful life (RUL).

Most of the existing literature focuses on analyzing a single sensor signal [33], and there are two commonly used approaches, including general path models [1], [2] and stochastic process models [3], [10]–[14]. General path models formulate the sensor signal using a random-effect model, where the random-effect parameters are used to capture the unit-to-unit variability. On the other hand, stochastic process models characterize the evolution of a sensor signal as a stochastic process, e.g., Wiener process [3], [10], [11], inverse Gaussian process [12], and gamma process [13], [14], to account for the temporal variation of sensor signals.

Unfortunately, these approaches are only effective under the assumptions that the physical degradation mechanism of a monitored unit is well understood, and thus a single sensor is sufficient to fully characterize the underlying degradation process. However, in reality, it is common that a single sensor only contains partial information on the degradation process.

In order to overcome this issue, much attention has been recently focused on using multiple sensors to monitor a single unit simultaneously. In this way, different aspects of the degradation process can be captured [5]. Therefore, there is a growing need to develop efficient multisensor degradation modeling approaches. However, different sensor signals usually have different levels of relevance to the degradation process. In many real-world applications, it is even possible that some sensors are unrelated to the underlying degradation process, which compromises the accuracy of RUL prediction by acting as noise. In addition, a collection of these non-informative sensor signals may incur unnecessary costs. As a result, there are two key challenging questions involved in the multisensor degradation modeling: (i) how to screen out non-informative sensors, and (ii) how to properly combine the information from the selected sensor signals to accurately estimate the underlying degradation status of a unit.

To address these challenges, this study presents a novel health index (HI)-based data fusion model for multisensor degradation modeling and sensor selection. In particular, we combine the observable data, i.e., the failure time and the multiple sensor signals, via a novel latent linear model to accurately characterize the unobservable underlying degradation status of the unit. Consequently, the contributions of this work are summarized as follows. First, unlike the previous HI-based methods which were heuristic in nature, the proposed method ensures to discover the optimal combination of sensor signals to better understand the underlying degradation mechanism. In fact, by solving the latent linear model, our method is able to derive the best linear unbiased estimator (BLUE) of the fusion coefficients. To the best of our knowledge, this is the first work in the context of multisensor degradation modeling that has this nice property. Second, the proposed method significantly reduces the computational time. This is due to the analytical solution of the fusion coefficients that we obtain from the latent linear model. Third, the proposed method is more

generic since it does not require restrictive assumptions, such as the specific form of the degradation process, which were imposed in the previous HI-based methods. Thus, it can be widely applied to a variety of situations. Fourth, the proposed method does not require to know the exact value of the failure threshold to predict the RUL, which is usually unknown in practice. As a comparison, most of the existing studies need to assume that the failure threshold is known as a priori. Last but not least, variable selection methods for linear regression models such as adaptive lasso can be directly incorporated in our proposed method to achieve a systematic sensor selection. This would lead to more accurate prediction results and reduce unnecessary costs.

The rest of this study is organized as follows. Section 2.2 provides a literature review of the data fusion methods for prognostics. Section 2.3 describes the details of the proposed data fusion methods to construct a composite HI of a degraded unit and to predict its RUL. Section 2.4 conducts a simulation study to illustrate the effectiveness and evaluate the sensitivity of the proposed method. Section 2.5 further tests the proposed method using the degradation dataset of aircraft gas turbine engines and compares the results with the existing benchmark method. Section 2.6 provides a conclusion and a discussion of future research directions.

2.2 Related Works

In the literature, several efforts have been made to tackle multisensor degradation modeling using data fusion. In general, data fusion methods can be classified into two main categories based on the implementation level of the fusion operation: decision-level fusion and data-level fusion [5], [34].

Decision-level fusion integrates multiple results derived from different diagnostic/prognostic approaches. For example, Hu *et al*. [35] combined the RUL prediction results from different member algorithms by weighted average, where k-fold cross validation was used to determine the

weights. One of the main drawbacks of decision-level fusion approaches is that they are post-processing techniques, and thus the performance highly relies on the quality of the raw data and the data pre-processing procedure. In addition, most of these methods only produce a point estimate of the RUL.

In contrast, data-level fusion methods directly combine the sensor signals or the extracted features. In the literature, a number of data-level fusion methods have been proposed including machine learning approaches [26], [36], [37], state-space models [38], [39], PCA [40], and HI-based approaches [20]–[22], [41]–[43]. In particular, machine learning approaches such as artificial neural network directly take the most recent sensor signals or features as the inputs and provide the predicted RUL as the outputs. However, sensor signals are time-series data and conventional machine learning approaches fail to effectively capture the autocorrelation of sensor signals in the context of the degradation process. To overcome this drawback, Guo *et al.* [26] recently applied a recurrent neural network (RNN) to fuse multiple features of bearings. Though RNNs are known to be useful for handling time-series data, the constructed RNNs behave like a black box which makes it less explainable and hard to incorporate domain knowledge into the models. In addition, RNNs need to be trained by very large amounts of historical data, which is costly and often inapplicable in degradation systems. Another commonly used approach is to utilize the state-space models and discretize the degradation status into a finite state space. For example, in Yu [39], the state-space model was used to model the degradation of lithium-ion battery and to predict RUL. However, this approach relies on the memoryless assumption, i.e., the future degradation depends only on the current degradation status of a unit rather than the past, which does not always hold in real-world applications [33], [44]. The PCA has also been used for data-level fusion. Recently, in Fang *et al*. [40], functional PCA (FPCA) was used to select the

informative sensors, and multivariate functional PCA (MFPCA) was used to extract features from multiple sensor signals for prognosis. Unfortunately, the extracted features are quite difficult to interpret in practice.

In this study, we focus on HI-based methods. The key idea of the HI-based method is to construct a one-dimensional HI by directly combining multiple sensor signals to characterize the underlying degradation process. Compared with the aforementioned data fusion approaches, HI-based methods are highly desired in practice due to three main reasons. First, the rich literature based on a single sensor signal for degradation modeling and prognostics can be directly applied based on the constructed HI as the HI can be regarded as another single sensor signal but with more information. Second, the constructed HI shows a real-time characterization of the degradation process of a unit, which results in better interpretation than most other data fusion models that behave like a black box by providing only a final prediction result. Third, the one-dimensional HI can be easily visualized to help practitioners make better decisions. In fact, most of the existing prescriptive models, such as maintenance scheduling and spare parts logistics have already assumed such a real-time HI is available when making decisions.

Despite these advantages, great challenges also exist in HI-based methods. One major challenge is that the underlying degradation status is unobservable. To address this issue, Yang *et al.* [43] explicitly expressed the HI of a unit as a deterministic function of time and regressed the multiple sensor signals against the function values. However, this approach failed to capture the stochastic nature of the degradation process. Alternatively, [20]–[22] identified the desired properties of a good degradation signal and constructed the HI in the way such that these desired properties were optimized. Although these methods showed a promising prognostic performance, they were heuristic and could not guarantee to find the optimal combination of sensor signals.

Recently, Song *et al.* [42] developed a new approach that solved the HI construction by the quantile regression technique. While [42] showed that it was theoretically possible to find the best combination of sensor signals for HI construction by solving the quantile regression problem, restrictive assumptions were made to ensure the theoretical properties. For example, [42] modeled the HI by a mixed effect model with the random-effect parameter assumed to be multivariate normally distributed, which thus limited its applications. Also, [42] required to solve a large-scale quantile regression problem, which was time-consuming and might not be able to numerically find the global optimal solution in practice.

Since some sensors may not be related to the underlying degradation status, a sensor selection algorithm is necessary to ensure the effectiveness of the constructed HI and prognostic performance. However, there is still a lack of a systematic approach to identify the informative sensors signals in the current literature of multisensor degradation modeling. Very few studies attempted to provide systematic sensor selection procedures [40]. Nevertheless, these procedures are not generic enough, i.e., they are designed for specific data-level fusion models, and still cannot guarantee to select out the optimal subsets of sensors to recover the underlying degradation status of a unit.

To fill this literature gap, this study aims at developing a more generic HI-based method that allows to derive the optimal combination of sensor signals with greater applicability and also the incorporation of a unified sensor selection procedure.

2.3 Methodology

In this section, we will introduce the proposed data-level fusion method in details. In Sections 2.3.1 and 2.3.2, we describe the formulation of our problem and present the parameter estimation method. Section 2.3.3 elaborates the latent linear model involving the multiple sensor signals and

the failure time. In Section 2.3.4, the adaptive lasso technique is incorporated for sensor selection. Section 2.3.4 discusses several considerations in implementing the proposed method in practice. Finally, in Section 2.3.6, we discuss RUL prediction using the constructed HI.

2.3.1 Problem Formulation

Following most of the existing studies [2], [12], [45], we first provide a definition of failure as the result of degradation. Specifically, let $\eta_i(t)$ denote the underlying degradation status of unit i at time t. Then, the failure time T_i of unit i is defined as the time that the underlying degradation status of unit i first reaches a predefined failure threshold l:

$$T_i = \underset{t}{\operatorname{argmin}}\, \eta_i(t) \geq l \tag{2.1}$$

While the specific form of $\eta_i(t)$ is not required, we consider p linearly independent basis functions $\boldsymbol{\psi}(t) = \left[\psi_1(t), \cdots, \psi_p(t)\right] \in \mathbb{R}^{1\times p}$ and decompose $\eta_i(t)$ as

$$\eta_i(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_i, \tag{2.2}$$

where $\boldsymbol{\Gamma}_i = \left[\Gamma_{i,1}, \cdots, \Gamma_{i,p}\right]^{\mathrm{T}} \in \mathbb{R}^{p\times 1}$ are the coefficients of the basis functions for unit i. For example, if $\boldsymbol{\psi}(t) = [1, t, \ldots, t^{p-1}]$, then $\eta_i(t)$ is represented as the $(p-1)$-order polynomial model. The existing literature (e.g., [21], [22], [42]) often restricted $\eta_i(t)$ to be a special general path model, where $\boldsymbol{\Gamma}_i$ was assumed to follow a p-dimensional multivariate normal distribution. However, the assumption of multivariate normality can be quite limited. First, the symmetry required by the normal distribution may not be satisfied in general. Second, the underlying degradation process should be monotonic [46]; however, the normally distributed $\Gamma_{i,j}$ can have either positive or negative values, which may violate the monotonicity property. In this study, we do not impose any restriction on the specific form of the degradation process $\eta_i(t)$, nor the normality assumption for $\boldsymbol{\Gamma}_i$. In other words, a wide range of degradation models including the

general path models and the stochastic process models can be adopted to describe $\eta_i(t)$. As a result, the proposed method is more generic and can be applied to various situations.

We follow [42] to define the composite HI. In particular, we assume that there exists a fusion function $z(\cdot)$ to recover the underlying degradation status of a unit from the multiple sensor signals with the contamination of a white noise, i.e.,

$$\eta_i(t) = z\big(\boldsymbol{L}_i(t)\big) - \varepsilon_i(t). \tag{2.3}$$

Here, $\boldsymbol{L}_i(t) = \big[L_{i,1}(t), \dots, L_{i,s}(t)\big] \in \mathbb{R}^{1\times s}$ is a vector of the sensor signals collected from s sensors of unit i at time t; $L_{i,j}(t)$ is the jth sensor signal of unit i at time t; and $\varepsilon_i(t) \sim N(0, \sigma_0^2)$ is independent and identically distributed noise. Then the composite HI of unit i at time t, denoted by $h_i(t)$, is defined as

$$h_i(t) = z\big(\boldsymbol{L}_i(t)\big). \tag{2.4}$$

Without loss of generality, in this study, we consider the linear fusion function, i.e.,

$$z\big(\boldsymbol{L}_i(t)\big) = \boldsymbol{L}_i(t)\boldsymbol{w}_0. \tag{2.5}$$

Here, $\boldsymbol{w}_0 = [w_1, \dots, w_s]^T \in \mathbb{R}^{s\times 1}$ is a vector of fusion coefficients to combine multiple sensor signals. In fact, the conventional degradation model for a single sensor signal is only a special case with $z\big(\boldsymbol{L}_i(t)\big) = L_{i,j}(t)$, which assumes the jth sensor can fully characterize the degradation process. In addition, note that nonlinear fusion functions can be approximated in the linear form. In particular, with K basis functions, denoted by $B_k(\cdot)$ $(k = 1, \dots, K)$, a nonlinear fusion function $z(\boldsymbol{L}_i(t))$ can be approximated as

$$z\big(\boldsymbol{L}_i(t)\big) \approx \sum_{k=1}^{K} B_k\big(\boldsymbol{L}_i(t)\big) w_k = \sum_{k=1}^{K} L'_{i,k}(t) w_k\,, \tag{2.6}$$

where $L'_{i,k}(t) = B_k\big(\boldsymbol{L}_i(t)\big)$ is an artificial signal (i.e., transformed features from the original sensor signals). One of the most commonly used methods for the nonlinear mappings are kernel based

methods [47], [48]. However, to limit the scope of this work, we will consider this extension to nonlinear fusion functions in the future study.

To summarize, the HI $h_i(t)$, the sensor signals $\boldsymbol{L}_i(t)$, and the degradation status $\eta_i(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_i$ can be expressed as follows:

$$h_i(t) = \boldsymbol{L}_i(t)\boldsymbol{w}_0 = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_i + \varepsilon_i(t). \tag{2.7}$$

Assume there are m historical units that have failed, and for historical unit i, the sensor signals $\boldsymbol{L}_i(t)$ are measured at time $t = t_{i,1}, t_{i,2}, \cdots t_{i,n_i}$, where n_i is the total number of measurements of unit i. Let $\boldsymbol{h}_i = \left[h_i\left(t_{i,1}\right), \dots, h_i\left(t_{i,n_i}\right)\right]^{\mathrm{T}} \in \mathbb{R}^{n_i \times 1}$ denote a vector of HI for unit i; $\boldsymbol{L}_i = \begin{bmatrix} \boldsymbol{L}_i(t_{i,1}) \\ \vdots \\ \boldsymbol{L}_i(t_{i,n_i}) \end{bmatrix} \in \mathbb{R}^{n_i \times s}$ denote a matrix containing all the sensor signals of unit i; $\boldsymbol{\Psi}_i = \begin{bmatrix} \boldsymbol{\psi}(t_{i,1}) \\ \vdots \\ \boldsymbol{\psi}(t_{i,n_i}) \end{bmatrix} \in \mathbb{R}^{n_i \times p}$ denote a design matrix; and $\boldsymbol{\varepsilon}_i = \left[\varepsilon_i\left(t_{i,1}\right), \dots, \varepsilon_i\left(t_{i,n_i}\right)\right]^{\mathrm{T}} \in \mathbb{R}^{n_i \times 1}$ denote a vector containing errors. Then, (2.7) can be rewritten in the following matrix form:

$$\boldsymbol{h}_i = \boldsymbol{L}_i\boldsymbol{w}_0 = \boldsymbol{\Psi}_i\boldsymbol{\Gamma}_i + \boldsymbol{\varepsilon}_i, \tag{2.8}$$

Our goal is to estimate the fusion coefficients $\boldsymbol{w}_0$. Eq. (2.8) looks similar to the conventional linear regression models at the first glance. However, since the response variable $\boldsymbol{h}_i$ is unobservable, we cannot directly derive $\boldsymbol{w}_0$ using the existing linear regression approaches. Next, we propose a novel method to estimate the fusion coefficients $\boldsymbol{w}_0$.

2.3.2 Estimation of Fusion Coefficients

At first, we regard $\boldsymbol{w}_0$ as known and obtain the least squares estimation of $\boldsymbol{\Gamma}_i$ based on (2.8) as

$$\hat{\boldsymbol{\Gamma}}_i = \left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i\boldsymbol{w}_0. \tag{2.9}$$

Since $\boldsymbol{L}_i\boldsymbol{w}_0$ is normally distributed given $\boldsymbol{\Gamma}_i$ according to (2.8), i.e., $\boldsymbol{L}_i\boldsymbol{w}_0|\boldsymbol{\Gamma}_i \sim N_{n_i}(\boldsymbol{\Psi}_i\boldsymbol{\Gamma}_i, \sigma_0^2\boldsymbol{I})$, the conditional distribution of $\hat{\boldsymbol{\Gamma}}_i|\boldsymbol{\Gamma}_i$ also follows a p-dimensional multivariate normal distribution with mean and variance as

$$\mathrm{E}\left(\hat{\boldsymbol{\Gamma}}_i|\boldsymbol{\Gamma}_i\right) = \left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\mathrm{E}(\boldsymbol{L}_i\boldsymbol{w}_0|\boldsymbol{\Gamma}_i) = \left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\boldsymbol{\Gamma}_i = \boldsymbol{\Gamma}_i, \text{ and}$$

$$\mathrm{Var}\left(\hat{\boldsymbol{\Gamma}}_i|\boldsymbol{\Gamma}_i\right) = \left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\mathrm{Var}(\boldsymbol{L}_i\boldsymbol{w}_0|\boldsymbol{\Gamma}_i)\boldsymbol{\Psi}_i\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1} = \sigma_0^2\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}. \quad (2.10)$$

This distribution relies on the unknown variable $\boldsymbol{\Gamma}_i$ and thus cannot be directly used. To address this challenge, our new idea is to utilize the observable failure time T_i from historical units to characterize the unobservable $\boldsymbol{\Gamma}_i$ according to (2.1). Specifically, recalls that the degradation status is $\eta_i(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_i$, and thus we can write $\boldsymbol{\psi}(T_i)\boldsymbol{\Gamma}_i = l$. This motivates us to investigate the distribution of $\boldsymbol{\psi}(T_i)\hat{\boldsymbol{\Gamma}}_i|\boldsymbol{\Gamma}_i$. Since the failure time T_i can be regarded as a function of $\boldsymbol{\Gamma}_i$, T_i will be a constant given $\boldsymbol{\Gamma}_i$ (later on we will show that the failure threshold l can be set as any positive number and thus l can be treated as known here already). This indicates that $\boldsymbol{\psi}(T_i)\hat{\boldsymbol{\Gamma}}_i|\boldsymbol{\Gamma}_i$ also follows a multivariate normal distribution with mean and variance as

$$E\left[\boldsymbol{\psi}(T_i)\hat{\boldsymbol{\Gamma}}_i|\boldsymbol{\Gamma}_i\right] = \boldsymbol{\psi}(T_i)\boldsymbol{\Gamma}_i = l, \text{ and}$$

$$\mathrm{Var}\left[\boldsymbol{\psi}(T_i)\hat{\boldsymbol{\Gamma}}_i|\boldsymbol{\Gamma}_i\right] = \sigma_0^2\boldsymbol{\psi}(T_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\psi}(T_i)^{\mathrm{T}}.$$

Therefore,

$$\boldsymbol{\psi}(T_i)\hat{\boldsymbol{\Gamma}}_i|\boldsymbol{\Gamma}_i \sim N_p\left(l, \sigma_0^2\boldsymbol{\psi}(T_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\psi}(T_i)^{\mathrm{T}}\right). \quad (2.11)$$

Interestingly, this distribution does not require $\boldsymbol{\Gamma}_i$ to be known. Therefore, we can pretend the realizations of $\boldsymbol{\Gamma}_1, \cdots, \boldsymbol{\Gamma}_m$ are known and use maximum likelihood estimation (MLE) to estimate $\boldsymbol{w}_0$. Let $\boldsymbol{\Gamma}_1^*, \cdots, \boldsymbol{\Gamma}_m^*$ denote the realizations of $\boldsymbol{\Gamma}_1, \cdots, \boldsymbol{\Gamma}_m$ for the historical units, and τ_i denote the observed failure time of unit i. The conditional likelihood is

$$L_p = P\left(\boldsymbol{\psi}(\tau_1)\hat{\boldsymbol{\Gamma}}_1, \cdots, \boldsymbol{\psi}(\tau_m)\hat{\boldsymbol{\Gamma}}_m \middle| \boldsymbol{\Gamma}_1^*, \cdots, \boldsymbol{\Gamma}_m^*\right) = \prod_{i=1}^{m} P\left(\boldsymbol{\psi}(\tau_i)\hat{\boldsymbol{\Gamma}}_i | \boldsymbol{\Gamma}_i^*\right). \tag{2.12}$$

It is straightforward to obtain the log-likelihood function as

$$\log L_p = -\frac{1}{2\sigma_0^2}\sum_{i=1}^{m}\frac{\left(\boldsymbol{\psi}(\tau_i)\hat{\boldsymbol{\Gamma}}_i - l\right)^2}{\boldsymbol{\psi}(\tau_i)(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i)^{-1}\boldsymbol{\psi}(\tau_i)^T} + C, \tag{2.13}$$

where C is a constant. Therefore, we can estimate $\boldsymbol{w}_0$ by maximizing $\log L_p$, i.e.,

$$\begin{aligned}\widehat{\boldsymbol{w}} &= \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{i=1}^{m}\frac{\left(\boldsymbol{\psi}(\tau_i)\hat{\boldsymbol{\Gamma}}_{\mathrm{i}} - l\right)^2}{\boldsymbol{\psi}(\tau_i)(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i)^{-1}\boldsymbol{\psi}(\tau_i)^T} \\ &= \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{i=1}^{m}\frac{\left(\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i\boldsymbol{w} - l\right)^2}{\boldsymbol{\psi}(\tau_i)(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i)^{-1}\boldsymbol{\psi}(\tau_i)^T}.\end{aligned} \tag{2.14}$$

Please note that although the likelihood function is conditioned on $\boldsymbol{\Gamma}_1^*, \cdots, \boldsymbol{\Gamma}_m^*$, the true realizations of $\boldsymbol{\Gamma}_1, \cdots, \boldsymbol{\Gamma}_m$ are not required in the above optimization problem. As a result, we can get the analytical solution of (2.14) as

$$\widehat{\boldsymbol{w}} = l\left(\sum_{i=1}^{m}\boldsymbol{a}_i\boldsymbol{a}_i^{\mathrm{T}}\right)^{-1}\left(\sum_{i=1}^{m}\boldsymbol{a}_i b_i\right), \tag{2.15}$$

where

$$\boldsymbol{a}_i^{\mathrm{T}} = \frac{\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i}{\sqrt{\boldsymbol{\psi}(\tau_i)(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i)^{-1}\boldsymbol{\psi}(\tau_i)^{\mathrm{T}}}} \in \mathbb{R}^{1\times \mathrm{s}}, \text{ and}$$

$$b_i = \frac{1}{\sqrt{\boldsymbol{\psi}(\tau_i)(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i)^{-1}\boldsymbol{\psi}(\tau_i)^{\mathrm{T}}}}.$$

Since the failure threshold l in (2.15) only acts as a scale factor, we can arbitrarily set l to any positive number if it is unknown, e.g., $l = 1$. This is a particularly useful result in that the failure threshold is often unknown and is hard to obtain its exact value in practice. We will further explain

in Section 2.3.6 that setting l to any positive number does not affect the RUL prediction result. Since the closed-form solution in (2.15) only requires the computation of the inverse of a $s \times s$ matrix, the proposed approach is very computationally efficient, and thus can be easily applied even to massive data, i.e., with many historical units.

2.3.3 Latent Linear Model

As we can see, the estimation of fusion coefficients is very similar to the least squares estimation of a linear regression model. The following proposition provides more insights on the proposed approach and presents the latent linear model for HI construction.

Proposition 1: The fusion coefficients $\boldsymbol{w}_0$ satisfies the weighted linear model as

$$\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i \cdot \boldsymbol{w}_0 + \tilde{\varepsilon}_i = l, \quad \forall i = 1, \cdots, m \tag{2.16}$$

where $\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i \in \mathbb{R}^{1\times s}$ can be regarded as covariates, and $\tilde{\varepsilon}_i \sim N\left(0, \boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\psi}(\tau_i)^{\mathrm{T}}\sigma_0^2\right)$ are the mutually independent noises.

To prove this proposition, we can write

$$l = \boldsymbol{\psi}(\tau_i)\boldsymbol{\Gamma}_i = \boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\boldsymbol{\Gamma}_i = \boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}(\boldsymbol{L}_i\boldsymbol{w}_0 - \boldsymbol{\varepsilon}_i)$$

$$= \boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_{\mathrm{i}}^{\mathrm{T}}\boldsymbol{\Psi}_{\boldsymbol{i}}\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i \cdot \boldsymbol{w}_0 + \tilde{\varepsilon}_i.$$

The first equality is due to the definition of failure time in (2.1), and the third equality results from (2.8). Then, it is straightforward to obtain the distribution of $\tilde{\varepsilon}_i$ that is $\tilde{\varepsilon}_i \sim N\left(0, \boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\psi}(\tau_i)^{\mathrm{T}}\sigma_0^2\right)$.

This latent linear model provides a meaningful physical interpretation of the HI construction. Let $\boldsymbol{L}_{i,j}$ denote a vector containing the jth sensor signal of unit i for all measurements. We can write the jth entry of the covariates $\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i$ as $\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_{i,j}$. This entry

can be interpreted as the fitted jth sensor signal at the observed failure time τ_i. Thus, we can consider $\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i \cdot \boldsymbol{w}_0$ as the fitted HI of unit i at the observed failure time τ_i. This implies that the latent linear model connects the HI at the failure time and the failure threshold. Compared with the original linear model in (2.8), we can see that the latent linear model does not require any unobservable variables to estimate $\boldsymbol{w}_0$.

We can easily transform the weighted linear model in (2.16) to an unweighted linear model by multiplying both sides with $\left\{\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\psi}(\tau_i)^{\mathrm{T}}\right\}^{-\frac{1}{2}}$ and obtain

$$\boldsymbol{a}_i^T\boldsymbol{w}_0 + \varepsilon_i^* = b_i l$$

with $\boldsymbol{a}_i$ and b_i as defined in (2.15), and $\varepsilon_i^* \sim N(0, \sigma_0^2)$ are mutually independent noises.

Recall that Gauss Markov theorem says that under certain conditions, the ordinary least squares (OLS) estimator of the coefficients of a linear regression model is the best linear unbiased estimator (BLUE), that is, the estimator that has the smallest variance among those that are unbiased and linear in the observed output variables [49]. In the case that $\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i$ has full rank, all Gauss-Markov assumptions are met in the latent linear model. Thus, this finding indicates that $\widehat{\boldsymbol{w}}$ is the BLUE in such cases. To the best of our knowledge, this is the first and only study that provides the BLUE of the fusion coefficients for HI-based approaches.

2.3.4 Practical Considerations

In practice, it is possible that there is multicollinearity among the entries of $\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i$; then, the minimizer of (2.14) may not converge to $\boldsymbol{w}_0$. Specifically, in such a case, there may exist $\widetilde{\boldsymbol{w}} \neq \boldsymbol{w}_0$ that satisfies $\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_i\widetilde{\boldsymbol{w}} + \tilde{\varepsilon}_i' = l$ for some residuals $\tilde{\boldsymbol{\varepsilon}}_i'$ with smaller variance than $\tilde{\boldsymbol{\varepsilon}}_i$, and thus the estimation will converge to $\widetilde{\boldsymbol{w}}$ rather than $\boldsymbol{w}_0$. For

example, let us consider the case where the signal of the first sensor is constant, e.g., $L_{i,1}(t) = 1$ for $\forall t$ and $\forall i = 1, \cdots, m$, and all entries of the first column of $\boldsymbol{\Psi}_i$ are 1, i.e., $\boldsymbol{\Psi_i} = \left[\mathbf{1}, \widetilde{\boldsymbol{\Psi}}_i\right]$. Then the fitted sensor signal of the first sensor $\boldsymbol{\psi}(\tau_i)\left(\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{\Psi}_i\right)^{-1}\boldsymbol{\Psi}_i^{\mathrm{T}}\boldsymbol{L}_{i,1} = 1$ is also a constant. In this case, the minimizer of (2.14) will be $\widetilde{\boldsymbol{w}} = [l, 0, \cdots, 0]^{\mathrm{T}}$, which perfectly fits the linear model. However, the constructed HI based on $\widetilde{\boldsymbol{w}}$ does not show any trends, and thus fails to provide any meaningful information for prognostics. Similarly, if there exists a linear combination of sensor signals that is almost flat (i.e., no clear degradation trend), the minimizer of (2.14) may converge to the wrong vector $\widetilde{\boldsymbol{w}}$ as well.

If the constructed HI does not show any clear trend, there are three possible strategies to address this issue. First, we can conduct a pre-selection of sensors to make sure that there is no sensor signal nor a linear combination of different sensor signals which does not show any clear degradation trend. Second, we can pre-select sensors that only show consistent increasing/decreasing trend across all units and add sign constraints to $\boldsymbol{w}$ according to the trend information of each sensor signal. In particular, we constrain that the sensors with increasing (decreasing) trends only have positive (negative) fusion coefficients when solving (2.14). In this way, we can avoid the situations where an increasing sensor signal and a decreasing sensor signal cancel out and result in a constant value. The third strategy is inspired by PCA. In particular, we may repeat solving (2.14) as long as the solution constructs a HI without trend, where for the Kth iteration ($K \geq 2$), we add the constraints $\widetilde{\boldsymbol{w}}_k^T\boldsymbol{w} = 0, \forall k = 1,2, \cdots, K-1$. Here $\widetilde{\boldsymbol{w}}_k$ is the optimal solution in the kth iteration which results in a constant HI. In other words, we seek the minimizer of (2.14) only in the space that is orthogonal to the previous solutions. The idea behind this strategy is that $\boldsymbol{w}_0$ should be orthogonal to any $\widetilde{\boldsymbol{w}}$ which constructs a HI without a clear trend. This is because if they are not orthogonal, then we can decompose $\boldsymbol{w}_0$ as $\boldsymbol{w}_0 = \alpha_1\widetilde{\boldsymbol{w}} + \alpha_2\boldsymbol{w}'$ for some

scalars α_1 and α_2, where $\boldsymbol{w}'$ is a vector orthogonal to $\widetilde{\boldsymbol{w}}$. Thus, the constructed HI, $\boldsymbol{L}_i\boldsymbol{w}_0 = \alpha_1\boldsymbol{L}_i\widetilde{\boldsymbol{w}} + \alpha_2\boldsymbol{L}_i\boldsymbol{w}'$ contains a component $\boldsymbol{L}_i\widetilde{\boldsymbol{w}}$ that does not show any information, and only the component $\boldsymbol{L}_i\boldsymbol{w}'$ is informative. We will consider more systematic investigations to avoid the constant HI construction in the future study.

2.3.5 Sensor Selection

Note that if a sensor does not relate to the underlying degradation process, it should be assigned a fusion coefficient of 0. Thanks to the latent linear model developed in Section 2.3.3, a variety of existing variable selection approaches based on a linear model can be directly applied to the multisensor degradation modeling problems. In this subsection, we apply the well-known adaptive lasso proposed by Zou [50] to this latent linear model for sensor selection.

Similar to the popular lasso method [51], the adaptive lasso also uses the l_1-norm of the coefficient as a penalty. The difference is that the adaptive lasso imposes different penalty weights for different regression coefficients. In other words, larger penalty weights are applied to less important covariates. In this way, the adaptive lasso enjoys the oracle property, i.e., it performs as well as if the true underlying model was given in advance.

Recall that since the failure threshold l only acts as a scale factor, we simply set l to 1. After introducing the adaptive lasso to the latent linear model, we estimate $\boldsymbol{w}_0$ by

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{i=1}^{m} \left(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{w} - b_i\right)^2 + \lambda \sum_{j=1}^{s} \delta_j |w_j| \tag{2.17}$$

where λ is a regularization parameter and δ_j is a penalty parameter for the jth sensor. Following the existing literature [50], the penalty parameter is set to $\delta_j = 1/\left|\widehat{w}_j^{LS}\right|^{\gamma}$, where $\widehat{w}_j^{LS}$ is the OLS estimate of $\boldsymbol{w}_0$ and γ is some positive constant . If a sensor is less related to the underlying

degradation process (i.e., smaller $\left|\widehat{w}_j^{LS}\right|$), it is assigned a larger penalty weight (i.e., larger δ_j) which results in a smaller fusion coefficient (i.e., smaller w_j). As a result, the fusion coefficients of non-informative sensors will be forced to 0.

2.3.6 Remaining Useful Life Prediction

Once the estimated fusion coefficients $\widehat{\boldsymbol{w}}$ is derived, we can construct the HI, $\boldsymbol{h}_i = \boldsymbol{L}_i\widehat{\boldsymbol{w}}$, for each historical unit i. Similarly, for an in-service unit r which is partially degraded with the collected signals $\boldsymbol{L}_r$ in real time, we can construct its HI as $\boldsymbol{h}_r = \boldsymbol{L}_r\widehat{\boldsymbol{w}}$. Then, a degradation model for a single sensor signal can be used to analyze the constructed HI and predict the RUL of the in-service unit.

As mentioned before, since we do not make any restrictions on the degradation process $\eta_i(t)$, a variety of degradation models including the general path models and stochastic process models can be used to analyze the constructed HI. As an example, here we show how to analyze the HI based on a popular general path model [1]. Specifically, we consider the general path model as $h_i(t) = \eta_i(t) + \varepsilon_i(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_i + \varepsilon_i(t)$ where $\boldsymbol{\Gamma}_i$ is a random-effect parameter with prior distribution $\boldsymbol{\Gamma}_i \sim G(\cdot)$, and the prior distribution $G(\cdot)$ can be estimated based on historical units. We can then update the posterior distribution of $\boldsymbol{\Gamma}_r$ for the in-service unit r as $P(\boldsymbol{\Gamma}_r|\boldsymbol{h}_r) \propto P(\boldsymbol{h}_r|\boldsymbol{\Gamma}_r)P(\boldsymbol{\Gamma}_r)$. If there is no analytical solution for the posterior distribution, numerical methods such as Monte Carlo Markov Chain can be employed. Therefore, the cumulative distribution function (CDF) of the failure time T_r of unit r is $F_{T_r}(t|\boldsymbol{h}_r) = P(T_r \le t|\boldsymbol{h}_r) = P(\boldsymbol{\psi}(t)\boldsymbol{\Gamma}_r \ge l|\boldsymbol{h}_r)$ according to the definition in (2.1).

From this CDF and (2.15), we can confirm that the failure threshold l only acts as a scale factor and does not affect the RUL prediction results. Specifically, if we replace l with $l' = \xi l$ in (2.15),

where ξ is a positive constant, the estimated fusion coefficient will change from $\widehat{\boldsymbol{w}}$ to $\xi\widehat{\boldsymbol{w}}$. Then, the constructed health index is also scaled by a factor ξ, i.e., $\xi\boldsymbol{h}_i$. According to the RUL prediction procedure described above, it is straightforward to see that the new health index $\xi\boldsymbol{h}_i$ coupled with the new failure threshold ξl will lead to the same CDF of the estimated failure time as the health index $\boldsymbol{h}_i$ coupled with the failure threshold l. This verifies that when the true failure threshold is unknown, we can arbitrarily set l to any positive number.

Since the in-service unit has not failed yet, the CDF needs to be updated in real time given the latest measurement time t_{r,n_r},

$$F_{T_r}\left(t|\boldsymbol{h}_r, T_r > t_{r,n_r}\right) = \frac{P(\boldsymbol{\psi}(t)\boldsymbol{\Gamma}_r \geq l|\boldsymbol{h}_r) - P(\boldsymbol{\psi}\left(t_{r,n_r}\right)\boldsymbol{\Gamma}_r \geq l|\boldsymbol{h}_r)}{1 - P(\boldsymbol{\psi}\left(t_{r,n_r}\right)\boldsymbol{\Gamma}_r \geq l|\boldsymbol{h}_r)}.$$

Since the truncated CDF is skewed, we estimate the failure time $\hat{T}_r$ as the median of $F_{T_r}\left(t|\boldsymbol{h}_r, T_r > t_{r,n_r}\right)$, i.e., $F_{T_r}\left(\hat{T}_r|\boldsymbol{h}_r, T_r > t_{r,n_r}\right) = 0.5$. Thus, the estimated RUL is $\hat{T}_r - t_{r,n_r}$.

As a special case, if $G(\cdot)$ is a multivariate normal distribution, i.e., $\boldsymbol{\Gamma}_i \sim N_p(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, the posterior distribution $P(\boldsymbol{\Gamma}_r|\boldsymbol{h}_r)$ has a close-form expression as

$$\boldsymbol{\Gamma}_r|\boldsymbol{h}_r \sim N(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r), \tag{2.18}$$

where $\boldsymbol{\mu}_r = \left(\frac{\boldsymbol{\Psi}_r^{\mathrm{T}}\boldsymbol{\Psi}_r}{\sigma_0^2} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1}\left(\frac{\boldsymbol{\Psi}_r^{\mathrm{T}}\boldsymbol{h}_r}{\sigma_0^2} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right)$, and $\boldsymbol{\Sigma}_r = \left(\frac{\boldsymbol{\Psi}_r^{\mathrm{T}}\boldsymbol{\Psi}_r}{\sigma_0^2} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1}$. It is then straightforward to obtain the conditional CDF of the failure time as

$$F_{T_r}\left(t|\boldsymbol{h}_r, T_r > t_{r,n_r}\right) = \frac{\Phi\left(g(t)\right) - \Phi\left(g\left(t_{r,n_r}\right)\right)}{1 - \Phi\left(g\left(t_{r,n_r}\right)\right)}$$

Here $\Phi(\cdot)$ is CDF of the standard normal distribution, and $g(t) = (\boldsymbol{\psi}(t)\boldsymbol{\mu}_r - l)/\sqrt{\boldsymbol{\psi}(t)\boldsymbol{\Sigma}_r\boldsymbol{\psi}(t)^{\mathrm{T}}}$ (the detailed proof can be referred to [20]).

2.4 Simulation Studies

In this section, a series of numerical studies are conducted to demonstrate the effectiveness and evaluate the sensitivity of the proposed method using simulated degradation signals. Specifically, we investigate the performance of the proposed method in three different scenarios. Section 2.4.1 introduces how we generate the simulated degradation signals. Section 2.4.2 studies the parameter estimation accuracy, sensor selection accuracy and computational time of the proposed method under an ideal scenario. In Section 2.4.3, we consider the scenario when the unknown failure threshold is a random variable rather than a fixed value. Finally, in Section 2.4.4, the simulation is carried out when only sparse data is available to realize the data challenge in practice.

2.4.1 Data Generation

Without loss of generality, we generate units with a linear degradation process according to

$$\eta_i(t) = \Gamma_{i,0} + \Gamma_{i,1} t, \tag{2.19}$$

where we draw the random-effect parameter from a bivariate normal distribution:

$$\boldsymbol{\Gamma}_i = \begin{pmatrix} \Gamma_{i,0} \\ \Gamma_{i,1} \end{pmatrix} \sim N_2\left(\begin{pmatrix} -1 \\ 2 \end{pmatrix}, \begin{pmatrix} 100 & 1 \\ 1 & 0.5 \end{pmatrix} \right).$$

As mentioned before, since $\Gamma_{i,1}$ follows a normal distribution, it is possible to generate a sample with $\Gamma_{i,1} \leq 0$. In such cases, we discard the sample and generate a new one to ensure the monotonicity, i.e., the underlying degradation processes of all units are increasing. The true failure threshold is set to be $l = 400$. Then, we record the true failure time of unit i, denoted as τ_i, according to (2.1). True HI is generated by adding a random noise as defined in (2.3) to the underlying degradation process in (2.19), i.e., $h_i(t) = \eta_i(t) + \varepsilon_i(t)$ where $\varepsilon_i(t) \sim N(0, 20^2)$.

Each unit has four sensors (i.e., $s = 4$) with the true value of fusion coefficients $\boldsymbol{w}_0 = [w_1, w_2, w_3, w_4]^{\mathrm{T}} = [0.6,\ 0.2,\ -0.5,\ 0]^{\mathrm{T}}$. Four sensor signals are randomly generated as

$$
\begin{aligned}
L_{i,1}(t) &= U_{i,1}^{(1)}\sqrt{t} - U_{i,1}^{(2)}\sin(0.05t) + \varepsilon_{i,1}(t),\\
L_{i,2}(t) &= U_{i,2}^{(1)}t + U_{i,2}^{(2)}\sin(0.1t) + \varepsilon_{i,2}(t),\\
L_{i,3}(t) &= \left(h_i(t) - w_1 L_{i,1}(t) - w_2 L_{i,2}(t)\right)/w_3, \text{ and}\\
L_{i,4}(t) &= U_{i,4}^{(1)}t + U_{i,4}^{(2)} + \varepsilon_{i,4}(t).
\end{aligned}
\tag{2.20}
$$

where $U_{i,1}^{(1)}, U_{i,1}^{(2)}, U_{i,2}^{(1)}, U_{i,4}^{(2)} \sim \text{Uniform}(0, 30)$, $U_{i,2}^{(1)}, U_{i,4}^{(1)} \sim \text{Uniform}(0, 2)$, and $\varepsilon_{i,1}(t), \varepsilon_{i,2}(t), \varepsilon_{i,4}(t) \sim N(0, 20^2)$. Note that the signal of Sensor 3 is calculated according to $\boldsymbol{w}_0$ using $h_i(t)$ and the first two sensors to satisfy (2.7). Sensor 4 is not related to the underlying degradation process, and thus the corresponding fusion coefficient is 0.

All the signals of unit i are recorded at equidistant times $t = 1, \ldots, n_i$, where $n_i = \lfloor \tau_i \rfloor$ is the largest integer less or equal to the failure time τ_i. Figure 2.1 shows the true HI and four signals of three randomly generated units.

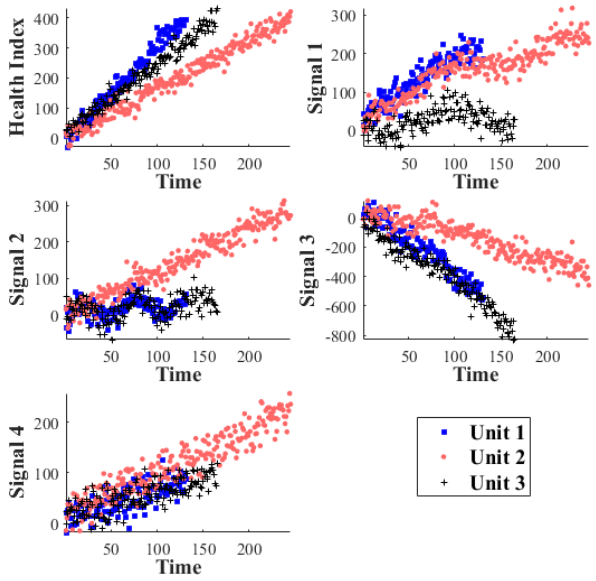


Figure 2.1 Degradation signal plots for the constructed HI and four sensor signals of three randomly generated units.

2.4.2 Ideal Scenario

The first simulation is conducted to verify the parameter estimation performance of the proposed method in the ideal situation. We randomly select m units as the historical units. Based on the sensor signals and the failure time of the historical units, we estimate the true fusion coefficient of each sensor. The procedure is replicated 100 times for each selected value of m. Recall that since l only acts as a scale factor which does not affect the RUL prediction, here we use $l = 400$ when estimating the fusion coefficients; in this way, we can obtain the correct scale of the fusion coefficients and easily compare our estimation with the true values. Figure 2.3 shows the mean and variance of the estimation of the fusion coefficients. The x-axis represents the number of the sampled historical units m. The dotted horizontal line represents the true fusion coefficient of each sensor. The solid and dashed curves represent the mean and one standard deviation of the fusion coefficient estimation, respectively. From Figure 2.2, we can see that the estimation is very accurate and improves as the number of the sampled historical units increases.

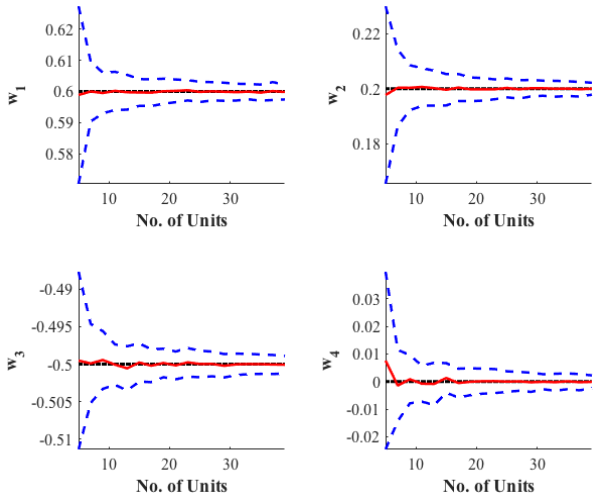


Figure 2.2 Fusion coefficients estimation results for the ideal scenario. The solid line is the mean estimation for each entry of $\boldsymbol{w}_0$. The dashed lines show one standard deviation of the fusion coefficient estimation. The dotted horizontal line is the true value of each entry of $\boldsymbol{w}_0$.

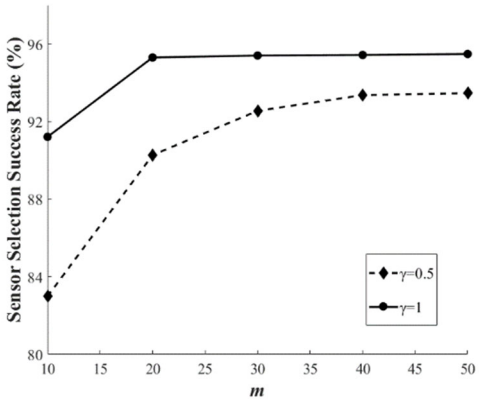


Figure 2.3 The proportion of trials that the proposed method selects the right set of sensors in the ideal scenario.

Table 2.1 Average Computational Time In Seconds For Fusion Coefficients Estimation

m	The Proposed Method (sec)	Benchmark Method (sec)
20	0.020	197.964
40	0.022	274.204
60	0.027	311.350
80	0.034	430.213
100	0.041	492.171

To verify the sensor selection performance of our method, we again randomly select m historical units and repeat the sensor selection for 1000 times for each selected value of m. When applying the adaptive lasso, we set γ to 0.5 and 1, and use five-fold cross validation to search for the optimal λ for a given γ. Figure 2.3 shows the proportion of trials that the proposed method selects the right set of sensors, i.e., only Sensors $1, 2$ and 3. As can be seen from the figure, the accuracy in finding the right set of sensors increases as more historical units are available.

The computational time of the proposed method for fusion coefficients estimation is also measured and compared with the results of the benchmark method: quantile regression data fusion method in [42]. We use [42] as the benchmark method since the method also ensures that the estimated fusion coefficients converge to the true values under some assumptions. The

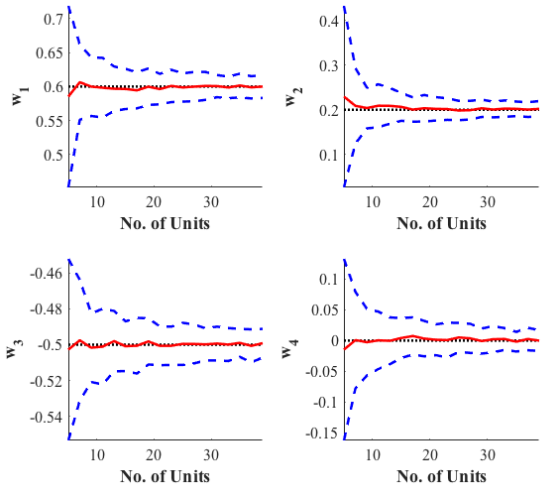


Figure 2.4 Estimation results under the random failure threshold scenario. The solid line is the mean estimation for each entry of $\boldsymbol{w}_0$. The dashed lines show one standard deviation of the fusion coefficient estimation. The dotted horizontal line is the true value of each entry of $\boldsymbol{w}_0$.

computational time measurements of the proposed method and the benchmark method are replicated 50 times for each selected value of m. All simulations are implemented in MATLAB and executed on an Intel Core i5-6300U 2.40 GHz processor with 16 GB RAM. The average computational time of the proposed method and the benchmark method is represented in Table 2.1. We can see that the proposed method requires much less computational time than the benchmark method. This is because the proposed method provides an analytical solution of the fusion coefficients, whereas the benchmark method has to solve a large-scale optimization to estimate the fusion coefficients.

As mentioned earlier, in many real-world applications, it is possible that the prior distribution of $\boldsymbol{\Gamma}_i$ does not follow a multivariate normal distribution. Thus, we conduct additional similar simulations, except that the prior distribution of $\boldsymbol{\Gamma}_i$ is non-normal. Specifically, we consider two cases where each entry of $\boldsymbol{\Gamma}_i$ follows a beta distribution and a gamma distribution, respectively. The result of fusion coefficients estimation is very similar to that with normally distributed $\boldsymbol{\Gamma}_i$ and

thus is omitted here. This further verifies that the proposed method is not limited to the prior distribution of $\mathbf{\Gamma}_i$, which is different from many existing works (e.g., [20]–[22], [42]).

2.4.3 Sensitivity to Random Failure Threshold

In this subsection, a simulation is further carried out to test the proposed method with a relaxation of the assumption that the failure threshold l is a fixed value. In real-world applications, different units indeed may fail at different levels of degradation status [21]. Thus, we generate a new dataset following the same procedures as described in Section 2.4.1, except that the failure threshold is uniformly distributed in $[375,425]$ rather than fixed. We apply the proposed method to the new dataset while still assuming the failure threshold is a fixed value.

Following the same procedure as in the previous subsection, m units are randomly selected as historical units. Then, the fusion coefficients estimation is repeated 100 times for each m, and is shown in Figure 2.4 which indicates that the estimations are still very accurate. This is because, as described in 2.3.6, the estimation of the fusion coefficients does not require to know the exact value of failure threshold.

2.4.4 Sensitivity to Data Sparsity

In practice, it is common that the collected sensor signals are sparse or incomplete due to limited resources for data collection or data losses during transmission. To evaluate the sensitivity of the proposed method to data sparsity, we randomly choose 10 units as the training set. For each unit, we randomly sample a number of measurements to estimate the true fusion coefficients. This procedure is repeated 500 times and the result of the fusion coefficients estimation is shown in Figure 2.5. The x-axis means the number of available measurements for each historical unit. The dotted horizontal lines represent the true value for each entry of $\boldsymbol{w}_0$, and the solid and dashed

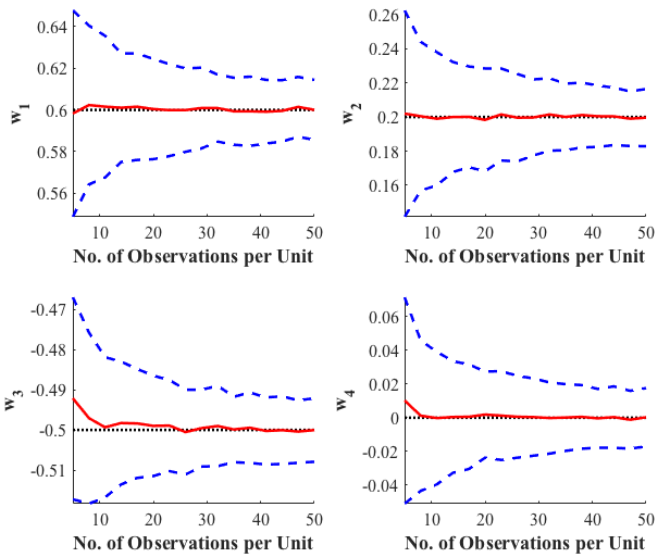


Figure 2.5 Estimation results when only sparse measurements are available. The solid line is the mean estimation for each entry of $\boldsymbol{w}_0$. The dashed lines show one standard deviation of the fusion coefficient estimation. The dotted horizontal line is the true value of each entry of $\boldsymbol{w}_0$.

curves are the mean and one standard deviation of the estimation of the fusion coefficients. Figure 2.5 shows that as the number of available sensor measurements per unit increases, the fusion coefficients estimation becomes more accurate.

2.5 Case Study

In this section, we employ the proposed method to predict the RUL of aircraft gas turbine engines. In addition, the results are compared with the benchmark approach: quantile regression data fusion model in [42]. The benchmark model utilizes quantile regression for HI construction and has been shown to outperform other existing data-level fusion methods (e.g., [20]–[22], [41]) as well as each single sensor signal in RUL prediction for the same dataset.

2.5.1 Overview of the System and Dataset

The degradation-based sensor data is generated from C-MAPSS, a software widely used to simulate the degradation of turbofan aircraft engines [52]. The degradation in engine performance is due to wear and tear according to the usage pattern. To make it more realistic, each unit starts with a different degree of initial wear and manufacturing variation.

At each measurement, a total of 21 sensor signals are collected. The detailed descriptions of these 21 sensors are given in Table 2.2. The dataset consists of 100 historical units (i.e., $m = 100$) that include a total of 20631 measurements (i.e., $\sum_{i=1}^{m} n_i = 20631$) and 100 in-service units that

Table 2.2 Detailed description of sensors and environmental settings.

Type	Symbol	Description	Units
Sensor	T2	Total temperature at fan inlet	°R
	T24	Total temperature at LPC outlet	°R
	T30	Total temperature at HPC outlet	°R
	T50	T50 Total temperature at LPT outlet	°R
	P2	Pressure at fan inlet	psia
	P15	Total pressure in bypass-duct	psia
	P30	Total pressure at HPC outlet	psia
	Nf	Physical fan speed rpm	rpm
	Nc	Physical core speed rpm	rpm
	epr	Engine pressure ratio (P50/P2)	--
	Ps30	Static pressure at HPC outlet	psia
	phi	Ratio of fuel flow to Ps30	pps/psi
	NRf	Corrected fan speed	rpm
	NRc	Corrected core speed	rpm
	BPR	Bypass Ratio	--
	farB	Burner fuel-air ratio	--
	htBleed	Bleed Enthalpy	--
	Nf_dmd	Demanded fan speed	rpm
	PCNfR_dmd	Demanded corrected fan speed	rpm
	W31	HPT coolant bleed	lbm/s
	W32	LPT coolant bleed	lbm/s
Environmental Variable	--	Altitude	ft
	--	Mach number	--
	TRA	Throttle resolver angle	--

include a total of 13096 measurements. All units have a single failure mode and operates under the same environmental condition.

The sensor signals for each historical unit are collected until failure, whereas the sensor signals for each in-service unit are truncated at some random point prior to its failure. The failure time of all historical units and the actual RUL of all in-service units are also recorded.

2.5.2 Data Preprocessing

We first rule out 10 sensors to avoid the construction of constant HI as discussed in Section 2.3.4. Specifically, if the sensor does not exhibit consistent increasing or decreasing trend in all historical units or if its variance is less than 10^{-4}, it is excluded. As a result, 11 candidate sensors are selected out of 21 sensors; including T24, T50, P30, Nf, Ps30, phi, NRf, BPR, htBleed, W31, and W32. To achieve a fair comparison, for these sensors, we then apply a log-transformation and standardize all logged sensor signals in the same way as in [42].

2.5.3 Results and Comparison

The quadratic degradation model (i.e., $\boldsymbol{\psi}(t) = [1, t, t^2]$) is applied since it provides a good fitting based on the existing studies [41], [21], [42]. At first, we conduct sensor selection based on the historical units. When implementing the adaptive lasso, we consider three choices for γ: 0.5, 1 and 2. The five-fold cross validation is employed to find the optimal λ for a given γ. As a result, $\gamma = 0.5$ and $\lambda = 0.015$ are chosen as an optimal pair and all 11 sensors are selected as informative sensors. Our sensor selection result turns out to agree with previous studies [20]–[22], [42], which manually selected the 11 sensors. The estimates of the fusion coefficient for each sensor are presented in Table 2.3. Note that since the failure threshold l does not affect the RUL prediction,

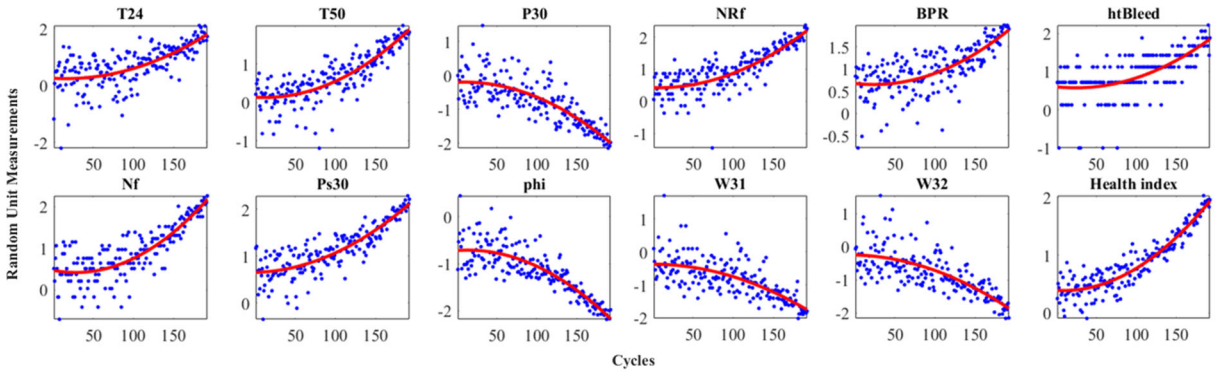


Figure 2.6 Degradation signals plot and model fittings for 11 selected sensors and the constructed HI of a randomly selected in-service unit.

Table 2.3 Estimated Fusion Coefficients $\widehat{\boldsymbol{w}}$ for Each Sensor

Sensor	Value	Sensor	Value
T24	0.184	NRf	-0.058
T50	0.289	BPR	0.146
P30	-0.074	htBleed	0.131
Nf	-0.071	W31	-0.115
Ps30	0.129	W32	-0.245
phi	-0.140		

here we arbitrarily set $l = 2$ in the fusion coefficient estimation. The HI of each unit is then constructed using the estimated fusion coefficients.

Figure 2.6 compares each individual sensor and the constructed HI of a randomly selected in-service unit. From the figure, we can see that the constructed HI provides a much better model fitting result than original single sensors.

Based on the constructed HI, we then predict the RUL for each in-service unit. To provide a fair comparison, we also adopt the assumption in [42] that the random-effect parameter $\boldsymbol{\Gamma}_i$ follows a multivariate normal distribution. In assessing the prediction error, we use the following error criteria:

$$\text{error} = \frac{|\text{Estimated RUL} - \text{Actual RUL}|}{\text{Actual Failure Time}}. \tag{2.21}$$

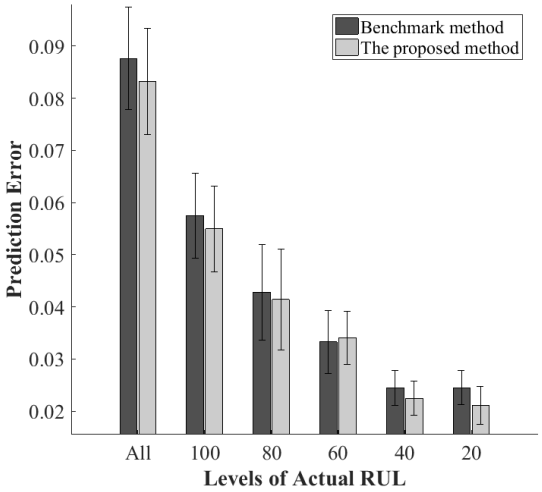


Figure 2.7 Comparison results of the RUL prediction errors for the in-service units by using the benchmark method and the proposed method.

Since the sensor measurements for different in-service units are truncated at different time points, we compare the prediction error at different levels of actual RUL, as shown in Figure 2.7. For example, the level "80" on the x-axis represents the prediction error of the in-service units whose actual RUL is equal to or less than 80. The prognostic results from the proposed method and the benchmark method are represented in Figure 2.7. The bars refer to the average of the prediction errors and the error bars show one standard deviation. We can see that the proposed method yields lower overall RUL prediction errors than the benchmark method. In addition, the advantage of the proposed method seems to be more significant when the in-service units approach the end of life, which is very important for practical applications.

2.6 Discussion and Conclusion

The utilization of multiple sensors in condition monitoring has received much attention in recent years. In particular, different sensor signals may have different degrees of relevance to the degradation process. Hence, the key challenges in multisensor degradation modeling are twofold.

One is how to select the informative sensors. The other is how to effectively combine the information from the selected sensor signals.

In this study, we propose a generic HI-based data fusion method that constructs a HI by automatically selecting and combining the multiple sensor signals to better understand the degradation process. Unlike existing HI-based approaches, we propose a latent linear model for HI construction and a systematic sensor selection procedure, which resolve these abovementioned two challenges in a unified manner. The proposed method has the following contributions. First, the estimated fusion coefficients converge to the true value. In fact, by solving the latent linear model, our method obtains the BLUE. To the best of our knowledge, this is the only method that has this nice property when dealing with multisensor degradation signals. Second, the proposed method requires much less computational time since the closed-form solution of the fusion coefficients is available. Third, compared to the previous HI-based methods, the proposed method is more generic with greatly relaxed assumptions. Specifically, a variety of degradation models can be employed to represent the degradation process and the random-effect parameter can have any distribution. Fourth, the proposed method can accurately predict RUL without requiring to know the exact value of the failure threshold. Finally, variable selection methods for linear regression models can be directly adopted to the latent linear model for a systematic sensor selection.

The effectiveness and the sensitivity of the proposed method under different scenarios were investigated through simulation studies and the case study. The simulation results showed that the proposed method estimated the fusion coefficients accurately even when the failure threshold was not fixed or only sparse measurements were available. For the case study, the degradation dataset

of aircraft engines was used to evaluate the proposed method, which showed that our method had better RUL prediction performance compared to the existing benchmark method.

There are several important topics for future research. First, an extension of the proposed data fusion method to cases with multiple failure modes and multiple operation conditions will be of interest in our future research. Second, more systematic and effective approaches are needed to tackle the constant HI construction issue as described in Section 2.3.4. Finally, to highlight our main idea, a linear fusion function is adopted in this study when constructing the HI. It would be interesting to study how to construct the nonlinear mappings between the HI and each individual sensor signal.

Chapter 3 A Bayesian Deep Learning Framework for Interval Estimation of Remaining Useful Life in Complex Systems by Incorporating General Degradation Characteristics

3.1 Introduction

Unexpected system failures may result in severe consequences, including operation downtime, significant repair or replacement cost, and catastrophic safety hazards. To prevent these consequences, it is important to accurately predict the remaining useful life (RUL) of a partially degraded system. Existing literature on RUL prediction has been focusing on inferring the unobservable degradation process of a simple system based on the collected sensor signals, in which a single degradation signal is often assumed to be able to fully characterize the degradation process, and there is only a single failure mode or a single operational condition involved. Under these assumptions, several popular methods are developed to characterize the underlying degradation process and to predict the failure time, such as the general path models [1], [2], stochastic process models [3], and machine learning approaches that apply support vector machine (SVM) [15], k-nearest neighbors [16], and decision trees [17]. General path models and stochastic process models assume that degradation signals can be modeled as mixed effect models and stochastic processes, respectively. Meanwhile, conventional machine learning approaches often require extensive feature engineering based on domain-specific knowledge to achieve satisfactory prognostic performance. However, these assumptions and requirements are often difficult to justify

and satisfy in complex systems which may involve multiple sensor signals, multiple failure modes, and multiple operational conditions [53].

Recent advances in sensor technologies have dramatically accelerated the use of multiple sensors to monitor the degradation process of a complex system simultaneously. As a system degrades, different kinds of sensor signals evolve over time reflecting the severity of the degradation process in different ways. These sensor signals are often correlated and each sensor signal may contain only partial or even no information about the underlying degradation status [22], [41], [42]. Moreover, a modern engineering system often consists of multiple and mutually interactive subsystems or components and may involve multiple failure mechanisms [23], [24]. Although multiple sensors can be installed to fully capture the multiple failure mechanisms of a complex system, it is still challenging to understand how each failure mechanism affects the changes in sensor signals. Furthermore, modern complex systems commonly operate under multiple operational conditions to meet certain task requirements [25]. In such cases, depending on different operational conditions such as workload assigned or environmental conditions (temperature, pressure, etc.), the degradation processes may be significantly accelerated or decelerated, while the exact effects of the operational conditions are difficult to characterize. As a result, for modern complex systems, it may not be feasible to explicitly formulate the mathematical relationships between the collected degradation signals and the underlying degradation processes or the RULs.

Accordingly, deep learning (DL) approaches have gained increasing attention in prognostics due to their outstanding performance in dealing with complex systems [54]. In complex systems, it can be very difficult to manually extract high-level, meaningful features capturing the intricate relationships between multiple sensor signals, multiple failure modes, and multiple operational

conditions. DL avoids this labor-intensive feature engineering by directly learning features from data itself. The "deep" in DL stands for the idea of many successive layers (more than one hidden layer) to express complex features. DL effectively structures the feature space by expressing complex features in terms of other simpler features. For example, one popular DL approach developed for RUL estimation is to construct neural networks (NNs) with multiple layers which take the most recent degradation signals or operational conditions as inputs and provide the point estimations of RULs or the life percentage as outputs [55], [56].

While DL has been widely applied in various fields and has achieved remarkable performance, it can result in significant errors to directly apply existing NNs developed for different purposes like image processing or speech recognition to prognostics. This is because there are distinctive characteristics (referred to as "General Degradation Characteristics" in this study) that differentiate degradation modeling and prognostics from other DL applications, and thus must be considered. First, compared to other applications using DL, limited amount of training data is more common in degradation applications due to low signal sampling frequency or high cost for data acquisition. This issue of limited data availability can easily lead to the incorrect modeling and increase the uncertainties in model outputs. Second, degradation processes are stochastic in nature, and thus a systematic approach is necessary to account for the randomness caused by various sources such as sensor measurement errors and system-to-system variations. The limited data availability and the stochastic nature of degradation processes make it impossible to provide point estimations of RULs with absolute certainty. Alternatively, interval estimations of RULs should be obtained to better interpret and quantify the uncertainties involved in RUL estimations, laying a foundation for subsequent risk analysis and maintenance decision making [6]. Based on the uncertainty quantification of the estimated RUL, practitioners can then detect abnormal cases and decide

whether further analysis or more data is required. Last, there are other general characteristics of degradation processes that are desired in practice. For example, it is often desired to make a more accurate RUL prediction as a system approaches the end of life to avoid unexpected sudden failure. Such characteristics make the analysis of multi-sensor degradation signals for degradation modeling and prognostics fundamentally different from the analysis of multiple sensor signals for general prediction purposes.

The objective of this study is to propose a novel Bayesian DL framework explicitly designed for degradation modeling and prognostics in complex systems which provides more accurate interval estimations of RULs. The major contributions of this work are summarized as follows. First, the proposed method does not assume any particular type of degradation processes or domain-specific prior knowledge such as a failure threshold, and thus it can be easily adopted to a variety of complex systems that may involve multiple sensor signals, multiple failure modes, and multiple operational conditions. Second, the proposed DL framework provides interval estimations of RULs. As a comparison, most of the existing DL-based prognostic approaches only produce point estimations of RULs which limit their use in real-world applications. Third, to the best of our knowledge, this is the first paper which systematically quantifies two types of uncertainties embedded in degradation modeling and prognostics. In particular, the first part of the proposed framework which is a Bayesian deep NN models the uncertainties resulting from the unknown model parameters. Meanwhile, the second part of the framework quantifies the uncertainties stemming from the stochastic nature of degradation processes. These two parts are trained simultaneously in a unified manner. Last, the proposed method provides the end-to-end solutions of the RULs, by taking observable degradation data like multiple sensor signals and operational variables as inputs and RULs as outputs. In particular, the method is explicitly designed to

incorporate general characteristics of degradation processes, whereas the hand-crafted feature extraction procedures are avoided which can be inaccurate and time-consuming to be obtained in complex systems.

The rest of this study is organized as follows. Section 3.2 reviews the existing DL-based prognostic approaches and the related studies on Bayesian NNs. Section 3.3 describes the details of the proposed Bayesian DL framework for prognostic applications. Section 3.4 evaluates the proposed method using the degradation dataset of aircraft gas turbine engines and compares the results with the existing benchmark methods. Section 3.5 draws a conclusion and discusses future research topics.

3.2 Literature review

This section contains two parts. In Section 3.2.1, we review existing DL-based prognostic approaches. Then, in Section 3.2.2, we briefly introduce the concept and review the recent literature on Bayesian NNs.

3.2.1 DL-based prognostics

DL has attracted great attention from many researchers and practitioners due to its simplicity, flexibility, and general applicability [54]. DL constructs models with multiple levels of layers which are composed of nonlinear processing neurons. Each neuron learns to transform the input representation into the output representation at a higher, more abstract level. By stacking multiple layers, we can build a network that learns a very complex function. Recently, more and more studies have been conducted on DL-based prognostics, especially for complex systems where it is difficult to build physics-based representations of degradation processes or to justify restrictive assumptions made on the processes.

In the past, most of the literature on DL-based prognostics used the standard feed-forward neural network (FFNN) structure, i.e., neurons arranged in layers have only forward connections to neurons in the subsequent layers. For example, Tian *et al.* [55] developed a NN with one input layer, two hidden layers, and one output layer. The NN takes the cycle time and degradation signals at the current and the previous measurements as the inputs, and the normalized life percentage at the current time as the outputs. However, the standard FFNNs cannot effectively capture the characteristics of degradation signals which are time-series data and often multichannel.

To overcome this issue, two types of NNs are growing in popularity in DL-based prognostics; convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNN is a specialized kind of NNs that uses sequential operations of convolution and pooling to extract abstract features [57]. While some CNNs with 1D inputs were applied to model a single degradation signal in simple systems, to date, CNNs for prognostics primarily take inputs of 2D representations of degradation processes. Zhu *et al.* [58] first extracted 2D features from vibration signals and fed the features to a CNN to predict RULs of bearings. In some multi-sensor scenarios, existing CNNs designed 2D inputs where the first dimension specifies the sensor index and the second dimension refers to time [56].

RNN and its variants, e.g., Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM), focus more on processing sequential data like speech, text, or signal by sharing the same weights across every time step [59]. Guo *et al.* [26] applied an RNN to estimate the degradation percentage of bearings based on features extracted from the vibration signals. Huang *et al.* [18] recently proposed a Bi-Directional LSTM for RUL prediction, where two Bi-Directional LSTM networks extract features from raw sensor signals and environmental variables, respectively, and the third network outputs the final RUL estimations based on these features.

Besides standard FFNNs, CNNs, and RNNs, other NNs have also been developed for RUL predictions. Zhang *et al.* [60] integrated a multiobjective evolutionary algorithm with deep belief networks to build an ensemble model for RUL predictions. Multiple deep belief networks were trained simultaneously to optimize two conflicting objectives: accuracy and diversity. In Liao *et al.* [61], a restricted Boltzmann machine was used to predict RULs, where a new regularization term was proposed to automatically extract useful features.

Despite high accuracy and general applicability, great challenges still exist when applying DL to prognostics in practice. First, while interval estimations of RULs are essential in many real-world degradation modeling and prognostic applications, most of the existing DL-based prognostic models provide only point estimations of RULs. A few studies attempted to develop NNs which can provide interval estimations of RULs. However, these methods often consist of two separate steps rather than one unified procedure, which may introduce additional uncertainties and deteriorate the performance. For example, Gebraeel and Lawley [62] first used multiple FFNNs to produce point estimations of RULs and combined these results with parametric degradation forms to obtain the RUL distributions. Recently, Huang et al., [63] combines deep learning and particle filter to provide the interval estimations of RULs of complex systems. Second, most of the existing NNs designed for RUL estimations are purely data-driven and very few studies have considered the general characteristics of degradation processes. Garga *et al.*, [64] expressed domain-knowledge related to the degradation process in the form of rules and used it to train a FFNN. Yet, the systematic procedure for RUL prediction was not included, and the prediction accuracy was not quantified. Peng et al., [65] proposed a Bayesian Deep Learning framework to provide interval estimations of RULs. However, this study models only the uncertainties resulting from the

unknown model parameters, but not the uncertainties from the stochastic nature of degradation processes.

To fill this literature gap, this study aims at developing a novel DL-based prognostic approach that constructs a Bayesian DL framework with the incorporation of the general characteristics of degradation processes, which provides higher accuracy, general applicability, and uncertainty quantifications of RUL estimations in complex systems.

3.2.2 Bayesian Neural Networks

Unlike most DL methods treating NNs as deterministic functions, Bayesian DL views the NN as a probabilistic model and quantifies the uncertainties with Bayesian inference. A Bayesian NN places a prior distribution over the network's weights and yields posterior distributions over the weights given the observed data. Although Bayesian NNs offer higher accuracy, robustness to over-fitting, and uncertainty quantification, one major challenge is that existing methods are often computationally expensive, as exact Bayesian inference is computationally intractable for most of the NN structures.

To overcome this challenge, various approximations have been proposed including Laplace approximation [66], variational inference [67], [68], expectation propagation [69], and Hamiltonian methods [70]. For example, Graves [67] applied data sampling-based variational inference and improved the scalability to a large amount of data. Unfortunately, the method showed unsatisfactory performance in practice due to noises from Monte Carlo approximations within the stochastic gradient computations. Alternatively, Soudry *et al.* [69] used expectation propagation for modeling networks with binary weights; however, the method did not provide posterior variance estimations for continuous weights. In addition, one common drawback is that these approaches still come with a significant computational cost.

To reduce the computational cost, bootstrapping has been applied by separately training many randomly initialized models on the different subsets of training data [71]. Although this approach is more computationally efficient than the conventional Bayesian NN approaches, it can easily lead to unreliable uncertainty estimations. For instance, it has been shown that the prediction on the test data very far from the training data can be still associated with unjustified high confidence [72].

In order to overcome these drawbacks, Gal and Ghahramani [68] Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning recently developed Monte Carlo dropout (MC dropout), which combines approximate Bayesian NN inference with dropout. Dropout is a stochastic regularization technique widely used to prevent co-adaptation and overfitting in NNs [73]. The key idea of dropout is to randomly remove (drop) some neurons and their connections from the NN during each iteration of stochastic gradient descent. At the test time, all neurons are present, and the trained weights are multiplied by the probability of the present (not dropped). In contrast to conventional dropouts which drop neurons only at training time, MC dropout trains a model with dropout and performs dropout at the test time as well. In this way, dropout can be interpreted as a variational inference approximation under certain conditions [68]. MC dropout has drawn much attention recently due to its simplicity, scalability, and computational efficiency compared to the conventional Bayesian NN approaches. In this study, we will employ MC dropout to model the uncertainty stemming from unknown model parameters.

3.3 Methodology

In degradation modeling and prognostics, there are two main types of uncertainties: the uncertainties arising from the stochastic nature of degradation processes and the uncertainties resulting from the unknown model parameters. Unlike most of the existing prognostic approaches which capture only the former type of uncertainties or mix the two types of uncertainties together, the proposed method separately models both types of uncertainties. Figure 3.1 shows the illustration of the proposed framework. The first part (dark grey blocks) is the Bayesian deep NN (BDNN) establishing the relationship between the collected degradation data and the RUL. The second part (a light grey block) is the FFNN which takes the RUL as inputs to quantify the uncertainties resulting from the stochastic nature of degradation processes. The outputs of the first and second parts are then combined to obtain the loss. This means that these two parts are *jointly*

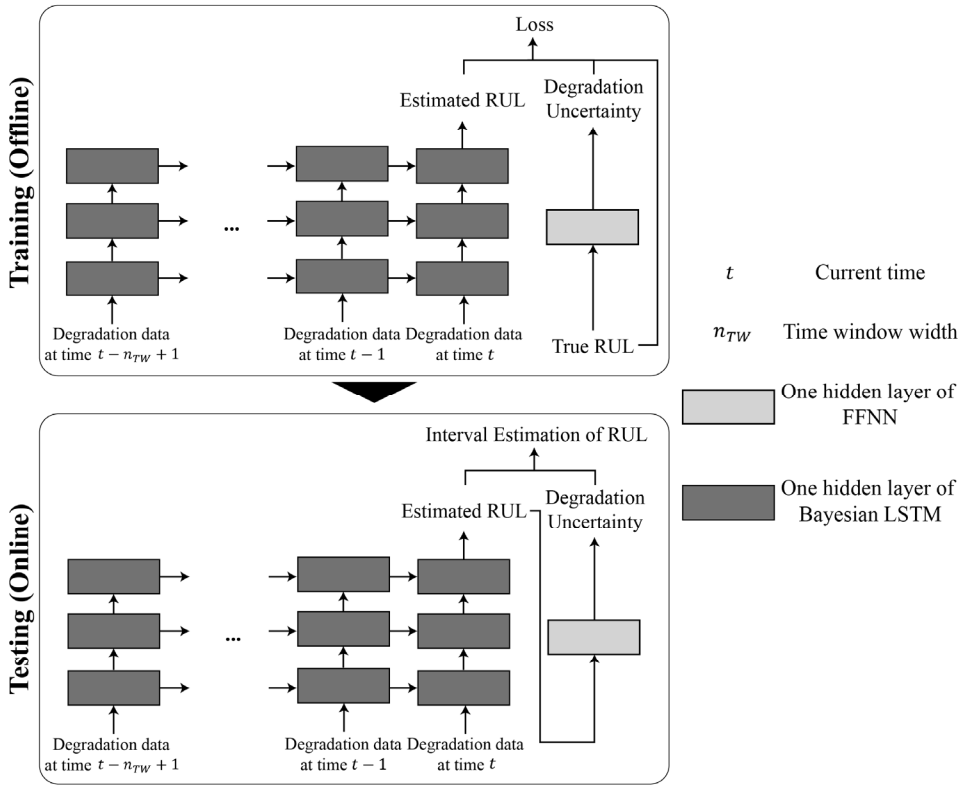


Figure 3.1 Proposed Bayesian DL framework.

trained in a systematic manner to maximize the prognostic performance. At the test time, true RUL values are unknown, and thus the outputs of the trained BDNN, i.e., the estimated values of RULs, serve as the inputs of the FFNN. Section 3.3.1 first discusses the concepts of the two types of uncertainties in degradation modeling and prognostics and the differences between them. In Section 3.3.2, we provide the details of the proposed Bayesian DL framework described in Figure 3.1 including the loss function, the quantifications of the two types of uncertainties, and the interval estimations of RULs. Section 3.3.3 explains the data augmentation procedure referred to as sliding time window, which can alleviate the limited data availability issue and improve the prognostic performance.

3.3.1 Two types of uncertainties in DL-based prognostics

The proposed Bayesian DL framework provides end-to-end solutions, i.e., the network takes the most recent observable degradation data, such as multiple sensor signals, operational variables, and cycle time, as an input $\boldsymbol{x}$ and the corresponding RUL as a target y. Given the value of input $\boldsymbol{x}$, the corresponding target value y can be expressed as

$$y = f^{\boldsymbol{W}}(\boldsymbol{x}) + \varepsilon \tag{3.1}$$

where $f^{\boldsymbol{W}}(\boldsymbol{x})$ denotes a network output with weight parameters $\boldsymbol{W}$ and $\varepsilon \sim N(0, \sigma^2)$ is the additive Gaussian noise term. Under this formulation, the uncertainties in applying DL to prognostics can be classified into two types. First, there are uncertainties from the unknown weight parameters $\boldsymbol{W}$, and we call it '*weight uncertainty*'. The limited data availability discussed in Section 3.1 can cause large weight uncertainties. Second, there are uncertainties which account for the stochastic nature of degradation processes captured by σ^2, and we refer to it as '*degradation uncertainty*'. System-to-system variability or signal measurement error can be considered as the degradation uncertainty.

Existing literature on degradation modeling and prognostics often formulates the RUL prediction problem into two steps. First, the observed degradation signals are expressed as a function of underlying degradation status plus an error term. Then, the failure time is predicted as the time when the projected underlying degradation status exceeds the predefined failure threshold. The failure threshold is commonly assumed to be fixed and known as priori; however, different systems may actually fail at different threshold values and the exact value of the failure threshold is often unknown, difficult to obtain and even does not exist in practice [21], [74]. Alternatively, this study directly links the observed degradation data to the corresponding RULs by considering the degradation uncertainty as formulated in (1). In this way, the degradation uncertainty possesses a unique characteristic in degradation modeling and prognostics: the monotonic relationship with RULs. As the system degrades over time (the RUL decreases), the corresponding degradation uncertainty is also expected to decrease. This is because, when the system is in the initial phase (e.g., the non-defective stage with a large RUL), the degradation signals do not show a significant trend and have large system-to-system variations due to different initial wears and manufacturing differences, which makes it more challenging to predict the exact RUL [20], [75]. In contrast, as the system degrades over time, the observed sensor signals present more significant degradation trends or patterns, which makes it possible to conduct more accurate RUL predictions. It should be emphasized that the monotonicity here is different from the monotonic trends of degradation signals discussed in the existing literature [20], [21], [76]. It is common that a degradation signal is noisy and non-monotonic, but shows clearer degradation trends or patterns as a system approaches its end of life.

There are several additional advantages of explicitly modeling the relationship between the degradation uncertainty σ^2 and RUL to be monotonic. First, it drives the model to achieve the

desired behavior which allows imposing larger penalties on the RUL prediction errors as the systems get closer to failures. This characteristic is critical to prevent sudden breakdown in real-world applications. Second, the proposed approach is expected to act more consistently for prognostics as it assigns similar uncertainties to similar values of estimated RULs, which allows practitioners to better interpret and understand the prognostic results. Third, the monotonic relationship between σ^2 and RUL is a general characteristic of degradation processes, which ensures the wide applicability in different complex systems.

3.3.2 Proposed network

In this section, we will explain how we design the Bayesian DL framework to systematically model the two types of uncertainties described in Section 3.3.1 and produce more accurate RUL estimations.

3.3.2.1 Loss function

Suppose there are N training samples. Sample $n \in \{1, \dots, N\}$ contains degradation data measured over T_n equidistant time steps, where $\boldsymbol{x}_{n(t)}$ denotes the degradation data in sample n measured at time step t, $\boldsymbol{x}_n = \{\boldsymbol{x}_{n(1)}, \boldsymbol{x}_{n(2)}, \dots, \boldsymbol{x}_{n(T_n)}\}$ contains all degradation data measurements in sample n, and y_n denotes the corresponding RUL with respect to the time step T_n. We can write the negative log-likelihood of sample n according to (3.1) as

$$-\log p\left(y_n \middle| f^{\boldsymbol{W}}(\boldsymbol{x}_n)\right) \propto \frac{1}{2\sigma^2}\{f^{\boldsymbol{W}}(\boldsymbol{x}_n) - y_n\}^2 + \frac{1}{2}\ln \sigma^2. \tag{3.2}$$

In most of the existing literature on DL, the variance of the error term σ^2 is often fixed and ignored. Thus, minimizing the sum of squared errors, i.e., $\{f^{\boldsymbol{W}}(\boldsymbol{x}_n) - y_n\}^2$, is a common way to optimize the parameter $\boldsymbol{W}$. However, as explained in Figure 3.1, in this study, we aim at minimizing the

entire negative log-likelihood, while designing a separate FFNN to represent the monotonic relationship between the degradation uncertainty σ^2 and the RUL y. Let σ_n^2 denote the degradation uncertainty of sample n. Considering that the unconstrained σ_n^2 may explode and result in very large degradation uncertainties, we propose to add a l_2 regularization term of σ_n^2 to the loss function. As a result, based on the training sets of $\boldsymbol{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ and $\boldsymbol{Y} = \{y_1, \dots, y_N\}$, the regularized loss takes the form

$$l = (1-\lambda)\left(\frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{2\sigma_n^2}\{f^{\boldsymbol{W}}(\boldsymbol{x}_n) - y_n\}^2 + \frac{1}{2}\ln\sigma_n^2\right)\right) + \lambda\left(\sum_{n=1}^{N}|\sigma_n^2|^2\right), \qquad (3.3)$$

where $\lambda \in (0,1)$ is a tuning parameter which can be determined by cross validation. From the term $\frac{1}{2\sigma_n^2}\{f^{\boldsymbol{W}}(\boldsymbol{x}_n) - y_n\}^2$ in (3.3), we can see that the samples with smaller degradation uncertainties would contribute more to the loss function. Since systems with smaller RULs (i.e., closer to failures) are modeled to have smaller degradation uncertainties, when minimizing the loss function, the proposed model is naturally encouraged to place more weights on those systems, which are more crucial for practical applications. At the same time, the model is discouraged from reducing l by assigning large σ_n^2 through the term $\frac{1}{2}\ln\sigma_n^2$ and the regularization term.

3.3.2.2 Feedforward Neural Network – Quantification of degradation uncertainties

Recall that the proposed Bayesian DL framework mainly consists of two parts: the BDNN and the FFNN. In this subsection, we will first explain the FFNN part which establishes the monotonic relationship between the RUL y and the degradation uncertainty σ^2. Since the true values of the RULs of test samples are unknown, at the test time, the outputs of the Bayesian deep LSTM, i.e., the estimated RULs, are used as the inputs of the FFNN. We propose to train the FFNN to predict the log variance $\zeta = \log(\sigma^2)$ instead of σ^2. This is more numerically stable by avoiding a division

by zero in the loss function. Also, the exponential transformation $\exp(\zeta)$ would result in a valid positive value for the variance σ^2. As both the inputs y and outputs ζ of the FFNN are scalar values, we can use the shallow structure, i.e., one hidden layer with H_1 neurons, to model their relationships:

$$\zeta = v_0 + \sum_{h=1}^{H_1} v_h \cdot \varphi(u_{h0} + u_h y),$$

where $\varphi(\cdot)$ is a nonlinear activation function, u_{h0} and u_h denote the bias and weight between the input layer and hidden layer, and v_0 and v_h denote the bias and weight between the hidden layer and output layer, respectively. As a demonstration, here we use a sigmoid activation function $\varphi(a) = 1/(1 + e^{-a})$. By applying the chain rule, we get the following first order derivatives:

$$\frac{d\zeta}{dy} = \sum_{h=1}^{H_1} \frac{d\zeta}{da_h} \frac{da_h}{dy},$$

where $a_h = u_{h0} + u_h y$. Since the sigmoid $\varphi(a)$ is monotonically increasing $\big(d\varphi(a)/da = \varphi(a)\big(1 - \varphi(a)\big) > 0\big)$, we can make $d\zeta/dy > 0$ by forcing v_h and u_h to be positive. For instance, we can use the exponential transform, e.g., $v_h = e^{\tilde{v}_h}$, and train the network to estimate $\tilde{v}_h$. We will use $\boldsymbol{W}_F$ to denote the parameters in the FFNN part, i.e., $\boldsymbol{W}_F = \{v_0, \cup_h \{u_{h0}, u_h, v_h\}\}$ and use $\boldsymbol{W}$ to denote the parameter in the BDNN part, which will be described in Section 3.3.2.3.

3.3.2.3 Bayesian Deep Neural Network – Interval estimation of RUL

The BDNN connects observable degradation data $\boldsymbol{x}$, such as multiple sensor signals, operational variables, and cycle time, with the corresponding RUL y, which is estimated by $f^{\boldsymbol{W}}(\boldsymbol{x})$ in (3.1). A variety of RNN designs including LSTM and GRU can be used as the BDNN. One major drawback of the standard RNNs is the vanishing gradient problem; the error gradients vanish

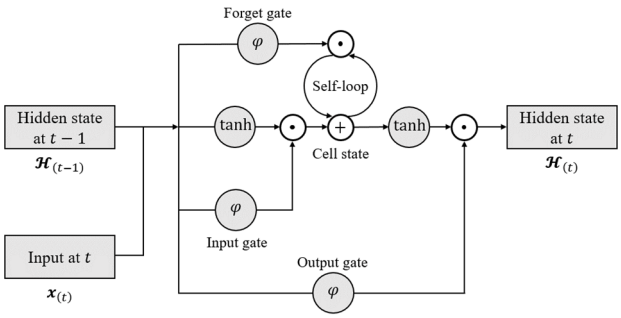


Figure 3.2 Diagram of an internal structure of a LSTM cell.

(become exceedingly close to 0) as they back-propagate through multiple time steps, which makes learning long-term relationships nearly impossible [77]. LSTM is designed to handle these long-term dependencies more efficiently than standard RNNs [78]. LSTM resembles a standard RNN except that each neuron in a hidden layer is replaced by a cell. Each cell has the inputs and outputs like a standard RNN, but also has gating systems and additional states. Without loss of generality, here we focus on the well-known LSTM model suggested in [79]. The extensions to other RNN variations are straightforward. In this LSTM design, a cell consists of a cell state, a hidden state, an input gate, an output gate, and a forget gate as shown in Figure 3.2. Henceforth, we drop the subscript n for notation simplicity and use $\boldsymbol{x}_{(t)} \in \mathbb{R}^{d\times 1}$ to refer to the degradation data in a sample measured at time step t, where d is the dimensionality of degradation data. The equations for the forward pass of the LSTM at time step t are as follows.

$$\begin{aligned}
\boldsymbol{\mathcal{F}}_{(t)} &= \boldsymbol{\varphi}\left(\boldsymbol{W}_{\mathcal{F}} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{U}_{\mathcal{F}} \cdot \boldsymbol{\mathcal{H}}_{(t-1)} + \boldsymbol{b}_{\mathcal{F}}\right) \\
\boldsymbol{\mathcal{I}}_{(t)} &= \boldsymbol{\varphi}\left(\boldsymbol{W}_{\mathcal{I}} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{U}_{\mathcal{I}} \cdot \boldsymbol{\mathcal{H}}_{(t-1)} + \boldsymbol{b}_{\mathcal{I}}\right) \\
\widetilde{\boldsymbol{\mathcal{C}}}_{(t)} &= \tanh\left(\boldsymbol{W}_{\mathcal{C}} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{U}_{\mathcal{C}} \cdot \boldsymbol{\mathcal{H}}_{(t-1)} + \boldsymbol{b}_{\mathcal{C}}\right) \\
\boldsymbol{\mathcal{C}}_{(t)} &= \boldsymbol{\mathcal{F}}_{(t)} \odot \boldsymbol{\mathcal{C}}_{(t-1)} + \boldsymbol{\mathcal{I}}_{(t)} \odot \widetilde{\boldsymbol{\mathcal{C}}}_{(t)} \\
\boldsymbol{\mathcal{O}}_{(t)} &= \boldsymbol{\varphi}\left(\boldsymbol{W}_{\mathcal{O}} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{U}_{\mathcal{O}} \cdot \boldsymbol{\mathcal{H}}_{(t-1)} + \boldsymbol{b}_{\mathcal{O}}\right) \\
\boldsymbol{\mathcal{H}}_{(t)} &= \boldsymbol{\mathcal{O}}_{(t)} \odot \tanh\left(\boldsymbol{\mathcal{C}}_{(t)}\right),
\end{aligned} \tag{3.4}$$

where the operator $\odot$ denotes element-wise multiplication, $\boldsymbol{\mathcal{F}}_{(t)}$ denotes a forget gate's activation vector at time step t, $\boldsymbol{\mathcal{H}}_{(t)}$ denotes a hidden state vector at time step t, $\boldsymbol{\mathcal{I}}_{(t)}$ denotes an input gate's activation vector at time step t, $\boldsymbol{\mathcal{O}}_{(t)}$ denotes an output gate's activation vector at time step t, $\boldsymbol{\mathcal{C}}_{(t)}$ is a cell state vector at time step t, and $\widetilde{\boldsymbol{\mathcal{C}}}_{(t)}$ is a candidate cell state vector at time step t. The input weight matrices, hidden weight matrices, and biases needed for all the components are denoted as $\boldsymbol{W}_{\bullet} \in \mathbb{R}^{H\times d}, \boldsymbol{U}_{\bullet} \in \mathbb{R}^{H\times H}$, and $\boldsymbol{b}_{\bullet} \in \mathbb{R}^{H\times 1}$, where the subscripts $\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{C}}$, and $\boldsymbol{\mathcal{O}}$ in (3.4) are for the forget gates, input gates, cell states, and output gates, respectively, and H is the dimensionality of the hidden state (i.e., the number of hidden neurons of the hidden layer). For simplicity, all hidden layers are considered to have the same number of hidden neurons. The initial values $\boldsymbol{\mathcal{C}}_{(0)}$ and $\boldsymbol{\mathcal{H}}_{(0)}$ are both set to 0. We can re-parametrized (3.4) as follows.

$$\begin{pmatrix} \boldsymbol{\mathcal{F}}_{(t)} \\ \boldsymbol{\mathcal{I}}_{(t)} \\ \boldsymbol{\mathcal{O}}_{(t)} \\ \widetilde{\boldsymbol{\mathcal{C}}}_{(t)} \end{pmatrix} = \begin{pmatrix} \varphi \\ \varphi \\ \varphi \\ \tanh \end{pmatrix} \begin{pmatrix} \boldsymbol{W}_{\mathcal{F}} & \boldsymbol{U}_{\mathcal{F}} & \boldsymbol{b}_{\mathcal{F}} \\ \boldsymbol{W}_{\mathcal{I}} & \boldsymbol{U}_{\mathcal{I}} & \boldsymbol{b}_{\mathcal{I}} \\ \boldsymbol{W}_{\mathcal{O}} & \boldsymbol{U}_{\mathcal{O}} & \boldsymbol{b}_{\mathcal{O}} \\ \boldsymbol{W}_{\mathcal{C}} & \boldsymbol{U}_{\mathcal{C}} & \boldsymbol{b}_{\mathcal{C}} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_{(t)} \\ \boldsymbol{\mathcal{H}}_{(t-1)} \\ 1 \end{pmatrix}$$

In this way, all the weights and biases needed for the forward pass of the LSTM can be expressed as a single matrix $\boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}_{\mathcal{F}} & \boldsymbol{U}_{\mathcal{F}} & \boldsymbol{b}_{\mathcal{F}} \\ \boldsymbol{W}_{\mathcal{I}} & \boldsymbol{U}_{\mathcal{I}} & \boldsymbol{b}_{\mathcal{I}} \\ \boldsymbol{W}_{\mathcal{O}} & \boldsymbol{U}_{\mathcal{O}} & \boldsymbol{b}_{\mathcal{O}} \\ \boldsymbol{W}_{\mathcal{C}} & \boldsymbol{U}_{\mathcal{C}} & \boldsymbol{b}_{\mathcal{C}} \end{pmatrix} \in \mathbb{R}^{4H\times(d+H+1)}$ as noted in (3.4).

Intuitively, at time step t, the input gate controls which new information will be used to update the cell state, the output gate controls which parts of the cell state we will output, and the forget gate decides which information will be removed from the previous cell state. Simultaneously, the cell state integrates the results from the previous cell state, forget gate, and input gate, and then keeps the useful information which will be used to predict the future. As a result, the gating mechanism in LSTM systematically regulates the flow of information through time and enables

the network to learn to keep, remove, and update the information. LSTMs can also be stacked in layers like other kinds of NNs. We will further examine how the number of hidden layers in LSTM affects the prognostic performance and training time in Section 3.4.4.1.

Next, we will explore how we model the weight uncertainty resulting from the unknown weight parameters $\boldsymbol{W}$. First, we place a prior distribution over $\boldsymbol{W}$ to perform Bayesian inference. Under the Bayesian framework, we recall the likelihood $y|f^{\boldsymbol{W}}(\boldsymbol{x}) \sim N(f^{\boldsymbol{W}}(\boldsymbol{x}), \sigma^2)$ from (3.1). Given the training sets $\boldsymbol{X}$ and $\boldsymbol{Y}$, we then look for the posterior distribution over $\boldsymbol{W}$ using Bayes' theorem: $p(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W})p(\boldsymbol{W})$. In BDNNs, the posterior $p(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y})$ is generally not tractable. Variational inference sidesteps this difficulty by defining an approximate variational distribution $q(\boldsymbol{W})$ which is computationally tractable, and minimize the Kullback–Leibler (KL) divergence between $q(\boldsymbol{W})$ and $p(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y})$ as:

$$\begin{aligned} KL\big(q(\boldsymbol{W})||p(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y})\big) &= \int q(\boldsymbol{W}) \log\left(\frac{q(\boldsymbol{W})}{p(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y})}\right) d\boldsymbol{W} \\ &\propto -\int q(\boldsymbol{W}) \log\big(p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W})\big)\, d\boldsymbol{W} + KL\big(q(\boldsymbol{W})||p(\boldsymbol{W})\big) \\ &= -\sum_{n=1}^{N} \int q(\boldsymbol{W}) \log\left(p\left(y_n \middle| f^{\boldsymbol{W}}(\boldsymbol{x}_n)\right)\right) d\boldsymbol{W} \\ &\quad + KL\big(q(\boldsymbol{W})||p(\boldsymbol{W})\big). \end{aligned} \tag{3.5}$$

Various researches have been conducted on BDNNs regarding the minimization of the KL divergence in (3.5). Here, we apply MC dropout approach which has shown to be easily implemented, highly scalable, and computationally efficient [68]. MC dropout showed that by choosing a specific form of an approximate distribution, the variational inference of a BDNN can be interpreted as performing one forward pass through the BDNN with dropout (referred to as a stochastic forward pass). In particular, the approximate distribution $q(\boldsymbol{W})$ is factorized over the

columns of the weight matrices, i.e., $q(\boldsymbol{W}) = \prod_{k=1}^{(d+H+1)} q(\boldsymbol{w}_k)$ where the kth column of $\boldsymbol{W}$ is denoted by $\boldsymbol{w}_k$. The approximate distribution for $\boldsymbol{w}_k$ is defined as a mixture of two Gaussians with small variances where one of the Gaussians has a mean zero:

$$q(\boldsymbol{w}_k) = pN(\boldsymbol{w}_k; \boldsymbol{0}, \sigma_v^2 \boldsymbol{I}) + (1-p)N(\boldsymbol{w}_k; \boldsymbol{\eta}_k, \sigma_v^2 \boldsymbol{I}).$$

Here, $\boldsymbol{\eta}_k$ is the variational parameter for the kth column of $\boldsymbol{W}$, p is the dropout probability, and σ_v^2 is a small variance. With this specific choice of the approximate distribution, one random sampling from the approximate posterior distribution is identical to one random output of the network where certain columns of $\boldsymbol{W}$ are randomly set to zero (dropped) with the probability p. This means that, at the test time, we can repeat multiple stochastic forward passes (for R repetitions) to obtain R empirical (Monte Carlo) samples from the approximate posterior distribution and compute the unbiased estimators of the mean and variance.

Specifically, in the context of our study, let y^* be the corresponding RUL of a test sample with a new input value $\boldsymbol{x}^*$. The approximate predictive distribution is given by $q(y^*|\boldsymbol{x}^*) = \int p(y^*|\boldsymbol{x}^*, \boldsymbol{W}) q(\boldsymbol{W}) \mathrm{d}\boldsymbol{W}$. The first two moments of $q(y^*|\boldsymbol{x}^*)$ can be empirically estimated using moment-matching. First, the predictive mean of y^* can be approximated by the sample mean as

$$\mathbb{E}_{q(y^*|\boldsymbol{x}^*)}(y^*) \approx \frac{1}{R}\sum_{r=1}^{R} p\left(y^*\middle|\boldsymbol{x}^*, \widehat{\boldsymbol{W}}_r\right) = \frac{1}{R}\sum_{r=1}^{R} f^{\widehat{\boldsymbol{W}}_r}(\boldsymbol{x}^*), \tag{3.6}$$

with R random outputs through stochastic forward passes $f^{\widehat{\boldsymbol{W}}_r}(\boldsymbol{x}^*)$, where $\widehat{\boldsymbol{W}}_r \sim q(\boldsymbol{W})$. In a similar way, we can obtain the unbiased estimator of the variance of the variational predictive distribution as:

$$Var_{q(y^*|\boldsymbol{x}^*)}(y^*) \approx \hat{\sigma}^2 + \frac{1}{R}\sum_{r=1}^{R}\left(f^{\widehat{\boldsymbol{W}}_r}(\boldsymbol{x}^*)\right)^2 - \left(\frac{\sum_{r=1}^{R} f^{\widehat{\boldsymbol{W}}_r}(\boldsymbol{x}^*)}{R}\right)^2, \tag{3.7}$$

which is equal to the degradation uncertainty $\hat{\sigma}^2$ as estimated in Section 3.3.2.2 plus the weight uncertainty represented as the sample variance of R stochastic forward passes. The detailed proof and the theoretical justifciations of the above estimations can be found in [68].

3.3.3 Sliding time window

As mentioned in Section 3.1, one common challenge in degradation applications is that the amount of degradation data is often limited due to low signal sampling frequency or high cost for data acquisition. One way to alleviate this issue is by using data augmentation. Data augmentation aims at increasing the amount of training data by transforming the existing samples to create new samples [80]. In this study, instead of creating one training sample from one training system, we can generate multiple training samples from one training system by sliding a time window of fixed width n_{TW} over the whole degradation data. While one key challenge for data augmentation is how to assign correct labels for the new samples, in the proposed framework, we can straightforwardly obtain the new labels by subtracting the signal measurement time from the recorded failure time (also called a linear RUL function) or applying a piece-wise linear RUL function [56], [60]. Figure 3.3 illustrates how sliding a time window of width n_{TW} generates multiple training samples from a system with d degradation data and assigns the corresponding target values (RULs) based on the linear RUL function. For any system whose lifetime is shorter

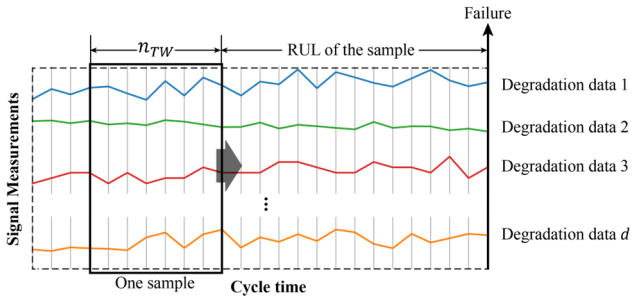


Figure 3.3 Sliding time window of width n_{TW} where there are d degradation data collected from each system.

than n_{TW}, we can create one sample using its full lifetime. The suitable width of the time window n_{TW} strongly depends on the characteristics of degrading systems, e.g., signal sampling frequency or the average lifetime. For instance, in applications where each system has a long average lifetime and shows a unique pattern appearing throughout the long-term period, large n_{TW} can be applied to learn long-term dependencies. We can either adopt domain knowledge or investigate historical degradation signals to decide the appropriate value of n_{TW}. In general, while a larger n_{TW} conveys more information to the model, it will slow down the model training. The effects of n_{TW} to prognostic performance and model training time will be further investigated in Section 3.4.4.1.

3.4 Numerical study

We applied the proposed method to two degradation datasets: turbofan aircraft engines and Li-ion batteries. For brevity, in this section, we focus on degradation processes of turbofan aircraft engines, but refer readers to Appendix for the results using the Li-ion battery dataset. In Section 3.4.1, we provide the overview of the system and dataset. Section 3.4.2 demonstrates how we preprocess the raw degradation data and design the inputs of the proposed framework. Section 3.4.3 introduces two evaluation metrics used to assess prognostic performance. In Section 3.4.4, the effects of hyper-parameters are investigated, and the prognostic results of the proposed method and existing benchmark approaches are compared. Additional numerical results analyzing the sensitivity and computational costs of the proposed method are also presented in Appendix.

3.4.1 Overview of the system and dataset

The multi-sensor signals are generated from C-MAPSS, a commercial software widely used to simulate degradation processes of turbofan aircraft engines [81]. The aircraft engine is a representative example of modern complex engineering systems, as various types of sensors

Table 3.1 Overview of C-MAPSS dataset.

	Sub-Dataset			
	FD001	FD002	FD003	FD004
# of Training Systems	100	260	100	248
# of Testing Systems	100	259	100	248
# of Environmental Conditions	1	6	1	6
# of Failure Modes	1	1	2	2

monitor the engine simultaneously and each engine may have multiple failure modes and operate under different operational conditions over time. The degradation in engine performance is due to wear and tear according to the usage pattern. Each system (engine) starts with different degrees of initial wear and manufacturing variations that are unknown. The dataset includes four sub-datasets denoted as FD001, FD002, FD003, and FD004. Each sub-dataset contains one training set and one test set. The overview of four sub-datasets is summarized in Table 3.1. Each system collects 21 sensor signals and 3 operational variables at each cycle time. Table 2.2 gives detailed descriptions of these sensor signals and operational variables. The signals for each training system are collected until failure, whereas the signals for each testing system are truncated at some random point prior to failure. Figure 3.4 shows normalized signal measurements of representative sensors collected from Training system 1 in each sub-dataset. From the figure, we can see that even the same types of sensors show significantly different degradation patterns and trends depending on the failure modes and operational conditions. Compared to the degradation signals collected from the system with a single failure mode and a single operational condition (FD001; the first row of plots in Figure 3.4), those from the system with multiple failure modes and multiple operational conditions (FD004; the last row of plots in Figure 3.4) show less clear degradation trend with larger noise. The failure time of all training systems and the actual RULs of all testing systems are also recorded. The goal is to accurately estimate the RUL of each testing system. A comprehensive evaluation of

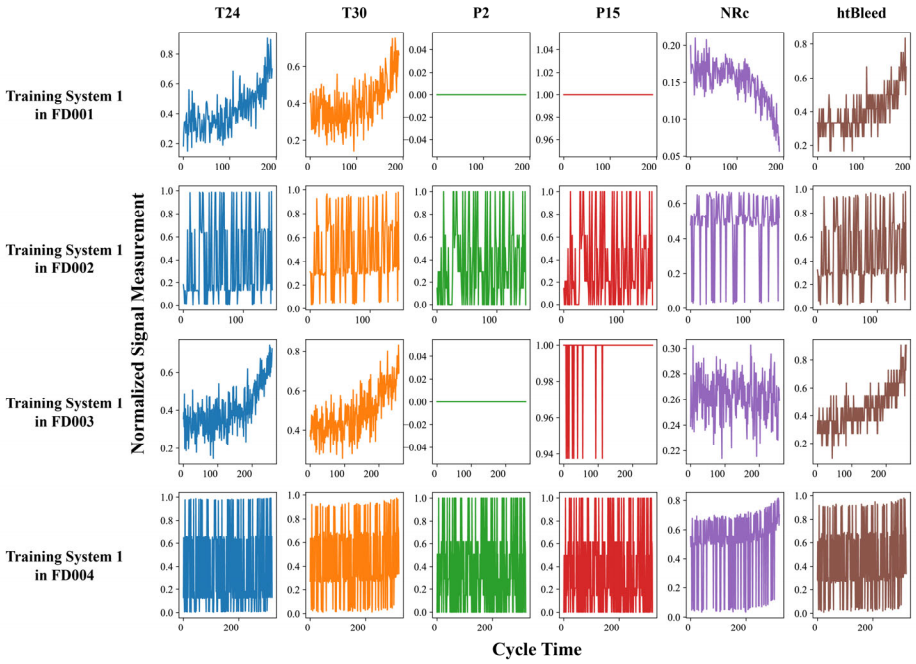


Figure 3.4 Normalized signal measurements of representative sensors collected from Training system 1 in each sub-dataset.

the proposed method is carried out on all four sub-datasets to investigate how the proposed method performs for such complex systems.

3.4.2 Data preprocessing

To provide end-to-end solutions, we use all sensor signals, operational variables, and cycle time as inputs, without any sensor selection or feature extraction procedures. We apply min-max normalization such that each degradation data is within the range of $[0, 1]$. Then, the sliding time window procedure is applied to the training systems to augment the training dataset. Following the existing studies using the same dataset [56], [60], we use a linear RUL function to label the RUL value of each training sample. As a result, one training sample contains all 21 sensor signals, 3 operational variables, and cycle time measured at the most recent n_{TW} time steps from a certain system. The resulted $25 \times n_{TW}$ input matrixes are fed into the BDNN as one training sample.

3.4.3 Evaluation metrics

In assessing the prediction error, we use two evaluation criteria: scoring function and root-mean-square error (RMSE). The following scoring function has been widely adopted in many studies using the same dataset [56], [60], [61]:

$$s = \sum_{m=1}^{M} s_m, \quad s_m = \begin{cases} e^{-\frac{\delta_m}{13}} - 1, & \text{if } \delta_m < 0, \\ e^{\frac{\delta_m}{10}} - 1, & \text{if } \delta_m \geq 0, \end{cases} \tag{3.8}$$

where s is the computed total score, s_m is the score of testing system m, M is the number of testing systems, and $\delta_m = (\text{Estimated RUL of testing system } m - \text{True RUL of testing system } m)$. In addition to the score, $\text{RMSE} = \sqrt{\frac{\sum_{m=1}^{M} \delta_m^2}{M}}$ is also selected for evaluation. Note that a smaller score and a smaller RMSE correspond to smaller errors between the true and estimated values of the RULs, hence better prognostic performance. The above scoring function is an asymmetrical function that penalizes late predictions ($\delta_m > 0$) more than early predictions ($\delta_m < 0$). As a comparison, RMSE equally penalizes both early and late predictions, and thus it provides fair comparisons especially when there are methods which attempt to reduce the score by underestimating RULs and favoring early predictions.

3.4.4 Results and comparison

3.4.4.1 Effects of hyper-parameters

In this subsection, the effects of two main hyper-parameters in the proposed framework are investigated based on the sub-dataset FD001: the number of hidden layers of BDNN (Bayesian LSTM) and the time window width n_{TW}. As explained in Section 3.3.2.2, since both the inputs and outputs of the FFNN part are one-dimensional real values, a shallow architecture is used to

save the computational cost. Furthermore, we observe that the FFNN was generally robust to small changes in the number of hidden neurons. Thus, the number of hidden layers and the number of hidden neurons of the FFNN part are fixed to 1 and 5, respectively.

To clearly illustrate the effects of the number of hidden layers of BDNN and n_{TW}, the value of one parameter is fixed while another parameter varies. In particular, the number of hidden layers of BDNN is set to 2 when the value of n_{TW} changes. In contrast, n_{TW} is set to 40 when the number of hidden layers of BDNN varies. Then, the number of hidden neurons per hidden layer of BDNN and the tuning parameter λ are optimized via 10-fold cross validation. We train the model using RMS-Prop with a learning rate of 0.001, dropout probability of 0.2, and mini-batch size 200. RMS-prop is an iterative optimization algorithm in which the learning rate is adaptively scaled for each dimension according to the accumulated squared gradient at each iteration [82]. As an aside, it is worth noting that different values of the learning rate, dropout probability, and mini-batch size were used; however, we found that small changes in these parameters did not have a significant effect on the prognostic performance, and thus the above values are used to train the model. For each trial, $R = 100$ realizations of stochastic forward passes are conducted at the test time. The simulations are repeated 20 times for each value of the number of hidden layers of BDNN or n_{TW}. Every trial was executed with two Intel(R) Xeon(R) CPU E5-4620 0 2.20GHz processors and 192 GB RAM. Figure 3.5 illustrates the score of the testing dataset and model training time according to the different number of hidden layers in the BDNN when n_{TW} is fixed to 40. The results of the RMSE present very similar trends to the results of the score, and thus are omitted. We can see that it takes a longer time to train the model with more hidden layers. Two hidden layers of BDNN lead to the lowest score and achieve the best prognostic performance. Although deeper networks can capture more complex patterns in general, they are more likely to overfit the training set due

to involving a large number of parameters. Figure 3.6 describes how n_{TW} affects the score and model training time when the BDNN has two hidden layers. Again, the results of the RMSE show very similar patterns to the results of the score, and thus are omitted. We can see that the score generally decreases, and the model training time increases as more information is included in one training sample, i.e., a larger n_{TW}. Interestingly, after n_{TW} exceeds 25, further increases in the value of n_{TW} provide only marginal improvements in the score but suffer increased training time. One possible reason is that even when n_{TW} is larger than 25, the proposed network still automatically focuses on the data collected at the most recent 25 time steps since those signals present more clear degradation patterns and provide enough information to make the accurate RUL prediction.

3.4.4.2 Comparison with existing methods

In this subsection, the prognostic performance of the proposed method is compared with the results of the following benchmark methods:

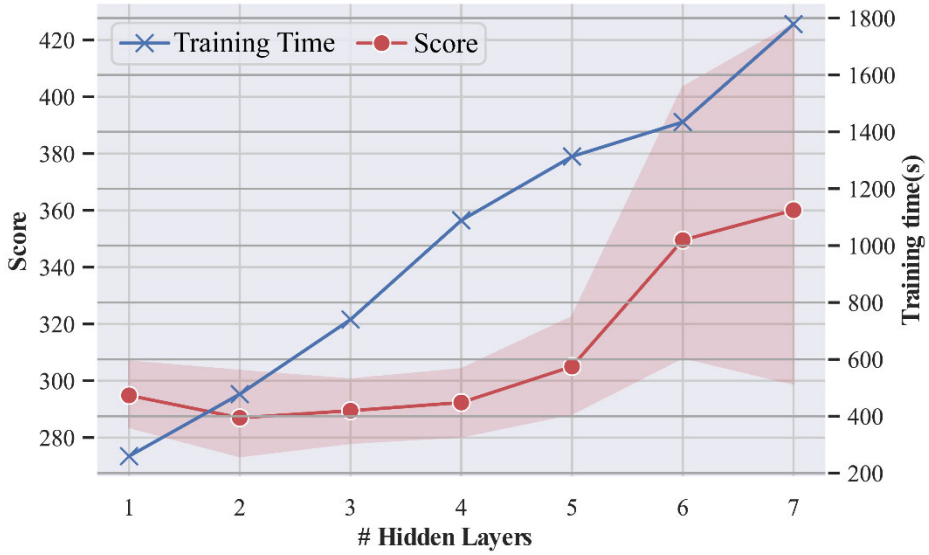


Figure 3.5 Average score (circle-marked line), one standard deviation of the score (shaded area), and average training time (X-marked line) for different numbers of hidden layers in the Bayesian LSTM where n_{TW} is fixed to 40.

(1) The proposed network without the monotonic constraint for the degradation uncertainty
(2) Conventional deep LSTM directly taking degradation data as inputs and RULs as outputs and using the sum of squared error as a loss
(3) Deep CNN based regression approach (DCNN) [83]
(4) Multiobjective deep belief networks ensemble (MODBNE) [60]
(5) Generic health index (HI) approach [84]

Comparison with Benchmark (1) will show how the consideration of a unique characteristic of degradation uncertainties improves the prognostic performance. Benchmark (2) adopts well-known deep LSTM structures designed for general sequential data. By comparing the proposed method with Benchmark (2), we can highlight the benefit of tailoring the network to address the distinctive characteristics of degradation modeling and prognostics. Benchmark (3) DCNN and Benchmark (4) MODBNE are used as the benchmark methods since they are advanced deep networks in the existing literature explicitly designed for degradation modeling and prognostics. These models have been shown to outperform a variety of conventional machine learning

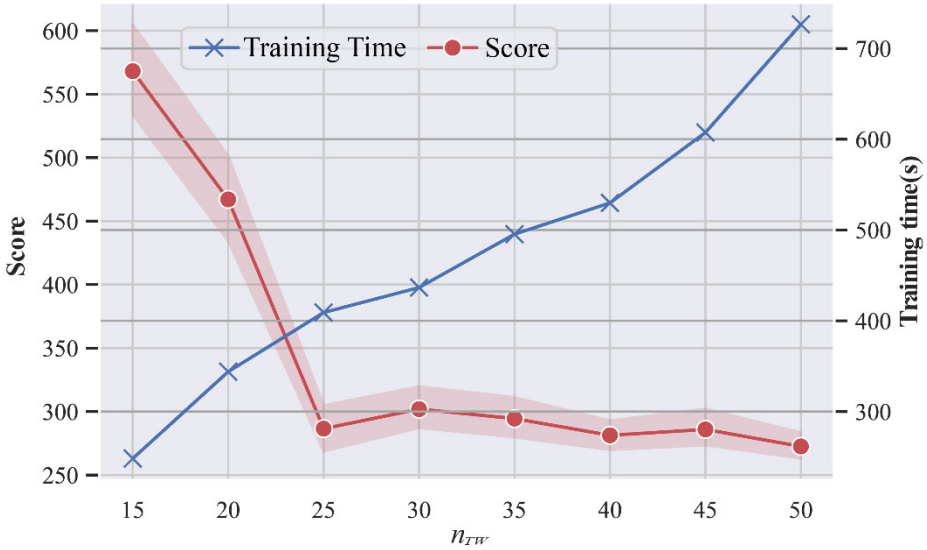


Figure 3.6 Average score (circle-marked line), one standard deviation of the score (shaded area), and average training time (X-marked line) for different values of n_{TW} where the number of hidden layers in the Bayesian LSTM is fixed to 2.

approaches including multilayer perceptron (MLP), extreme learning machine (ELM), SVM, extra tree regressor, random forest, and gradient boosting regarding RUL prediction. However, these methods do not provide interval estimations of RULs, and thus the generic HI approach (5) is used as another benchmark. The generic HI approach develops a latent linear model to select informative sensors and construct one-dimensional health index based on the selected sensors. The constructed health index is used to recover the underlying degradation process and to quantify the uncertainties in RUL estimations. The HI approach achieved higher prognostic accuracy over several other existing data-driven prognostic approaches using the C-MAPSS dataset.

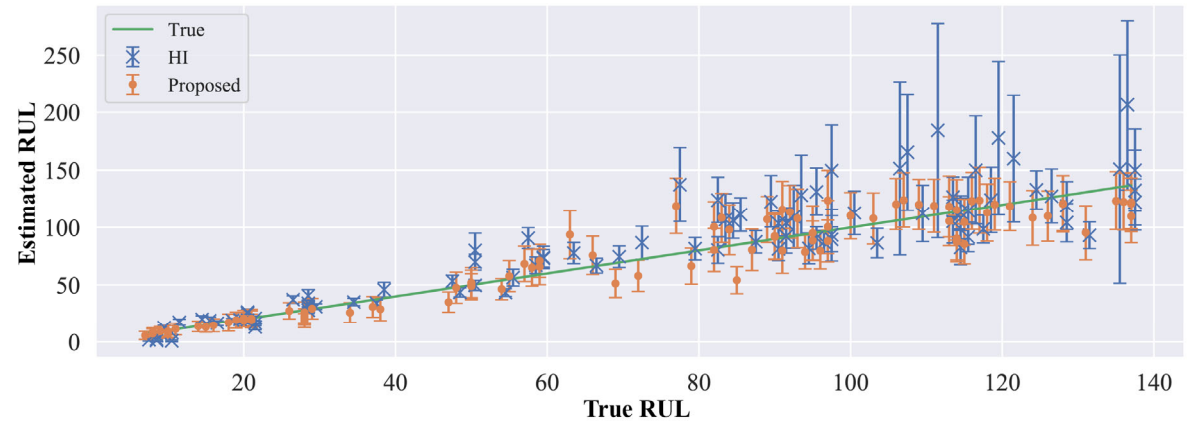
The proposed method, Benchmarks (1), and (2) apply the sliding time window preprocessing with $n_{TW} = 25$. In LSTM-based methods (including the proposed method, Benchmarks (1), and (2)), the number of hidden layers of LSTM is set to 2 and the number of hidden neurons per hidden layer of LSTM is again optimized via 10-fold cross validation. Other detailed training settings follow Section 3.4.4.1. The mean and standard deviation of the prognostic performance of each method on each of the four sub-datasets are summarized in Table 3.2. Each of the proposed method, Benchmarks (1), (2), and (4) are repeated for 10 times to examine the robustness of the methods and to obtain the average performance. Note that the standard deviations of the prognostic results of Benchmarks (3) and (4) were not provided in the existing studies. The lowest mean and standard deviation of score and RMSE for each sub-dataset are highlighted in boldface. We can see that the proposed Bayesian DL framework outperforms the benchmark methods by achieving the lowest score and RMSE, including when there are multiple operational conditions or multiple failure modes (FD002, FD003, and FD004). By imposing the monotonic constraint on degradation uncertainties, the proposed method improves the prognostic performance compared to Benchmark

Table 3.2 The mean and standard deviation (in parentheses) of prognostic results of all methods on each of four sub-dataset (the best performance is highlighted in bold).

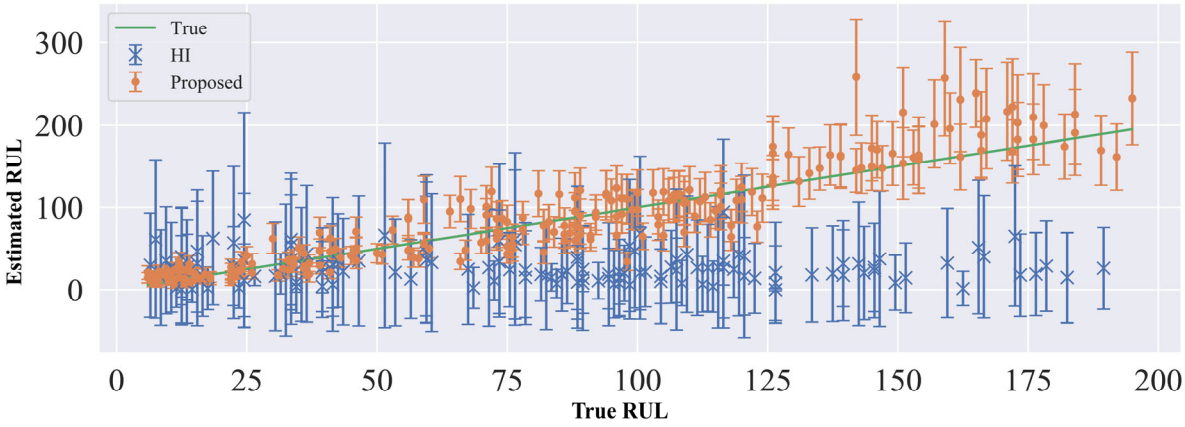
		Proposed	(1) Propose without monotonic constraint	(2) Deep LSTM	(3) DCNN	(4) MODBNE	(5) HI
FD001	Score	267.21 (14.78)	301.19 (14.18)	356.47 (21.73)	1286.7	334.23	473.05
	RMSE	12.19 (0.22)	13.59 (0.36)	15.11 (1.42)	18.45	15.04	15.18
FD002	Score	2007.81 (87.83)	2039.38 (101.86)	34937.80 (3351.41)	13570	5585.34	5631267
	RMSE	18.49 (0.34)	19.65 (1.56)	40.98 (5.85)	30.29	25.05	63.86
FD003	Score	409.39 (14.02)	534.25 (21.74)	2095.45 (244.64)	1596.2	421.91	905.93
	RMSE	12.07 (0.18)	14.68 (1.38)	21.98 (3.07)	19.82	12.51	14.76
FD004	Score	2415.71 (134.12)	2950.36 (162.47)	20986.27 (3973.22)	7886.4	6557.62	12798220
	RMSE	19.41 (1.61)	20.27 (1.84)	29.78 (5.31)	29.16	28.66	75.78

(1). For systems with a single failure mode and a single operational condition (FD001), Benchmark (2) (the deep LSTM for general purposes) shows comparable performance with other benchmark methods. However, when there are multiple failure modes or multiple operational conditions (FD002, FD003, and FD004), Benchmark (2) shows significantly higher RMSE and score with larger standard deviations than the other DL-based methods. This indicates that although the conventional DL architecture developed for general purposes might be used for prognostics of simple systems, consideration of the unique characteristics of degradation processes are crucial to achieve satisfactory prognostic performance in complex systems.

Benchmark (5) (the HI approach) shows the comparatively poor performance, especially when there are multiple operational conditions. This is because while Benchmark (5) models degradation processes as the mixed effect models with a fixed set of basis functions, it is possible that each sensor signal shows considerably different degradation forms depending on the operational conditions. Nevertheless, Benchmark (5) offers better interpretability and practicality than the



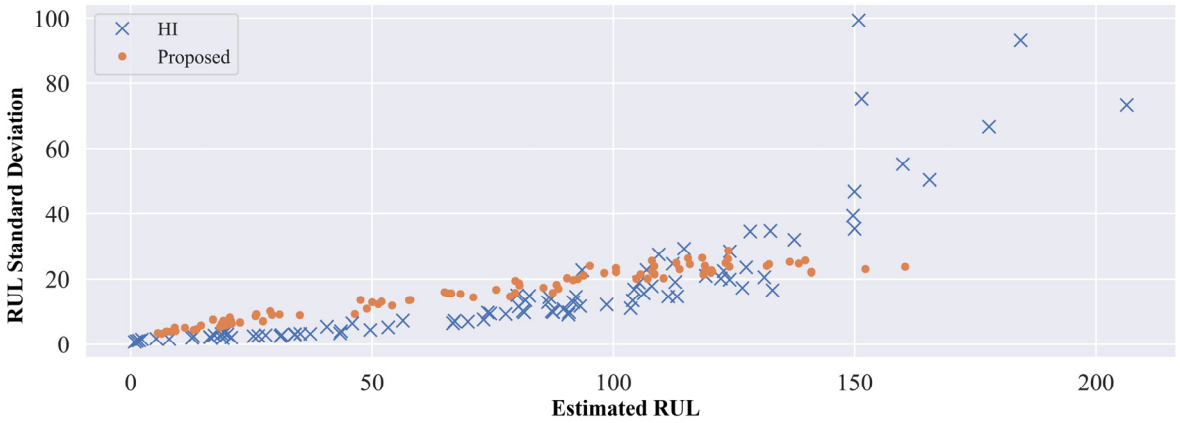
(a)



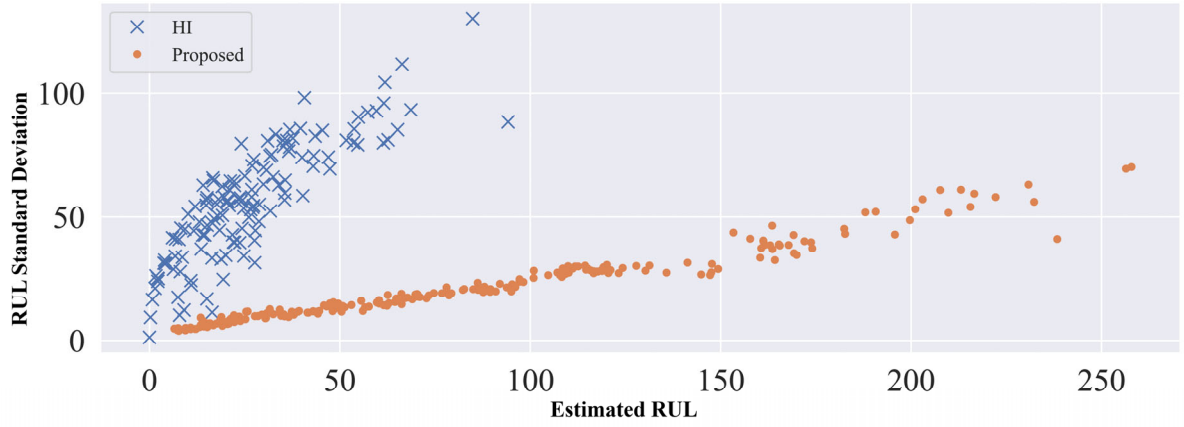
(b)

Figure 3.7 Interval estimations of RULs of testing systems in (a) FD001 and in (b) FD004 by using the proposed network and Benchmark (5) HI approach. The solid line represents the actual RULs of testing systems. The solid line represents the actual RULs of testing systems. The X markers show the mean plus/minus one standard deviation of RULs estimated using the HI approach. The circle markers show the mean plus/minus one standard deviation of RULs estimated using the proposed method.

existing DL-based prognostic methods since it provides the distributions of estimated RULs instead of the point estimations. Figure 3.7 illustrates the estimated mean and uncertainties on the RULs of testing systems in (a) FD001 and in (b) FD004 by using the proposed method and Benchmark (5). The X markers show the mean plus/minus one standard deviation of RULs estimated using Benchmark (5). The circle markers show the mean plus/minus one standard deviation of RULs estimated using the proposed method. The interval estimations using



(a)



(b)

Figure 3.8 Standard deviations of estimated RULs of testing systems in (a) FD001 and in (b) FD004 according to the estimated values of RULs by using the proposed network and Benchmark (5) HI approach.

Benchmark (5) are slightly shifted along the x-axis for better visualization. Note that Benchmark (5) produces invalid estimations (RUL<0) for some testing systems in FD004 and these estimations are excluded from the figure. Figure 3.7 (a) shows that the proposed method provides more accurate and tighter interval estimations of RULs when there is a single failure mode and a single operational condition (FD001). In Figure 3.7 (b), when there are multiple failure modes and multiple operational conditions (FD004), Benchmark (5) greatly underestimates the RULs, whereas the proposed method gives RUL estimations closer to the true values.

Figure 3.8 shows a closer look at Figure 3.7 by focusing on the standard deviations of estimated RULs according to the corresponding mean values. Figure 3.8 (a) shows that for both methods in FD001, the prediction uncertainties are similar and overall increase as the estimated values of RULs increase for the testing systems. When there are multiple failure modes and multiple operational conditions (FD004), Figure 3.8 (b) shows that the proposed method controls similar prediction uncertainties to the similar estimated RULs. However, the standard deviations are significantly inflated with respect to the similar estimated RULs in Benchmark (5). From Figure 3.7 (b) and Figure 3.8 (b), we can see that the proposed method handles multiple failure modes and multiple operational conditions much better than Benchmark (5) by providing more accurate RUL estimations and well-controlled degradation uncertainties.

Next, we explore how the prognostic performance of the proposed method changes with the actual RUL values in Figure 3.9. The figure demonstrates that the proposed method in general yields smaller absolute errors ($|\delta_m|$) and smaller score (s_m) for the testing systems with smaller RULs. This indicates that the proposed method provides more accurate RUL predictions as a system approaches the end of life, which is important for practical applications.

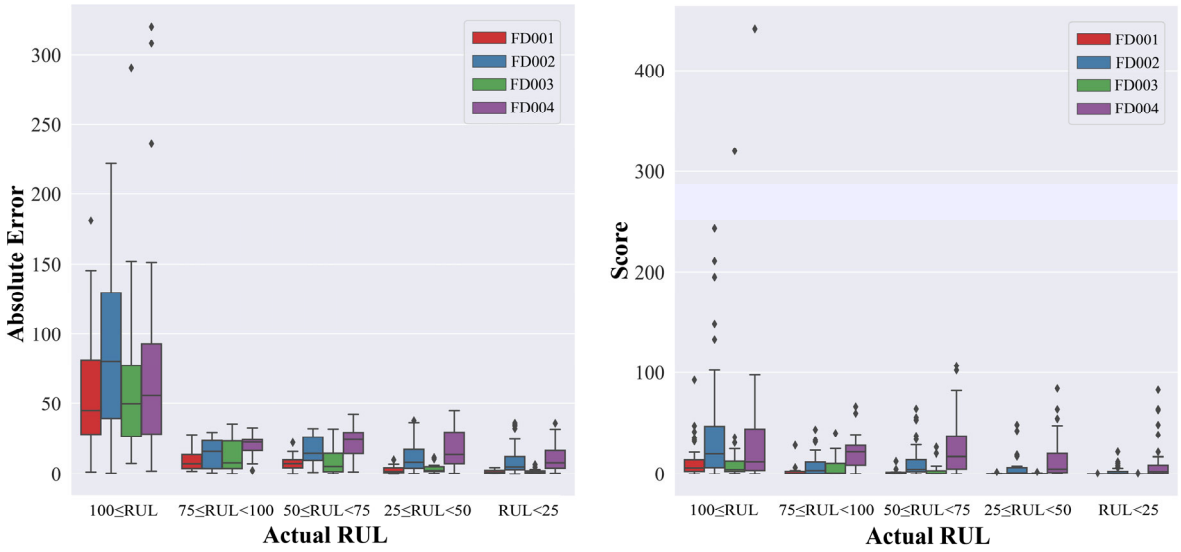


Figure 3.9 Prognostic performance of the proposed method on each of four sub-datasets according to the actual RULs.

3.5 Conclusion

In this study, we proposed a novel Bayesian DL framework that performs degradation modeling and prognostics in complex systems from a probabilistic point of view. One major obstacle that has prevented the use of DL in many degradation applications is that the existing DL-based prognostics remain mostly black boxes providing only point estimations of RULs and are not tailored to the distinctive characteristics of degradation modeling and prognostics. To overcome these issues, the proposed framework enhances the probabilistic interpretability by providing not only the final estimated values of RULs, but also the uncertainty quantifications. In particular, the proposed framework systematically models two types of uncertainties embedded in degradation modeling and prognostics: weight uncertainty and degradation uncertainty, by incorporating general characteristics of degradation processes. Furthermore, the proposed Bayesian DL framework does not assume any particular type of degradation processes nor the availability of domain-specific prior knowledge such as a failure threshold, and thus it can be widely applied to

various complex systems with multiple sensor signals, multiple failure modes, and multiple operational conditions. The extensive numerical studies on the degradation of aircraft engines and Li-ion batteries showed that the proposed method outperformed the existing benchmark methods by achieving higher accuracy and providing well-controlled uncertainty quantifications in complex systems.

There are several potential topics for future work. First, although we combine sensor signals and operational variables as one input data, it would be interesting to explore more systematic approaches to model sensor signals and operational variables separately. Second, it is worth extending the proposed model to adopt the available domain-specific knowledge related to the underlying degradation processes, e.g., spatial locations of sensors, and further improve the interpretability. Third, in degradation applications, it is common to have missing or asynchronous sensor measurements. Further studies are needed to investigate how to efficiently handle those issues in the proposed framework. Lastly, there can be various applications where historical training dataset with abnormal health condition is unavailable. It would be interesting to address this challenge especially from an unsupervised learning point of view.

3.6 Appendix

3.6.1 Comparison of computational costs

Table 3.3 shows the average computation time of the proposed method and other benchmark methods over 10 trials on FD001 sub-dataset. The proposed method, Benchmarks (2) and (5) are tested using Intel Core i5-6300U CPU 2.40-GHz and 16-GP RAM. The Benchmark (4) is tested with Intel Core i7-3770 3.40-GHz CPU and 16-GB RAM. The training time of Benchmark (3) is excluded as the authors of the paper did not provide detailed settings to reproduce the results and calculate the computational costs. The table shows that the proposed method yields much lower training time than the existing deep learning approach (MODBNE) and comparable training time to Deep LSTM (Benchmark (2)). Although the proposed method takes longer training time than the existing parametric approach (HI), please note that this training procedure is carried out offline. In online, we can obtain the interval estimations of RUL which take around 0.1s seconds using two Intel(R) Xeon(R) CPU E5-4620 0 2.20GHz processors and 192 GB RAM and around 0.5s using Intel Core i5-6300U CPU 2.40-GHz and 16-GP RAM. To obtain the interval estimation, we repeat multiple stochastic forward passes (for R repetitions) to obtain R empirical (Monte Carlo) samples. Owing to these samples being independent, computation time can be further reduced by using parallel computing if needed in practice.

Table 3.3 Average model training time of the proposed method and other benchmark approaches on FD001.

	The Proposed Method	Deep LSTM (Benchmark(2))	MODBNE (Benchmark(4))	HI (Benchmark(5))
Average Model Training Time (s)	1014.75	961.47	1153760.67	0.85

3.6.2 Numerical study – Li-ion battery

In this section, we further apply the proposed method to predict the RULs of Li-ion batteries. In Section 3.6.2.1, we provide an overview of the system and dataset. Section 3.6.2.2 demonstrates how we preprocess the raw degradation data and presents the prognostic results of the proposed method.

3.6.2.1 Overview of the system and dataset

In this dataset [85], a set of four 18650 Li-ion batteries (RW1, RW2, RW7 and RW8) were continuously operated under a randomly generated sequence of charging and discharging profiles (also referred to as random walk discharging). Each system (battery) starts from a fully charged cell in a stationary condition. At each cycle time, three sensor signals are collected from a battery: voltage, current and temperature. Here, we consider the last signal measurement time of each battery as its failure time. More detailed experimental settings can be found in [85].

3.6.2.2 Data preprocessing and results

We first conduct similar data preprocessing procedures to 3.4.2. Specifically, min-max normalization is applied to all three sensor signals, such that each degradation data is within the range of [0, 1]. Then, the sliding time window procedure is applied to the training systems to augment the training dataset. The hyper-parameters are optimized via 10-fold cross validation: $n_{TW} = 10$, the number of hidden layers of BDNN is set to 2, the number of hidden neurons per hidden layer of BDNN is set to 20.

Out of four batteries, the first three batteries (RW1, RW2 and RW7) are used to train the model while the fourth battery (RW8) is used to test the model. The RUL prediction results of RW8 are

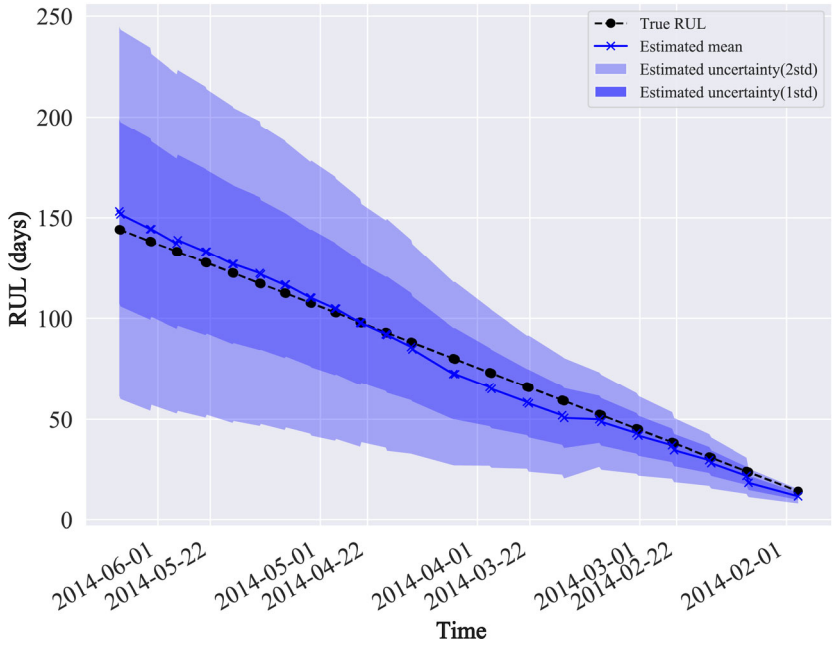


Figure 3.10 Estimated RULs of RW8. The black dashed line represents the actual RULs of RW8. The X marker line shows the mean of estimated RULs using the proposed method. The shaded areas show the one and two standard deviations of the estimated RULs.

illustrated in Figure 3.10. We can see that as the battery approaches the failure, the proposed method provides more accurate and tighter interval estimations of RULs.

Chapter 4 Individualized Degradation Modeling and Prognostics in a Heterogeneous Group via Incorporating Intrinsic Covariate Information

4.1 Introduction

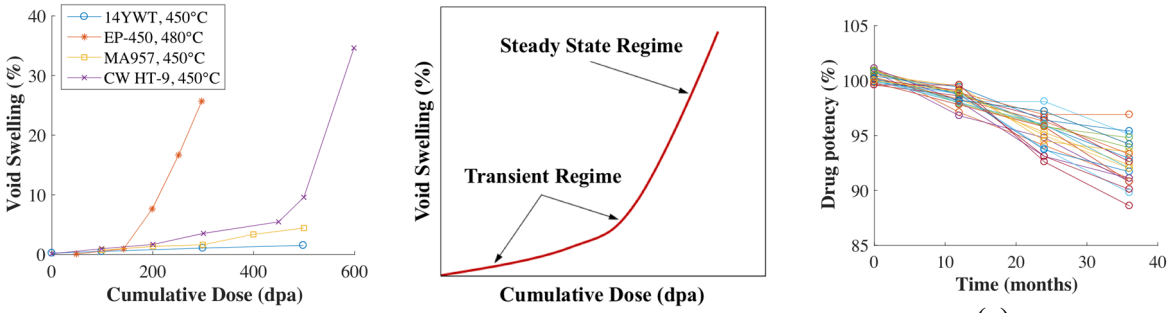
Degradation modeling and prognostics have been widely applied in practice to accurately monitor and predict the degradation process of a unit (e.g., a system, equipment, and a patient) based on the collected sensor signals. For instance, in many manufacturing applications, it is crucial to properly infer the underlying degradation status of a system and predict its RUL to minimize the economic losses and safety risks from the unexpected failure. In healthcare, accurate modeling and predicting the progression of Alzheimer's disease, which is a neurodegenerative disorder, can assist in early disease diagnosis and efficient treatment strategy, and reduce unnecessary medical costs for the patients [31], [86].

In the past, many approaches have assumed all units in a group are identical and constructed a group-level degradation model that all units share [7]. These methods often model the distribution of the failure time for the entire group using time-based parametric distributions (e.g., Weibull) and ignore the variations among individuals, and thus leads to significant prognostic errors. To overcome these shortcomings, recent approaches have applied mixed-effect models for the degradation signals, where random-effect parameters are used to characterize unit-to-unit variations [84], [87]. In particular, these methods update the posterior distribution of the random-effect parameters of each unit separately once the new degradation signals are collected from the unit. Consequently, when the degradation signals of the units of interest are not sufficient or

unavailable, these methods are either inapplicable or can only provide predictions according to the group-level behavior, which leads to large prediction uncertainties.

In many applications, whereas all units in a group share some similarities, each unit has its own distinct individual-level characteristics, such as the manufacturing design information of engineering systems and the risk factors of patients, which we refer to as *covariates* hereafter. The covariates of a unit can provide valuable information about the degradation process along with the degradation signals. A group of units often degrade according to a similar trend, but follow different paths depending on these covariates. In this study, we assume the existence of the relation between the degradation path and the covariates and thus different units with different covariates have different degradation paths. Furthermore, we mainly focus on the cases where the potential covariates that may affect the degradation process of a unit are known from experiments or domain experts, yet the exact relation is unknown. Below are some specific application examples:

- Void swelling: Void swelling is a nuclear-specific material degradation mechanism that causes an increase in volume of components exposed to high-energy neutrons at higher temperatures [32]. Various swelling processes of ferritic-martensitic and ferritic-ODS alloys are shown in Figure 4.1 (a) [88]. The general trends in swelling as a function of dose for steels are shown in Figure 4.1 (b). Void swelling is affected by many covariates such as alloy composition and



(a) (b) (c)

Figure 4.1 Practical examples of degradation processes of various heterogeneous groups: (a) and (b) Void swelling, and (c) Drug potency.

material structure. How to effectively model the complex relations between these covariates and degradation process, predict the future evolution and extend the degradation period is the key to mitigating the effect of swelling and ensuring safe operation.

- Drug potency: Development of a new drug involves performing a stability study to determine the drug's shelf life (lifetime). As the potency of a drug degrades over time, its shelf life is defined as the time interval that the drug potency (e.g., strength) will remain within the approved specifications after manufacture. Figure 4.1 (c) illustrates the drug potency degradation processes of 24 batches of a drug product over a 36-month period [89]. It is crucial to understand how different characteristics of a drug (e.g., solubility, melting point, formulation) affect degradation processes of drug potency to ensure the quality and safety of the drug.
- Progression of the Alzheimer's disease: The declination of cognitive status of the Alzheimer's disease patients can be also viewed as one example of degradation processes of a heterogeneous group. It possesses commonalities resulting from the progression of the same disease, while each patient has unique covariates, e.g., the demographic, genetic, and imaging information. Figure 4.9 shows examples of how cognitive status of the Alzheimer's disease patients change. We will discuss this dataset in more detail in Section 4.5.1. As mentioned earlier, accurate modeling and predicting the progression of the disease is crucial for early disease diagnosis and efficient treatment strategy, and reducing unnecessary medical costs.

In this study, the term "covariates" refers to the *static intrinsic* characteristics of units (e.g., genetic information of patients). Thus, the term "covariates" in this study is distinguished with some existing studies especially on accelerated degradation tests, where "covariates" often refers to dynamic *external* factors such as environmental stresses [90]–[92]. Dynamic external covariates

are often modeled to accelerate or decelerate the degradation process, while intrinsic covariates in the proposed study are modeled to inherently characterize the degradation process itself. For instance, Hong *et al*. [93] used the cumulative exposure model to describe the effect of dynamic covariates on the parametric failure time distribution, which assumes the dynamic environmental covariates can increase or decrease the degradation rate. The main difference is that intrinsic static covariates do not change over time, while external covariates do. For example, in void swelling application, the alloy composition of a unit is its basic nature and does not change over time. On the other hand, the external conditions such as temperature also affect the degradation process of the unit, yet it can change over time.

Another widely used approach that incorporates covariates for degradation modeling is through the Cox proportional hazards model (PHM) [28]. Specifically, the PHM assumes that the hazard rate $\lambda(t|x)$ of a unit at a given time t consists of two multiplicative factors, a baseline hazard function $\lambda_0(t)$ and an exponential term $\exp(\beta x)$ based on covariates x. Then, the PHM measures the impact of covariates on the hazard by $\lambda(t|x) = \lambda_0(t)\exp(\beta x)$. Unlike the proposed method where the covariates are static intrinsic characteristics affecting degradation signals, existing PHM-based prognostic methods often directly use observed degradation signals or their features as covariates [29], [30]. For instance, in Liao *et al*. [30], the features extracted from vibration signals (e.g., kurtosis) of bearings are used as covariates. This is certainly different from the intrinsic covariates defined in this study. In the proposed model, the intrinsic covariates are the static characteristics of bearings such as bearing type or boundary dimensions while vibration signals are dynamically changing over time. Various extensions of the PHM have been studied in the literature [92], [94], [95]; however, such approaches still carry out modeling and prognostics

of each in-service unit separately and fail to fully leverage the information from other in-service units and historical units.

To fill this literature gap, this study develops a generic degradation modeling framework to effectively characterize relations between the intrinsic covariate information and degradation processes. The proposed method is inspired by transfer learning, also known as multi-task learning in some context, which refers to transferring knowledge or information from related tasks or domains to improve the model accuracy for a target task or domain [96]. In recent years, transfer learning has emerged as a powerful technique which can benefit the learning of all tasks via appropriate information modeling and transfer between similar tasks [97]. A comprehensive review on transfer learning can be found in Pan and Yang [97] and Weiss *et al*. [96].

The major contributions of this work are summarized as follows. First, based on the covariates of any two units, we use a kernel function to quantify their similarities and adopt the multivariate Gaussian process (MGP) to jointly model their degradation processes. Second, the information in the degradation signals collected from one unit can be effectively transferred to other units and shared with the entire group to improve degradation modeling and prognostics of all individual units. This means that even if there are no new signals collected from an in-service unit, the uncertainties in its RUL prediction can still be decreased as long as new signals are collected from other similar units. As a comparison, most of the existing prognostic approaches based on a Bayesian framework only update the modeling and prediction of the in-service units collecting new signals. Third, the proposed new approach leads to better individualized degradation modeling and prognostics, even when the units of interest have collected only sparse or no degradation signals. This is because the model utilizes basis functions to represent group-level commonalities

and transfers information from one unit to another to compensate for the data shortage of the units of interest.

The rest of this article is organized as follows. In Section 4.2, we review the existing literature on degradation modeling and prognostics of multiple units and those employing the transfer learning idea. Section 4.3 describes the details of the proposed generic framework. Section 4.4 conducts a series of simulation studies to illustrate the effectiveness and to evaluate the sensitivity of the proposed method. Section 4.5 further demonstrates the proposed method based on the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset and compares the results with the existing benchmark methods. Section 4.6 provides our concluding remarks and a discussion of future research directions.

4.2 Literature Review

In this section, we briefly review the relevant literature which considered the degradation modeling and prognostics of multiple units. In the past, extensive efforts have been made to offline model the distribution of the failure time for the entire group using time-based parametric distributions [34], [98]. For instance, Goode *et al*. [98] used two Weibull distributions to model the "installation to potential failure" and the "potential failure to functional failure", and presented a group-level RUL prediction method. One crucial limitation is that pooling the information from all units together fails to consider individual variations. This may lead to significant prognostic errors, especially when the methods are applied to a heterogeneous group in which each unit shows a distinct degradation process.

To characterize unit-to-unit variations, Lu and Meeker [1] proposed a mixed-effect model to describe the degradation processes of multiple units by considering both group-level trends and individual variations through fixed and random effects, respectively. As an extension of this work,

Gebraeel [2] introduced a Bayesian framework to online update the RUL prediction of an in-service unit, once the new degradation signals of this unit are collected. Hong *et al*. [27] further extended the model by explicitly incorporating the effects of dynamic environmental conditions (e.g., UV spectrum and intensity, temperature, and humidity). In particular, the method proposed in [27] assumes the additive effects of dynamic environmental conditions. Note that as discussed in Section 4.1, these dynamic environmental conditions are different from the covariates defined in this study which are intrinsic characteristics of each unit. Recently, Lin *et al.* [31] proposed a collaborative learning framework for degradation modeling and prognostics of a heterogeneous group by using the idea of canonical models and model regularization. One drawback of these approaches is that the units can only share knowledge through a common prior distribution. In other words, the random-effect parameters of an in-service unit are updated only based on its own degradation signal, whereas the information from the historical units is only used to characterize the group-level behavior.

Recently, several approaches to transferring knowledge in degradation modeling and prognostics have been proposed. For example, Li *et al*. [99] proposed a transfer learning method based on PHM, called Transfer-Cox, where the $L_{2,1}$-norm penalty is used to infer the common sparseness of source and target domains, and then learn shared low-dimensional features for knowledge transfer. Zou *et al*. [100] developed a transfer learning method for modeling of degenerate biological systems under the Bayesian framework. Specifically, the model first constructs a graph to represent the qualitative knowledge about the degeneracy, and then converts it into a Laplacian matrix to specify and transfer the model parameters of the source and target domains. Recently, Kontar *et al*. [101] proposed a multi-task learning approach based on Gaussian process regression for degradation modeling and prognostics by treating each degradation signal

as an individual task. Although Gaussian process regression performs well in interpolating test data, its extrapolation performance deteriorates significantly when the test data (in-service units) are far different from the training data (historical units) [102]. Further, one common limitation is that these studies do not consider the intrinsic covariate information to quantitatively model the similarities of units, and thus the models are inapplicable for a newly launched unit.

In summary, existing methods do not take full advantage of the available information embedded in both the collected signals and covariates when performing degradation modeling and prognostics. To fill this literature gap, this study develops a generic degradation modeling framework to effectively quantify similarities between different units based on their intrinsic covariates. In particular, we use basis functions to explicitly model the group-level behaviors while employing the MGP to capture individual-level characteristics simultaneously. As a result, the information transfer among units arises in a natural and principled way, which thus enhances the individualized degradation modeling and prognostics.

4.3 Methodology

4.3.1 Proposed Degradation Framework Incorporating Covariate Information

Suppose there are I heterogeneous units in a group with a single operation condition and a single failure mode. For unit $i \in \{1, \cdots, I\}$, let $L_i(t)$ be the signal measurement collected at time t ($t = 0$ refers to the time when the unit was newly launched). The covariates of unit i are denoted by $\boldsymbol{x}_i = \left[x_{i,1}, \cdots, x_{i,s}\right]^T \in \mathbb{R}^{s\times 1}$, where s is the total number of covariates of each unit. The specific form of $L_i(t)$ is not required; however, without loss of generality, here we decompose $L_i(t)$ as

$$L_i(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_i + \varepsilon_i(t), \tag{4.1}$$

where $\boldsymbol{\psi}(t) = [\psi_1(t), \cdots, \psi_P(t)] \in \mathbb{R}^{1\times P}$ contains a set of basis functions with respect to time t (e.g., $\boldsymbol{\psi}(t) = [1, t, \ldots, t^{P-1}]$ can be used to represent the $(P-1)$thorder polynomial model), $\boldsymbol{\Gamma}_i = \left[\Gamma_{i,1}, \cdots, \Gamma_{i,P}\right]^T \in \mathbb{R}^{P\times 1}$ refers to the random-effect coefficients for unit i, and $\varepsilon_i(t) \sim N(0, \sigma_i^2)$ is the independently distributed noise. Since units in the group exhibit similar degradation mechanisms (e.g., the engine wear of a group of forklifts, the progression of a certain disease for a group of patients), it is reasonable to consider the same set of basis functions $\boldsymbol{\psi}(t)$ among all units to capture the commonalities of the group. We may either adopt domain knowledge or investigate the historical degradation signals to decide the appropriate form of the basis functions. For instance, it is known that the void swelling follows a transient period at low dose until reaching a steady-state growth as shown in Figure 4.1 (b). Therefore, piecewise linear or quadratic basis functions can be used. As another example, Figure 4.9 shows the progression curve (declination of cognitive function) of the Alzheimer's disease for patients. We can observe that patients show similar quadratic degradation (progression) trends. According to the previous studies, $\boldsymbol{\psi}(t) = [1, t, t^2]$ is appropriate to be used to represent the group-level commonalities among all units (patients) for modeling Alzheimer's disease [31], [86], [103]. However, the random-effect coefficients $\boldsymbol{\Gamma}_i$ and σ_i^2 are unique to unit i, which capture the unique individual-level characteristics. As a result, this model allows flexible yet distinct degradation paths for different units. In applications where the degradation paths have a complex form, data-driven basis functions such as functional principal components analysis (FPCA) or spline functions can be used. Similar to the conventional PCA, FPCA finds the set of orthogonal principal component functions that maximize the variance along each component [104]. In prognostics literature, FPCA

has been used to analyze degradation signals and led good RUL prediction performance [105], [106].

Figure 4.2 illustrates the proposed generic framework, where each node represents a single unit, and $\eta_i(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_i$ refers to the underlying degradation process of unit i. The connected line between any two nodes (units) in the group characterizes their similarities. In this figure, units 1, 2, and 3 represent historical units, and unit *I* represents a partially degraded in-service unit. The dashed lines represent the similarities between the historical units and the in-service unit, while the solid lines represent the similarities within the historical units. The proposed method assumes the existence of a relation between random-effect coefficients $\boldsymbol{\Gamma}$ and covariates $\boldsymbol{x}$ of the corresponding unit, i.e., a mapping $\boldsymbol{\Gamma} = \mathcal{F}(\boldsymbol{x})$. In this study, we consider a nonparametric method using MGP to approximate the mapping $\mathcal{F}$, i.e., $\boldsymbol{\Gamma}(\boldsymbol{x}) \sim \mathcal{MGP}\big(\boldsymbol{m}(\boldsymbol{x}), \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')\big)$, where $\boldsymbol{m}(\boldsymbol{x})$ is the mean function and $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')$ is the covariance function.

Many studies have shown that the MGP (an extension of the univariate GP) is a very powerful tool for modeling the vector-valued outputs $\boldsymbol{\Gamma}(\boldsymbol{x})$ at any inputs $\boldsymbol{x} \in \mathbb{R}^{s\times 1}$. Here, we choose the MGP due to its great flexibility to establish the non-parametric relation between $\boldsymbol{\Gamma}$ and $\boldsymbol{x}$, the interpolation capability at any input $\boldsymbol{x}$, and the ability to quantify prediction uncertainties. Specifically, for unit i with covariates $\boldsymbol{x}_i$, the prior distribution of the random-effect coefficients $\boldsymbol{\Gamma}(\boldsymbol{x}_i) = \boldsymbol{\Gamma}_i$ follows a multivariate normal distribution with mean $\boldsymbol{m}(\boldsymbol{x}_i)$ and variance $\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_i)$, i.e., $\boldsymbol{\Gamma}_i \sim \text{MVN}\big(\boldsymbol{m}(\boldsymbol{x}_i), \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_i)\big)$. Note that the normal assumption of the random-effect coefficients $\boldsymbol{\Gamma}$ has been widely used in the existing literature [20], [24], [41], and we will consider the relaxation of the normal assumption in the future study. For any two units $i, j \in \{1, \cdots, I\}$, the covariance between the random-effect coefficients $\boldsymbol{\Gamma}_i$ and $\boldsymbol{\Gamma}_j$ is quantified by $Cov\big(\boldsymbol{\Gamma}_i, \boldsymbol{\Gamma}_j\big) = \boldsymbol{K}\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big)$. Two units with closely related covariates, e.g., Alzheimer's disease patients with the

same ApoE genotypes, would be assigned with a large covariance which eventually leads to similar degradation processes.

This model differs from the existing methods in two fundamental aspects. First, existing methods use a common prior distribution of random-effect coefficients for all units. In contrast, in our method, the prior distribution of $\boldsymbol{\Gamma}_i$ is tailored to each unit according to its covariates $\boldsymbol{x}_i$, and thus it is very flexible and suitable for dealing with the heterogeneous group. Second, whereas most existing methods treat each unit separately, in our method, the covariance $\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ enables us to quantify the similarities between different units, and thus enables effective information transfer. In effect, this means that sensor signals collected from one unit contain not only the information about the degradation status of the unit itself, but also the information to be transferred to other units. As a result, the proposed framework allows us to well characterize the unique degradation path of each unit while taking full advantage of the available information embedded in the collected signals and covariates of units for more accurate degradation modeling and prognostics.

Next, we discuss how to specify $\boldsymbol{m}(\boldsymbol{x})$ and $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')$ in detail. Regarding the mean function, one possible approach is to consider the regression-based techniques. Alternatively, in the numerical studies of this study, we focus on the zero mean function $\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{0}$, which has been widely used in the literature due to the flexibility of the MGP model [107]. The central challenge here is how to model the covariance function of the vector-valued outputs $\boldsymbol{\Gamma}_i$. In the literature, one typical approach is to model the outputs with a separable covariance structure, i.e., via the product of two correlation functions [108]. Yet, since the separable covariance assumption restricts the model to have just one between-outputs correlation function for all the outputs, it may result in poor predictions in some cases [109]. Thus, in this study, we adopt the convolution process to construct

the nonseparable covariance function [110], [111], which is more flexible. In particular, let $\boldsymbol{\Gamma}(\boldsymbol{x}) = [\Gamma_1(\boldsymbol{x}), \cdots, \Gamma_P(\boldsymbol{x})]^T$ and $\boldsymbol{m}(\boldsymbol{x}) = [m_1(\boldsymbol{x}), \cdots, m_P(\boldsymbol{x})]^T$. For each $\Gamma_p^0(\boldsymbol{x}) = \Gamma_p(\boldsymbol{x}) - m_p(\boldsymbol{x})$, $p = 1, \ldots, P$, we construct the output using a convolution process $\Gamma_p^0(\boldsymbol{x}) = \sum_{q=1}^P k_{q,p}(\boldsymbol{x}) \star Z_q(\boldsymbol{x}) = \sum_{q=1}^P \int k_{q,p}(\boldsymbol{u} - \boldsymbol{x}) Z_q(\boldsymbol{u}) d\boldsymbol{u}$, where $\star$ denotes a kernel convolution, $k_{q,p}(\boldsymbol{x})$ is a smoothing kernel, and $Z_q(\boldsymbol{x})$ is a white noise process, i.e., $E[Z_q(\boldsymbol{x})] = 0$, $Cov\left(Z_q(\boldsymbol{x}), Z_q(\boldsymbol{x}')\right) = 0$ for $\boldsymbol{x} \neq \boldsymbol{x}'$, and $Var[Z_q(\boldsymbol{x})] = 1$. The P white noise processes $Z_1(\boldsymbol{x}), \cdots, Z_P(\boldsymbol{x})$ can be regarded as a set of basis functions to construct the MGP. Accordingly, $Cov\left(\boldsymbol{\Gamma}_p(\boldsymbol{x}), \boldsymbol{\Gamma}_{p'}(\boldsymbol{x}')\right) = \sum_{q=1}^P \int k_{q,p}(\boldsymbol{u} - \boldsymbol{x}) k_{q,p'}(\boldsymbol{u} - \boldsymbol{x}')\, d\boldsymbol{u}$. In this way, we are able to construct the covariance matrix for $\boldsymbol{\Gamma}(\boldsymbol{x})$ at any $\boldsymbol{x}$ and $\boldsymbol{x}'$ as follows:

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}') = \begin{bmatrix} Cov\big(\Gamma_1(\boldsymbol{x}), \Gamma_1(\boldsymbol{x}')\big) & \cdots & Cov\big(\Gamma_1(\boldsymbol{x}), \Gamma_P(\boldsymbol{x}')\big) \\ \vdots & \ddots & \vdots \\ Cov\big(\Gamma_P(\boldsymbol{x}), \Gamma_1(\boldsymbol{x}')\big) & \cdots & Cov\big(\Gamma_P(\boldsymbol{x}), \Gamma_P(\boldsymbol{x}')\big) \end{bmatrix}. \tag{4.2}$$

One potential challenge here is that $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')$ may be computationally expensive. To address this issue, we may consider the Gaussian kernel function, which provides an analytical expression of $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')$. Specifically, let the kernel be a scaled Gaussian kernel as follows:

$$\begin{aligned} k_{q,p}(\boldsymbol{x}) &= \alpha_{q,p} \mathrm{MVN}\big(\boldsymbol{x} \big| \boldsymbol{\mu}_{q,p}, \boldsymbol{\Lambda}_{q,p}^{-1}\big) \\ &= \frac{\alpha_{q,p} \sqrt{|\boldsymbol{\Lambda}_{q,p}|}}{\sqrt{(2\pi)^s}} exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_{q,p})^{\mathrm{T}} \boldsymbol{\Lambda}_{q,p}(\boldsymbol{x} - \boldsymbol{\mu}_{q,p})\right\} \end{aligned} \tag{4.3}$$

where $\alpha_{q,p}$ is a constant, $\boldsymbol{\mu}_{q,p}$ is the mean vector, and $\boldsymbol{\Lambda}_{q,p}$ is the precision matrix of the Gaussian kernel. Since the convolution of two multivariate Gaussians is also a multivariate Gaussian, we can write the covariance as

$$Cov\left(\Gamma_p(\boldsymbol{x}), \Gamma_{p'}(\boldsymbol{x}')\right) = \sum_{q=1}^{\mathrm{P}} \alpha_{q,p}\alpha_{q,p'}\mathrm{MVN}\left(\boldsymbol{x}-\boldsymbol{x}'\middle|\boldsymbol{\mu}_{q,p}-\boldsymbol{\mu}_{q,p'}, \mathbf{C}_{q,p,p'}\right)$$

$$= \sum_{q=1}^{\mathrm{P}} \frac{\alpha_{q,p}\alpha_{q,p'}}{\sqrt{(2\pi)^s\left|\mathbf{C}_{q,p,p'}\right|}} exp\left\{-\frac{1}{2}\left((\boldsymbol{x}-\boldsymbol{x}')-\left(\boldsymbol{\mu}_{q,p}-\boldsymbol{\mu}_{q,p'}\right)\right)^{\mathrm{T}} {\mathbf{C}_{q,p,p'}}^{-1}\left((\boldsymbol{x}-\boldsymbol{x}')\right.\right.$$

$$\left.\left.-\left(\boldsymbol{\mu}_{q,p}-\boldsymbol{\mu}_{q,p'}\right)\right)\right\}$$

where ${\mathbf{C}_{q,p,p'}}^{-1} = \left(\boldsymbol{\Lambda}_{q,p}^{-1}+\boldsymbol{\Lambda}_{q,p'}^{-1}\right)^{-1} = \boldsymbol{\Lambda}_{q,p}\left(\boldsymbol{\Lambda}_{q,p}+\boldsymbol{\Lambda}_{q,p'}\right)^{-1}\boldsymbol{\Lambda}_{q,p'}$.

Note that other kernel functions can be incorporated in MGP as well. For example, we can use a separable covariance structure with a linear kernel function to consider the case when the relation between $\boldsymbol{\Gamma}$ and $\boldsymbol{x}$ is simple and approximately linear, which may save computational costs and avoid overfitting.

In the cases where units have categorical covariates (e.g., model type), we may apply conventional transformations from categorical values to continuous values (e.g., one-hot encoding) or use Gaussian processes explicitly designed for both categorical and continuous inputs [112]. For instance, consider the general case with J categorical covariates $\boldsymbol{z} = \left[z_1, \ldots, z_J\right]$, where z_j has m_j levels. Following Qian *et al*. [112], we can introduce an $m_j \times m_j$ positive definite matrix with unit diagonal elements $\mathcal{T}_j = (\tau_{j,z_j,z_j'})$ and extend the covariance function as follows:

$$Cov\left(\Gamma_p(\boldsymbol{z},\boldsymbol{x}), \Gamma_{p'}(\boldsymbol{z}',\boldsymbol{x}')\right) = \prod_{j=1}^{J}\tau_{j,z_j,z_j'}\sum_{q=1}^{\mathrm{P}}\int k_{q,p}(\boldsymbol{u}-\boldsymbol{x})k_{q,p'}(\boldsymbol{u}-\boldsymbol{x}')\,d\boldsymbol{u}.$$

This is also a valid covariance function as it is the product of two valid covariance functions.

4.3.2 Parameter Estimation

For the proposed modeling framework in Section 4.3.1, the parameters to be estimated include the ones in $\boldsymbol{m}(\boldsymbol{x})$ and $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')$, and the variance of the random noise σ_i^2. Let $\boldsymbol{\Omega_m}$ be the set of parameters in $\boldsymbol{m}(\boldsymbol{x})$, $\boldsymbol{\Omega}_k(q,p) = \{\alpha_{q,p}, \boldsymbol{\Lambda}_{q,p}, \boldsymbol{\mu}_{q,p}\}$ be the set of parameters in the kernel function $k_{q,p}(\boldsymbol{x})$, $\boldsymbol{\Omega_K} = \cup_{q,p} \boldsymbol{\Omega}_k(q,p)$ be the set of parameters in $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')$, and $\boldsymbol{\Omega_\varepsilon} = \{\sigma_i^2, i = 1, \dots, I\}$.

Let $\boldsymbol{\Gamma} = [\boldsymbol{\Gamma}_1; \cdots; \boldsymbol{\Gamma}_I] \in \mathbb{R}^{IP\times 1}$ be all the random-effect coefficients and $\boldsymbol{X} = [\boldsymbol{x}_1; \cdots; \boldsymbol{x}_I] \in \mathbb{R}^{Is\times 1}$ be all the covariates of the units. Denote $\boldsymbol{m}(\boldsymbol{X}) = \begin{pmatrix} \boldsymbol{m}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{m}(\boldsymbol{x}_I) \end{pmatrix} \in \mathbb{R}^{IP\times 1}$ and $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) = \begin{bmatrix} \boldsymbol{K}(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & \boldsymbol{K}(\boldsymbol{x}_1, \boldsymbol{x}_I) \\ \vdots & \ddots & \vdots \\ \boldsymbol{K}(\boldsymbol{x}_I, \boldsymbol{x}_1) & \cdots & \boldsymbol{K}(\boldsymbol{x}_I, \boldsymbol{x}_I) \end{bmatrix} \in \mathbb{R}^{IP\times IP}$. For unit i, let the collected degradation signals be $\boldsymbol{L}_i = \left[L_i\left(t_{i,1}\right), \cdots L_i\left(t_{i,n_i}\right)\right]^T \in \mathbb{R}^{n_i\times 1}$, where $t_{i,j}$ is the jth signal measurement time of unit i, and n_i is the total number of signal measurements of unit i. Note that one advantage of the proposed approach is that the signals do not have to be collected at equidistant times. Denote $\boldsymbol{L} = [\boldsymbol{L}_1; \dots; \boldsymbol{L}_I] \in \mathbb{R}^{(\Sigma n_i)\times 1}$ to be the concatenated degradation signals, $\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Psi}_1 & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{\Psi}_I \end{bmatrix} \in \mathbb{R}^{(\Sigma n_i)\times IP}$ to be the design matrix, and $\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}_1; \cdots; \boldsymbol{\varepsilon}_I] \in \mathbb{R}^{(\Sigma n_i)\times 1}$ to be all errors, where $\boldsymbol{\Psi}_i = \begin{bmatrix} \boldsymbol{\psi}\left(t_{i,1}\right) \\ \vdots \\ \boldsymbol{\psi}\left(t_{i,n_i}\right) \end{bmatrix} \in \mathbb{R}^{n_i\times P}$ and $\boldsymbol{\varepsilon}_i = \begin{bmatrix} \varepsilon_i(t_{i,1}) \\ \vdots \\ \varepsilon_i(t_{i,n_i}) \end{bmatrix} \in \mathbb{R}^{n_i\times 1}$. Then, (4.1) can be rewritten in the matrix form:

$$\boldsymbol{L} = \boldsymbol{\Psi\Gamma} + \boldsymbol{\varepsilon}. \tag{4.4}$$

Since $\boldsymbol{\Gamma}\sim\text{MVN}\left(\boldsymbol{m}(\boldsymbol{X}), \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})\right)$, $\boldsymbol{L}$ also follows the multivariate normal distribution with

$$E(\boldsymbol{L}) = \boldsymbol{\Psi m}(\boldsymbol{X}) \text{ and } Var(\boldsymbol{L}) = \boldsymbol{\Psi K}(\boldsymbol{X}, \boldsymbol{X})\boldsymbol{\Psi}^T + \boldsymbol{\Sigma_\varepsilon}. \tag{4.5}$$

Here, $\boldsymbol{\Sigma_\varepsilon} = \text{diag}\left(\sigma_1^2 \boldsymbol{I}_{n_1 \times n_1}, \cdots, \sigma_I^2 \boldsymbol{I}_{n_I \times n_I}\right) \in \mathbb{R}^{(\Sigma n_i)\times(\Sigma n_i)}$, where $\boldsymbol{I}_{n\times n}$ is a $n \times n$ identity matrix. To estimate the parameters, we can apply the maximum likelihood estimation (MLE) approach:

$$\left(\widehat{\boldsymbol{\Omega}}_{\boldsymbol{m}}, \widehat{\boldsymbol{\Omega}}_{\boldsymbol{K}}, \widehat{\boldsymbol{\Omega}}_{\boldsymbol{\varepsilon}}\right) = \underset{\boldsymbol{\Omega_m}, \boldsymbol{\Omega_K}, \boldsymbol{\Omega_\varepsilon}}{\text{argmax}} \log L_p(\boldsymbol{L}|\boldsymbol{\Omega_m}, \boldsymbol{\Omega_K}, \boldsymbol{\Omega_\varepsilon}).$$

Here, $\log L_p(\boldsymbol{L}|\boldsymbol{\Omega_m}, \boldsymbol{\Omega_K}, \boldsymbol{\Omega_\varepsilon}) = -\frac{1}{2}\log|\boldsymbol{\Psi K}(\boldsymbol{X},\boldsymbol{X})\boldsymbol{\Psi}^T + \boldsymbol{\Sigma_\varepsilon}| - \frac{1}{2}\left(\boldsymbol{L} - \boldsymbol{\Psi m}(\boldsymbol{X})\right)^T (\boldsymbol{\Psi K}(\boldsymbol{X},\boldsymbol{X})\boldsymbol{\Psi}^T + \boldsymbol{\Sigma_\varepsilon})^{-1}\left(\boldsymbol{L} - \boldsymbol{\Psi m}(\boldsymbol{X})\right)$. Solving this optimization problem is computationally expensive as it requires the inversion of a large matrix $\boldsymbol{\Psi K}(\boldsymbol{X},\boldsymbol{X})\boldsymbol{\Psi}^T + \boldsymbol{\Sigma_\varepsilon} \in \mathbb{R}^{(\Sigma n_i)\times(\Sigma n_i)}$. To address this issue, we can use the matrix inversion lemma which greatly speeds up the computation:

$$(\boldsymbol{\Psi K}(\boldsymbol{X},\boldsymbol{X})\boldsymbol{\Psi}^T + \boldsymbol{\Sigma_\varepsilon})^{-1} = \boldsymbol{\Sigma_\varepsilon^{-1}} - \boldsymbol{\Sigma_\varepsilon^{-1}}\boldsymbol{\Psi}(\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})^{-1} + \boldsymbol{\Psi}^T\boldsymbol{\Sigma_\varepsilon^{-1}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^T\boldsymbol{\Sigma_\varepsilon^{-1}}.$$

Since the dimension of the matrix $\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})^{-1} + \boldsymbol{\Psi}^T\boldsymbol{\Sigma_\varepsilon^{-1}}\boldsymbol{\Psi}$ is $IP \times IP$, calculating the inverse of this matrix could be much more efficient.

4.3.3 Posterior Parameter Distribution

Using the collected degradation signals, we can update the distribution of the random-effect coefficients $\boldsymbol{\Gamma}_i$ for prognostics. Whereas most of the existing methods calculate the posterior distribution of each unit separately, the proposed method models the covariance between $\boldsymbol{\Gamma}_i$s by

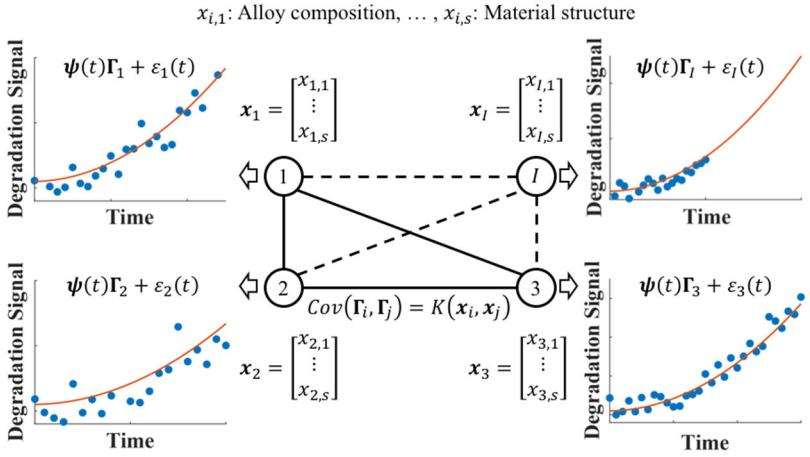


Figure 4.2 Proposed degradation framework for modeling the interrelations of units.

considering all units simultaneously through the concatenation $\boldsymbol{\Gamma} = [\boldsymbol{\Gamma}_1; \cdots; \boldsymbol{\Gamma}_I]$ as shown in Figure 4.2. Next, we discuss three scenarios for updating the posterior distribution of $\boldsymbol{\Gamma}$. In this way, we quantitatively evaluate the benefit of the information transfer enabled by the proposed model.

Scenario 1: batch updating according to all available signals. In the first scenario, we focus on updating the distribution $\boldsymbol{\Gamma} \sim \text{MVN}\big(\boldsymbol{m}(\boldsymbol{X}), \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})\big)$ based on all available signals $\boldsymbol{L}$. Since $\boldsymbol{L}|\boldsymbol{\Gamma} \sim \text{MVN}(\boldsymbol{\Psi}\boldsymbol{\Gamma}, \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}})$, we can show that the posterior distribution $\boldsymbol{\Gamma}|\boldsymbol{L}$ also follows the multivariate normal distribution:

$$\boldsymbol{\Gamma}|\boldsymbol{L} \sim \text{MVN}\big(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)}\big), \tag{4.6}$$

where $\boldsymbol{\mu}^{(1)} = \boldsymbol{\Sigma}^{(1)}\big(\boldsymbol{\Psi}^T\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{L} + [\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})]^{-1}\boldsymbol{m}(\boldsymbol{X})\big)$ and $\boldsymbol{\Sigma}^{(1)} = (\boldsymbol{\Psi}^T\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{\Psi} + [\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})]^{-1})^{-1}$. The proof is given in Section 4.7.1.

Scenario 2: online updating according to a newly collected measurement. In the second scenario, we suppose a new measurement is collected that can be used to further update the posterior distribution of $\boldsymbol{\Gamma}$. Denote $L_i(t)$ to be the new measurement from unit i. Then, the posterior distribution will be updated by $p(\boldsymbol{\Gamma}|\boldsymbol{L}, L_i(t)) \propto p(\boldsymbol{\Gamma}|\boldsymbol{L})p(L_i(t)|\boldsymbol{\Gamma}, \boldsymbol{L}) = p(\boldsymbol{\Gamma}|\boldsymbol{L})p(L_i(t)|\boldsymbol{\Gamma}_i)$. According to (4.6), we can further derive that this posterior distribution is normally distributed:

$$\boldsymbol{\Gamma}|\boldsymbol{L}, L_i(t) \sim \text{MVN}\big(\boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma}^{(2)}\big), \tag{4.7}$$

where $\boldsymbol{\mu}^{(2)} = \boldsymbol{\Sigma}^{(2)}\left(\frac{\boldsymbol{\delta}(i,t)^T L_i(t)}{\sigma_i^2} + \big(\boldsymbol{\Sigma}^{(1)}\big)^{-1}\boldsymbol{\mu}^{(1)}\right)$ and $\boldsymbol{\Sigma}^{(2)} = \left(\frac{\boldsymbol{\delta}(i,t)^T\boldsymbol{\delta}(i,t)}{\sigma_i^2} + \big(\boldsymbol{\Sigma}^{(1)}\big)^{-1}\right)^{-1}$. Here, $\boldsymbol{\delta}(i,t) = [\boldsymbol{\delta}_1(i,t), \cdots, \boldsymbol{\delta}_I(i,t)] \in \mathbb{R}^{1\times IP}$ is a vector with entries $\boldsymbol{\delta}_i(i,t) = \boldsymbol{\psi}(t)$ for unit i, and all other entries equal to $\mathbf{0}$ (i.e., $\forall j \neq i$, $\boldsymbol{\delta}_j(i,t) = \mathbf{0}$). The proof of (4.7) is similar to the proof of (4.6) in Section 4.7.1 and thus is omitted.

Using Scenario 2 as an example, we elaborate on how much benefit the proposed method can bring compared to the existing methods that treat each unit separately. If a new measurement is collected from unit i, then the existing methods only update the posterior distribution of $\boldsymbol{\Gamma}_i$. In contrast, for our proposed method, we can show the following proposition:

Proposition: Once a new measurement is collected from unit i, not only the variance for the posterior distribution of $\boldsymbol{\Gamma}_i$, but also those for other units can be decreased. (Details of the proof can be found in Section 4.7.2.)

If we consider each unit separately like in the existing methods, only the posterior variance for the units with new measurements will be updated, and those for other units remain the same. On the contrary, the proposition shows that in our proposed method, the uncertainties for other units are also decreased. This proposition further provides a theoretical justification for the advantage of our proposed framework.

Scenario 3: "cold start" case where the unit of interest $j \notin \{1, \cdots, I\}$ is newly launched, i.e., joins the group with covariates $\boldsymbol{x}_j$ and has not yet been collected with any degradation signals. This has been a long-standing challenge in the existing literature, especially when $\boldsymbol{x}_j$ is different from the covariates of historical units. We focus on updating $\boldsymbol{\Gamma}_j$ and show how we can leverage the degradation information from other units to compensate for the data shortage of unit j. Specifically, the random-effect coefficients of the historical units and those of the newly launched unit j have a joint Gaussian distribution:

$$\begin{pmatrix} \boldsymbol{\Gamma}_j \\ \boldsymbol{\Gamma} \end{pmatrix} \sim \text{MVN}\left(\begin{pmatrix} \boldsymbol{m}(\boldsymbol{x}_j) \\ \boldsymbol{m}(\boldsymbol{X}) \end{pmatrix}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{x}_j) & \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{X}) \\ \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_j) & \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) \end{bmatrix} \right), \tag{4.8}$$

where $\boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{X}) = \left(\boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{x}_1), \cdots, \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{x}_I)\right)$. Since $p(\boldsymbol{\Gamma}_j|\boldsymbol{\Gamma}) = p(\boldsymbol{\Gamma}_j, \boldsymbol{\Gamma})/p(\boldsymbol{\Gamma})$, we can show that $\boldsymbol{\Gamma}_j|\boldsymbol{\Gamma} \sim \text{MVN}\left(\boldsymbol{\mu}_j^{(1)}, \boldsymbol{\Sigma}_j^{(1)}\right)$, where $\boldsymbol{\mu}_j^{(1)} = \boldsymbol{m}(\boldsymbol{x}_j) + \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{X})[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})]^{-1}\left(\boldsymbol{\Gamma} - \boldsymbol{m}(\boldsymbol{X})\right)$ and $\boldsymbol{\Sigma}_j^{(1)} = \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{x}_j) - \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{X})[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})]^{-1}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_j)$. Since $p(\boldsymbol{\Gamma}_j|\boldsymbol{L}) = \int p(\boldsymbol{\Gamma}_j|\boldsymbol{\Gamma})p(\boldsymbol{\Gamma}|\boldsymbol{L})\, d\boldsymbol{\Gamma}$, we can further prove that

$$\boldsymbol{\Gamma}_j|\boldsymbol{L} \sim \text{MVN}\left(\boldsymbol{\mu}_j^{(2)}, \boldsymbol{\Sigma}_j^{(2)}\right), \tag{4.9}$$

where $\boldsymbol{\mu}_j^{(2)} = \boldsymbol{m}(\boldsymbol{x}_j) + \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{X})[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})]^{-1}\left(\boldsymbol{\mu}^{(1)} - \boldsymbol{m}(\boldsymbol{X})\right) \in \mathbb{R}^{P\times 1}$, $\boldsymbol{\Sigma}_j^{(2)} = \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{X})[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})]^{-1}\left(\boldsymbol{\Sigma}^{(1)}\right)[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})]^{-1}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_j) + \boldsymbol{\Sigma}_j^{(1)} \in \mathbb{R}^{P\times P}$, and $\boldsymbol{\mu}^{(1)}$, $\boldsymbol{\Sigma}^{(1)}$ are the mean and variance of $\boldsymbol{\Gamma}|\boldsymbol{L}$ as defined in (4.6). The proof of (4.9) is similar to the proof of (4.6) in Section 4.7.1 and thus is omitted. On the contrary, without information transfer, the variance of $\boldsymbol{\Gamma}_j$ would be $\boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{x}_j)$. Since

$$\boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{x}_j) - \boldsymbol{\Sigma}_j^{(2)} = \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{X})[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})]^{-1}\left[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) - \boldsymbol{\Sigma}^{(1)}\right][\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})]^{-1}\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_j) \succcurlyeq \boldsymbol{0},$$

where $\boldsymbol{M} \succcurlyeq \boldsymbol{0}$ means that the matrix $\boldsymbol{M}$ is positive semi-definite, the variance of $\boldsymbol{\Gamma}_j$ is decreased by transferring the knowledge from other units to unit j. Specifically, we can see that the covariance function $\left(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_j), \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{X})\right)$ enables the information transfer, which makes it possible to immediately model the degradation process of a newly launched unit even without any signal measurements from this unit.

4.3.4 Prognostics

Based on the posterior distribution $\boldsymbol{\Gamma}|\boldsymbol{L}$ as described in Section 4.3.3, we can obtain the posterior distribution of $\boldsymbol{\Gamma}_r$ of an in-service unit r via a Bayesian approach in real time. For instance, under Scenario 1, the posterior distribution of $\boldsymbol{\Gamma}_r$ can be obtained as $\boldsymbol{\Gamma}_r|\boldsymbol{L} \sim \text{MVN}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$, where $\boldsymbol{\mu}_r \in$

$\mathbb{R}^{P\times 1}$ is the rth sub-vector of $\boldsymbol{\mu}^{(1)}$ and $\boldsymbol{\Sigma}_r \in \mathbb{R}^{P\times P}$ is the rth diagonal sub-matrix of $\boldsymbol{\Sigma}^{(1)}$. Note that for notation simplicity, here $\boldsymbol{L}$ refers to all available signals from all units including unit r up to the current observation time. Then, we can derive the distribution estimation of the future degradation status $\eta_r(t) = \boldsymbol{\psi}(t)\boldsymbol{\Gamma}_r$ at time t as follows:

$$\eta_r(t)|\boldsymbol{L} \sim \text{MVN}(\boldsymbol{\psi}(t)\boldsymbol{\mu}_r, \boldsymbol{\psi}(t)\boldsymbol{\Sigma}_r\boldsymbol{\psi}(t)^T). \tag{4.10}$$

We can further predict the RUL of a partially degraded in-service unit based on the posterior distribution in (4.10). Specifically, following the existing studies [2], [12], [74], the failure time T_i of unit i is defined as the time when its degradation status $\eta_i(t)$ first reaches a predefined failure threshold D:

$$T_i = \underset{t}{\text{argmin}}\, \eta_i(t) \geq D. \tag{4.11}$$

The conditional cumulative distribution function (CDF) of the failure time of unit r is $F_{T_r}(t|\boldsymbol{L}) = P(T_r \leq t|\boldsymbol{L}) = P(\eta_r(t) \geq D|\boldsymbol{L})$. Given that unit r has not failed yet at t_{r,n_r}, i.e., $T_r > t_{r,n_r}$, the CDF can be updated in real time given the latest measurement time t_{r,n_r}:

$$F_{T_r}\left(t|\boldsymbol{L}, T_r > t_{r,n_r}\right) = \frac{P(\eta_r(t) \geq D|\boldsymbol{L}) - P\left(\eta_r\left(t_{r,n_r}\right) \geq D\middle|\boldsymbol{L}\right)}{1 - P\left(\eta_r\left(t_{r,n_r}\right) \geq D\middle|\boldsymbol{L}\right)}. \tag{4.12}$$

Since this CDF is skewed, the failure time $\hat{T}_r$ can be estimated as the median of the CDF, i.e., $F_{T_r}\left(\hat{T}_r|\boldsymbol{L}, T_r > t_{r,n_r}\right) = 0.5$. As a result, the estimated RUL of unit r is $\hat{T}_r - t_{r,n_r}$. Given some degradation signals have been collected from unit r, the CDF of the failure time has a closed-form expression:

$$F_{T_r}\left(t|\boldsymbol{L}, T_r > t_{r,n_r}\right) = \frac{\Phi\left(g_r(t)\right) - \Phi\left(g_r\left(t_{r,n_r}\right)\right)}{1 - \Phi\left(g_r\left(t_{r,n_r}\right)\right)},$$

where Φ denotes the CDF of the standard normal distribution and $g_r(t) = \frac{\boldsymbol{\psi}(t)\boldsymbol{\mu}_r - D}{\sqrt{\boldsymbol{\psi}(t)\boldsymbol{\Sigma}_r\boldsymbol{\psi}(t)^T}}$ (the detailed derivation can be referred to Liu *et al*. [20]).

4.4 Simulation Studies

In this section, a series of numerical studies are carried out to demonstrate the effectiveness and sensitivity of the proposed method. In Section 4.4.1, we first introduce four benchmark methods that are used in the numerical studies and compare them with the proposed method. Section 4.4.2 introduces how we generate the simulated dataset. Sections 4.4.3-4.4.5 study the RUL prediction accuracy of the proposed method and the benchmark methods under Scenarios 1-3 described in Section 4.3.3, respectively. Finally, in Section 4.4.6, we consider the scenario when there are irrelevant covariates present. We note that computational comparisons based on the simulation dataset are provided in Section 4.7.3.

4.4.1 Benchmark Approaches

In the following, we compare the proposed method with four benchmark methods. The first benchmark method is the parametric Bayesian mixed-effect model (BMEM), which is one of the most widely used techniques in the literature on degradation modeling [2]. This model first fits a mixed-effect model based on the degradation signals of the historical units, and then updates the random-effect coefficients of the in-service units under the Bayesian framework by incorporating online degradation signals. Yet, this method does not consider the covariates of units and separately models the degradation process of each testing unit.

Another possible approach is to investigate the regression function for $\boldsymbol{\Gamma}_i$ on $\boldsymbol{x}_i$ by locally fitting a line, also known as the varying-coefficient model (VCM) [113], [114]. The VCM was proposed to analyze the models in the form of $Y_i = \boldsymbol{Z}_i'\boldsymbol{\beta}(\boldsymbol{X}_i) + \varepsilon_i$, where Y_i is a response variable, $\boldsymbol{Z}_i$ is a

vector of predictors, $\boldsymbol{X}_i$ is a vector of covariates, $\boldsymbol{\beta}(\cdot)$ is an unknown function, and ε_i is a random error. The VCM is a semi-parametric method which approximates the function $\boldsymbol{\beta}(\cdot)$ using a kernel smoothing based on the observed Y_i, $\boldsymbol{Z}_i$, and $\boldsymbol{X}_i$. By substituting Y_i, $\boldsymbol{Z}_i$, $\boldsymbol{X}_i$, and $\boldsymbol{\beta}(\cdot)$ into $L_i(t), \boldsymbol{\psi}(t), \boldsymbol{x}_i$, and $\boldsymbol{\Gamma}(\cdot)$, respectively, we can use the VCM to estimate the regression function $\boldsymbol{\Gamma}(\boldsymbol{x})$ and obtain $\boldsymbol{\Gamma}_{\boldsymbol{r}}$ of the in-service unit r given its covariates $\boldsymbol{x}_r$. One major drawback of the VCM is that it only provides a point estimate of $\boldsymbol{\Gamma}_i$. Thus, it cannot effectively capture the stochastic nature of degradation processes, and more importantly, it does not provide the interval estimates of RUL. In this study, following the existing literature [113], [114], we use a Gaussian kernel and the leaving-one-out cross-validation to estimate the kernel bandwidth.

The univariate GP (uGP) is considered as another benchmark method in which the model incorporates the covariates, but not the group-level commonalities. Instead of employing a MGP which takes the P-dimensional random-effect coefficients $\boldsymbol{\Gamma}_i \in \mathbb{R}^{P\times 1}$ as one output, we can consider the uGP which takes each degradation signal $L_i\left(t_{i,j}\right) \in \mathbb{R}^{1\times 1}$ as one output, and covariates and signal measurement time as one input. We can extrapolate the fitted GP model up to the point when the estimated signal first reaches the failure threshold D to predict the failure time. Compared to the proposed method that adopts basis functions $\boldsymbol{\psi}(t)$ to capture the commonalities of all units, the uGP is a fully nonparametric model and does not take advantage of the group-level characteristics. As a result, although the uGP is flexible, it requires much more historical data to make an accurate prediction.

The last benchmark method is the multivariate Gaussian Convolution Process (MGCP) recently proposed in [101]. Unlike the uGP where each degradation signal is treated separately, here all degradation signals are represented as individual outputs from one common MGP. As a result, the

Table 4.1 Comparison of the Proposed Method and the Benchmark Methods

	Incorporation of Intrinsic Covariates	Group-level Degradation Form
Proposed	O	O
BMEM		O
VCM	O	O
uGP	O	
MGCP		O

information in the degradation signals collected from one unit can be shared with other units through this MGP. However, the model does not incorporate the covariate information to quantify the similarities between different units, and is also inapplicable for a newly launched unit. To ensure a fair comparison, here we use a zero mean function and Gaussian kernel when conducting the uGP, the MGCP and the proposed method. Also, all methods based on group-level basis functions use the same set of basis functions.

Table 4.1 highlights the difference between the proposed method and the benchmark methods.

4.4.2 Data Generation

Without loss of generality, we consider a heterogeneous group where each unit has a one-dimensional covariate, i.e., $x_i \in \mathbb{R}^{1\times 1}$, and the degradation signal follows a second-order polynomial degradation process, i.e., $P = 3$, where $\varepsilon_i(t) \sim N(0,5^2)$:

$$L_i(t) = \Gamma_{i,0} + \Gamma_{i,1}t + \Gamma_{i,2}t^2 + \varepsilon_i(t). \tag{4.13}$$

Note that the homoscedastic errors $\varepsilon_i(t)$, i.e., $\sigma_i = 5$ for all units i, are adopted for ease of computation, although the proposed method can handle heteroskedastic errors as explained in Section 4.3.1. The random-effect coefficients $\boldsymbol{\Gamma}_i(x) = \left[\Gamma_{i,0}(x), \Gamma_{i,1}(x), \Gamma_{i,2}(x)\right]^T$ are generated according to the following non-linear functions:

$$\Gamma_{i,0}(x) = x^2 + 0.5\sqrt{x} + 0.2\sin(10x) + U_{i,0}(x), \tag{4.14}$$

$$\Gamma_{i,1}(x) = 0.5x^2 + \sqrt{x} + 0.2\sin(15x) + U_{i,1}(x), \text{and}$$

$$\Gamma_{i,2}(x) = 1.8 - \left(1.5x^2 + 0.2\sqrt{x} + 0.2\sin(5x)\right) + U_{i,2}(x),$$

where $U_{i,0}(x), U_{i,1}(x), U_{i,2}(x) \sim \text{Uniform}(0,0.2)$, and the failure threshold is set as $D = 500$. The covariate x_i of each unit is sampled from $\text{Uniform}(0,1)$. If any $\boldsymbol{\Gamma}_i$ with $\Gamma_{i,2}(x) \leq 0$ or $\Gamma_{i,1}(x) \leq 0$ is generated, we reject the sample and generate a new one to ensure the monotonicity of the underlying degradation process. The observed failure time of unit i is denoted as τ_i. The degradation signals of unit i are collected at times $t = 1, \ldots, n_i$.

In Sections 4.4.3-4.4.6, we assume that all units in a group have quadratic degradation forms as in (13), which can be known a priori based on domain knowledge or analysis of historical records; however, the exact relationships between $\boldsymbol{\Gamma}_i$ and $\boldsymbol{x}_i$ as specified in (14) are unknown. Thus, we use the proposed MGP to approximate the relationship functions in (14) throughout Section 4.4, which ensures the wide applicability of the proposed approach in practice. Figure 4.3 (a) and (b) show the random-effect coefficients and degradation signals of 50 randomly generated units,

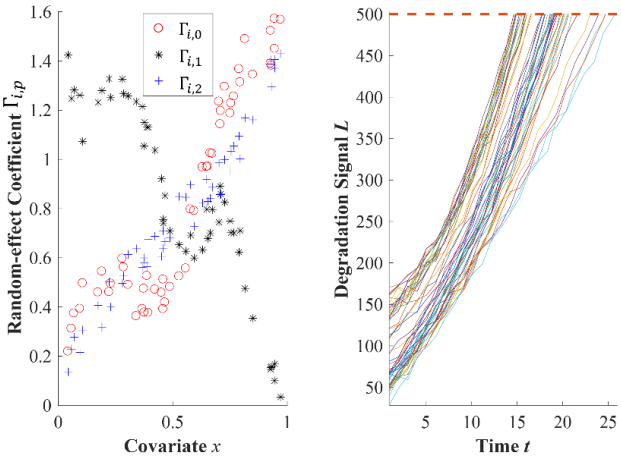


Figure 4.3 (a) Random-effect coefficients plot and (b) Degradation signals plot for 50 randomly generated units. The dashed line in (b) represents the failure threshold D.

respectively. We can see that while all units follow a similar quadratic degradation form, depending on their covariates, different units fail at different times.

4.4.3 Scenario 1 - Batch Updating

In the first simulation, based on the available degradation information of all the units, we predict the RUL of all in-service units using the proposed method and the benchmark methods. First, we randomly generate $n_{tr} = 30$ training units and $n_{ts} = 30$ testing units. Here, the training units are historical units whose degradation signals are collected until the failure occurs, whereas the testing units are in-service units whose degradation signals are available until some random time points prior to the failure. We use the term "unobserved percentage" to denote this truncation point. For example, "75% unobserved" means that for each testing unit, only the first 25% of degradation signals are observed. In assessing the prediction error, we use the mean absolute percentage error of failure time for testing units where we calculate the absolute percentage error of failure time of unit j as follows.

$$err_j = \frac{\left|T_j - \hat{T}_j\right|}{T_j}. \tag{4.15}$$

The simulations are carried out for different unobserved percentages. For each selected unobserved percentage, the procedure is replicated 100 times. Figure 4.5 illustrates the estimation of the random-effect coefficients for the proposed method, the BMEM, and the VCM when the unobserved percentage is 45%. We can see that the proposed method provides a more accurate and stable $\boldsymbol{\Gamma}_i$ estimation compared to the BMEM and the VCM. Figure 4.4 shows the RUL prediction results of the proposed method and the benchmark methods. The x-axis represents the unobserved percentage. The error bars show one standard deviation of the prediction errors. From

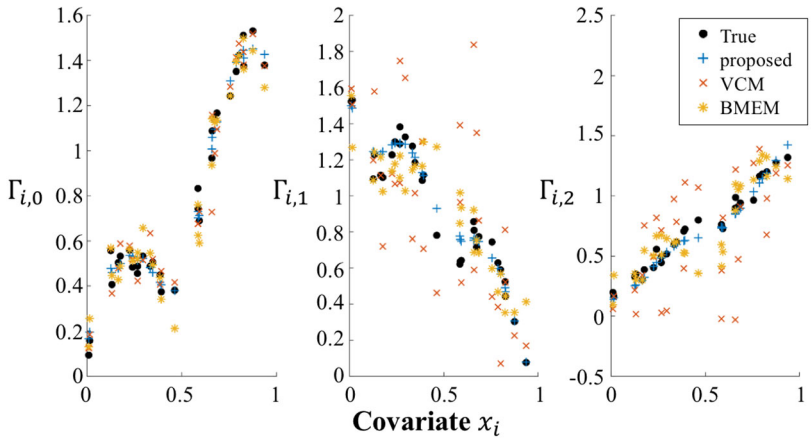


Figure 4.5 The random-effect coefficients estimation for the testing units when n_{tr}=30, n_{ts}=30, and the unobserved percentage is 45% by using the proposed method, the BMEM, and the VCM.

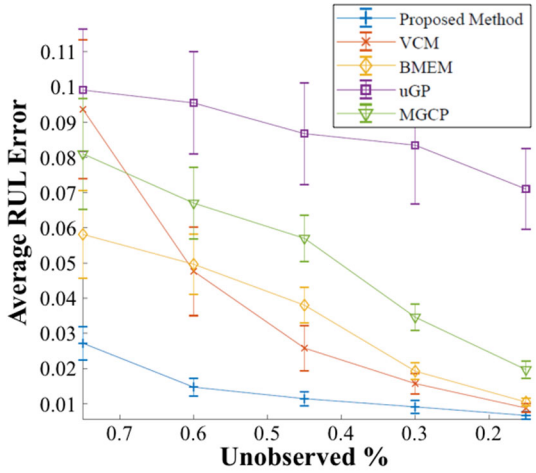


Figure 4.4 The RUL prediction errors for the testing units when n_{tr}=30 and n_{ts}=30 by using the benchmark methods and the proposed method.

Figure 4.4, we can see that the proposed method yields much lower RUL prediction errors than the benchmark methods. The figure also demonstrates that in all approaches, the estimation improves as the unobserved percentage decreases, i.e., as more data are available for each testing unit. Interestingly, although the BMEM does not employ any covariate x, it outperforms the uGP and the VCM at 75% unobserved point. This is probably because 1) in the VCM, when only limited degradation signals for testing units are available, it easily leads to a poor estimation of the kernel

bandwidth [115]; and 2) the uGP fails to incorporate the group-level similarities and its high model flexibility tends to overfit the data.

To see how the RUL prediction changes with the number of training units n_{tr}, we repeat the simulations while randomly selecting n_{tr} training units, but fix the number of testing units to 20, i.e., $n_{ts} = 20$. For each selected number of training units, the procedure is replicated 100 times. Table 4.2 shows the RUL prediction results of the proposed method and the benchmark methods when the unobserved percentage is (a) 75% and (b) 15%. The smallest mean prediction error for each value of n_{tr} and the unobserved percentage is highlighted in boldface. The results show that the accuracy increases as more training units are available for all methods, and the proposed method consistently outperforms other benchmarks under both low and high unobserved percentages. For similar reasons as in Figure 4.4, the left half of Table 4.2 again shows that when the unobserved percentage is high, the BMEM outperforms the VCM and the uGP, although the VCM and the uGP employ the covariates x and the BMEM does not. Generally speaking, it is difficult to accurately estimate the random-effect coefficients of testing units until their degradation signals have been collected for a certain period of time and show clear trends. Thus, even though the uGP does not consider the quadratic degradation process form, it shows

Table 4.2 RUL Prediction Errors and Corresponding Standard Errors (values in parentheses)

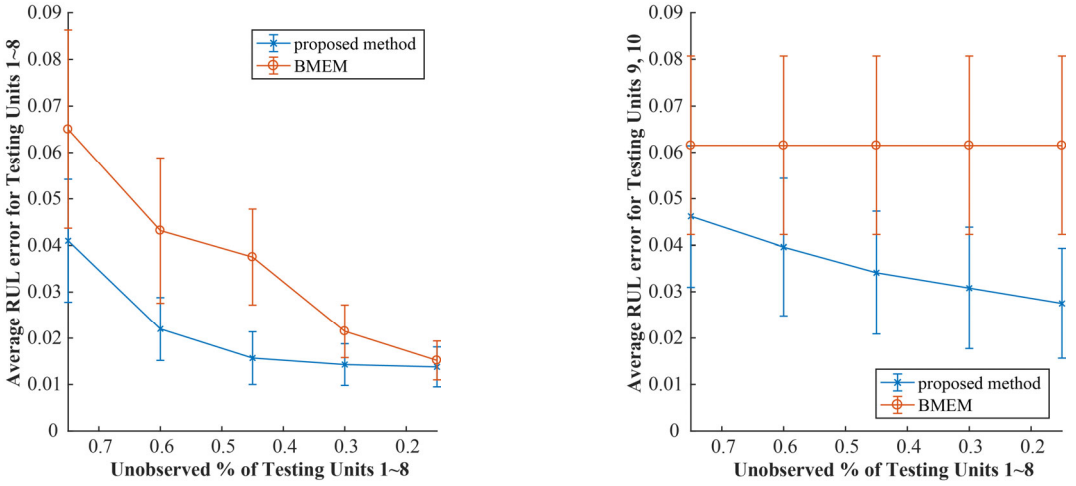
	75% unobserved					*15% unobserved*				
n_{tr}	proposed	VCM	BMEM	uGP	MGCP	proposed	VCM	BMEM	uGP	MGCP
10	**0.0642**	0.1148	0.0647	0.1049	0.1079	**0.0076**	0.0087	0.0107	0.0811	0.0246
	(0.0142)	(0.0168)	(0.0107)	(0.0149)	(0.0182)	**(0.0010)**	(0.0013)	(0.0014)	(0.0135)	(0.0082)
20	**0.0297**	0.1062	0.0623	0.1031	0.0867	**0.0071**	0.0085	0.0100	0.0808	0.0225
	(0.0043)	(0.0177)	(0.0109)	(0.0131)	(0.0152)	**(0.0009)**	(0.0014)	(0.0016)	(0.0095)	(0.0034)
30	**0.0288**	0.0951	0.0597	0.1003	0.0810	**0.0062**	0.0090	0.0097	0.0743	0.0207
	(0.0036)	(0.0159)	(0.0097)	(0.0120)	(0.0161)	**(0.0009)**	(0.0012)	(0.0012)	(0.0087)	(0.0026)
40	**0.0270**	0.0892	0.0592	0.0941	0.0673	**0.0058**	0.0080	0.0094	0.0649	0.0172
	(0.0031)	(0.0115)	(0.0100)	(0.0114)	(0.0132)	**(0.0007)**	(0.0013)	(0.0011)	(0.0092)	(0.0021)
50	**0.0265**	0.0884	0.0576	0.0921	0.0589	**0.0052**	0.0082	0.0095	0.0642	0.0151
	(0.0027)	(0.0102)	(0.0086)	(0.0109)	(0.0105)	**(0.0007)**	(0.0010)	(0.0011)	(0.0085)	(0.0023)

comparable performance to other approaches when the unobserved percentage is high. However,

as the testing units approach the end of life, i.e., low unobserved percentage, the uGP and MGCP show worse performance than the other three methods as shown in Figure 4.4 and the right half of Table 4.2. The uGP particularly shows much worse performance than the MGCP since it treats each degradation signal separately and fails to learn the overall degradation trends shared with the entire units. This result shows that using fully nonparametric methods can easily lead to overfitting of the data and poor prediction results.

4.4.4 Scenario 2 - Online Updating

In this subsection, we will further carry out numerical studies to highlight the benefit of establishing similarities among units and compare the online updating results described in Section 4.3.3 Scenario 2 with the BMEM. In each iteration, we randomly generate $n_{tr} = 30$ training units and $n_{ts} = 10$ testing units, and initially set the unobserved percentage to 75% for all testing units. The proposed method and the BMEM are applied to estimate the RUL for all testing units. Next, we suppose testing units 1~8 collect new degradation signals, i.e., the unobserved percentage of testing units 1~8 decreases, and then we update RUL predictions with new measurements. The simulations are repeated 100 times for each selected unobserved percentage of testing units 1~8. Figure 4.7 (a) shows the average RUL prediction errors of testing units 1~8 and (b) shows those of testing units 9 and 10. The x-axis represents the unobserved percentage of testing units 1~8. The error bars show one standard deviation of the prediction errors. From Figure 4.7 (a) and (b), we can see that the BMEM only improves the RUL predictions for the testing units that have new measurements, whereas the proposed method updates RUL predictions for all testing units. For better illustration, we show the failure time estimations of testing unit 1 and testing unit 10 in Figure 4.6. The figure demonstrates that the proposed method simultaneously reduces the



(a) (b)

Figure 4.7 The average RUL prediction errors for (a) testing units 1~8 and (b) testing units 9 and 10 when n_{tr}=30 and n_{ts}=10 by using the proposed method and the BMEM.

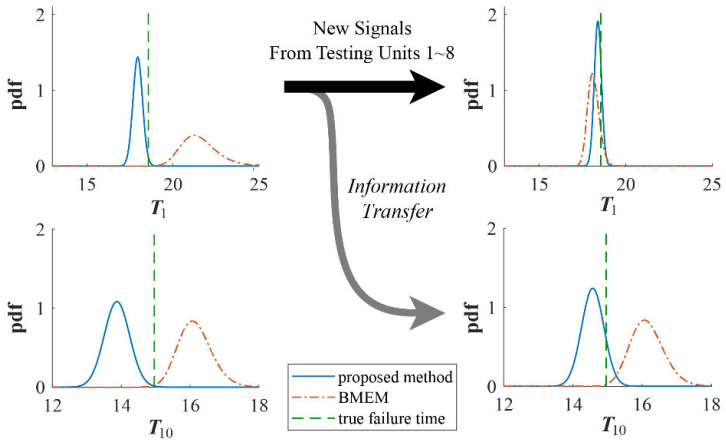


Figure 4.6 The estimated failure time's probability density function (pdf) for testing unit 1 (the upper plots) and testing unit 10 (the lower plots) when the unobserved percentage of testing unit 1 is 75% (the left plots) and 15% (the right plots) with n_{tr}=30 and $n_{ts} = 10$ by using the proposed method and the BMEM.

prediction uncertainty for all testing units, whereas the BMEM only reduces the prediction uncertainty for the testing units with the new measurements.

4.4.5 Scenario 3 – Cold-Start Scenario

One long-standing challenge in the literature is how to conduct degradation and prognostic analysis when there is a newly launched unit whose degradation signals have not yet been measured and no historical units in the group have the same covariates as this new unit. In this subsection, we generate the dataset as described in Section 4.4.2, but the unobserved percentage of testing units are 100%, i.e., there are no degradation signals available for testing units, and we only know their covariates. The number of testing units is set to $n_{ts} = 20$. Here, since all testing units are newly launched, instead of assessing RUL prediction, we consider the mean absolute percentage error where the error of testing unit j' $\boldsymbol{\Gamma}$s are as follows:

$$err_j = \frac{1}{P}\left(\sum_{p=1}^{P} \frac{\left|\hat{\Gamma}_{j,p} - \Gamma_{j,p}\right|}{\left|\Gamma_{j,p}\right|}\right). \tag{4.16}$$

Here, $\hat{\Gamma}_{j,p}$ is the estimated value of $\Gamma_{j,p}$.

We repeat the parameter estimation of testing units 100 times for each selected number of training units n_{tr}, and the results for the proposed method and the BMEM are shown in Table 4.3. Note that in the cold-start scenario, the covariates of the testing units are not included in the bandwidth estimation procedure of the VCM, as there are no corresponding degradation signals

Table 4.3 Parameter Estimation Errors under Cold-Start Scenario

n_{tr}	*Cold-Start Scenario*	
	proposed method	BMEM
10	**0.1483 (0.0286)**	0.4359 (0.0736)
20	**0.1363 (0.0291)**	0.4167 (0.0458)
30	**0.1031 (0.0265)**	0.4052 (0.0443)
40	**0.0918 (0.0249)**	0.3985 (0.0384)
50	**0.0859 (0.0213)**	0.3909 (0.0392)
75	**0.0774 (0.0206)**	0.3961 (0.0396)
100	**0.0726 (0.0192)**	0.3917 (0.0313)

available to fit the regression function $\boldsymbol{\Gamma}(\boldsymbol{x})$. As a result, the VCM's estimation results are very poor and highly sensitive to the dissimilarities between the testing and the training units, and thus it is excluded in this subsection. For the BMEM, since we do not have any degradation signals of testing units to derive the posterior distribution, we estimate the random-effect coefficients by the mean of the prior distribution. Table 4.3 shows that the proposed method yields much lower errors in estimating the random-effect coefficients than the BMEM. The table also demonstrates that using the proposed approach, the estimation improves as the number of training units increases. However, the BMEM shows a much larger variance than the proposed method and after n_{tr} exceeds 50, further increases in the value of n_{tr} do not provide clear improvements in the parameter estimation. This is because the estimation accuracy of the BMEM depends much on the similarities between the training units and the testing units.

4.4.6 Sensitivity to Irrelevant Covariates

In practice, it is common that there are some covariates which are irrelevant to the underlying degradation process and deteriorate the prediction accuracy by acting as noise. Thus, in this subsection, studies are conducted to test the sensitivity of the proposed method to the inclusion of irrelevant covariates. We generate a new dataset following the same procedures as described in Section 4.4.2, except that each unit has two-dimensional covariates, i.e., $\boldsymbol{x}_i \in \mathbb{R}^{2\times 1}$, where $\boldsymbol{x}_i = [x_i, \tilde{x}_i]^T$. For each unit, both x_i and $\tilde{x}_i$ are derived from $\mathrm{Uniform}(0,1)$ and form the covariates $\boldsymbol{x}_i$, but only x_i is used to calculate $\boldsymbol{\Gamma}_i$ as described in (4.14) and $\tilde{x}_i$ is a covariate which is irrelevant to the underlying degradation process. We apply the proposed method to the new dataset, while still assuming both x_i and $\tilde{x}_i$ are informative covariates.

Following the same procedure as in the previous subsection, n_{tr} units are randomly selected as training units. The number of testing units and the unobserved percentage is fixed to 20 and 50%. Then, the RUL prediction is repeated 100 times for each n_{tr}, and the results are shown in Figure 4.8. Although both the BMEM and MGCP do not use covariates, here we compare the proposed method with the BMEM since it outperforms MGCP in Section 4.4.3. Note that the results using the uGP and the VCM are not shown since they showed much higher errors with the presence of irrelevant covariates compared to the BMEM and the proposed method. In particular, we observed that the uGP overfitted the data according to the irrelevant covariates and the VCM showed unstable performance resulting from wrong estimations of kernel bandwidth. Figure 4.8 indicates that the prediction accuracy of the proposed method with the irrelevant covariate gets closer to that of the proposed method with only the informative covariate as n_{tr} increases. This is probably because as n_{tr} increases, the estimated parameters $\boldsymbol{\Omega_K}$ in the kernel function (4.3) is adjusted to put less weights on the covariate $\tilde{x}_i$ and more weights on the covariate x_i. As a result, the information transferred between two units depends more on the similarity of x_i of the two units

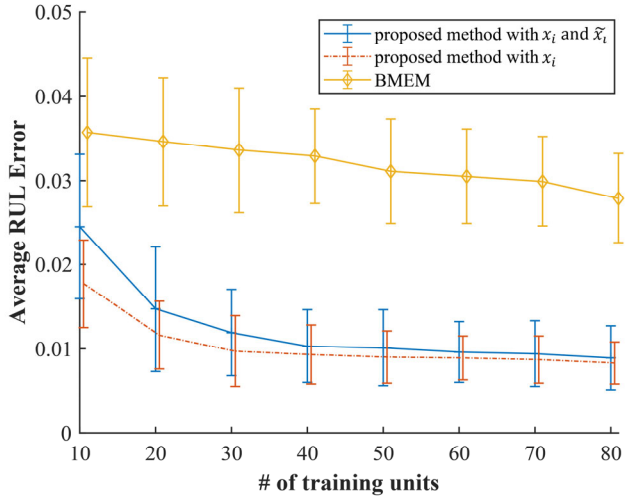


Figure 4.8 The RUL prediction errors for the testing units by using the proposed method with the inclusion of an irrelevant covariate, the proposed method with only the informative covariate, and the BMEM which does not use any covariates.

than that of $\tilde{x}_i$. Furthermore, even with the presence of the irrelevant covariate $\tilde{x}_i$, the figure shows that the proposed method effectively characterizes the similarities between different units based on their informative covariates x_i, and greatly outperforms the BMEM.

4.5 Case Study

In this section, we further evaluate the proposed method in a real healthcare case study, where we aim to accurately model the progression of a disease (declination of cognitive function) and predict the future health status of the patients. This case study is based on the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset, which is publicly available at *http://adni.loni.usc.edu/* [116]. We consider the ADNI patients as one example of a heterogeneous group since it possesses some commonalities resulting from the progression of the same disease, while each patient has unique intrinsic covariates including the demographic, genetic, and imaging information such as MRI images. Similar to Section 4.4, the prognostic results are compared with the four benchmark approaches: the BMEM, the VCM, the uGP, and the MGCP.

4.5.1 Description of the Dataset

The ADNI dataset contains personal information and longitudinal measurements of examinations and biomarkers that are related to the Alzheimer's disease progression for the participating patients. In particular, this case study aims to predict the corresponding Mini-Mental State Estimation (MMSE) score, which is commonly used to measure the cognitive status of the patients [31], [86], [117], [118]. MMSE is a neuropsychological test reflecting the severity of Alzheimer's disease [119]. The MMSE ranges between 0 (worst condition) and 30 (best condition), and it is expected to decrease with time for all patients. The MMSE score is collected repeatedly over a 6-month or 1-year interval for each patient. Patients may join the program at

different AD stages, skip some of their examinations, and drop out from the study for various reasons. Thus, analyzing the ADNI dataset is very challenging as the dataset is sparse in nature.

Following the existing studies [31], [86], [103], [120], we model the degradation of the MMSE measurements as a quadratic model with respect to time. In this study, we use a total of 7 covariates which are patients' basic nature or characteristics that do not change over time. These features were identified as potential ones influencing the Alzheimer's disease progression and widely used to predict the cognitive status [117], [121], [122]: 1) the ApoE genotypes; 2) the baseline MMSE score; and 3) the baseline MRI via FreeSurfer (here the "baseline" refers to time $t = 0$ when the patients first join the program). For the baseline MRI, 5 types of MRI features are extracted: surface area (Surf. Area), cortical thickness average (CTA), cortical thickness standard deviation (CTStd), white matter parcellation volume (Vol.WM.), and cortical parcellation volume (Vol.C.) [117]. We exclude the participants with missing covariates. To ensure the stable construction of degradation models, we consider 433 participating patients who have 6 or more MMSE scores with available measurements at the baseline, 6^{th} month, 12^{th} month, 24^{th} month, etc.

4.5.2 Results and Comparison

Among the 433 preselected patients, 81 patients have 8 or more MMSE scores, and we refer to this group as "TR subgroup" which includes patients who either show clear disease progression or maintain steady cognitive status for a sufficiently long period of time. To ensure a reliable degradation model, throughout the case study, patients used for model training are drawn from the "TR subgroup" whereas testing patients are drawn from the rest of the 433 preselected patients. Unlike Section 4.4 in which RUL prediction of a partially degraded in-service unit is the primary goal, here we want to model the declination of cognitive function and memory loss due to the Alzheimer's Disease and predict the future health status of the patients. To evaluate this prognostic

performance for testing patients, we predict the last two MMSE scores and calculate the prediction errors by following the same settings and procedures of the proposed method and the benchmark methods as in Section 4.4. As an illustrative example, Figure 4.9 shows the prediction results for 16 selected testing patients who have different covariates and show various disease progressions, when we randomly select 20 training patients from TR subgroup. From the figure, we can see that the proposed method predicts the hidden scores more accurately than the benchmark methods.

Next, we explore how the prognostic performance changes with the number of patients in the training group. Specifically, we randomly draw n_{tr} patients from the TR subgroup for training and draw 20 testing patients from the rest of $433 - n_{tr}$ patients. For each testing patient, we predict the last two MMSE measurements and calculate the mean of the absolute error between the predicted and the true MMSE scores, i.e., averaging $\left|L_j(t_{j,k}) - \hat{L}_j(t_{j,k})\right|$ over $k = n_j - 1, n_j$, where $L_j(t_{j,k})$

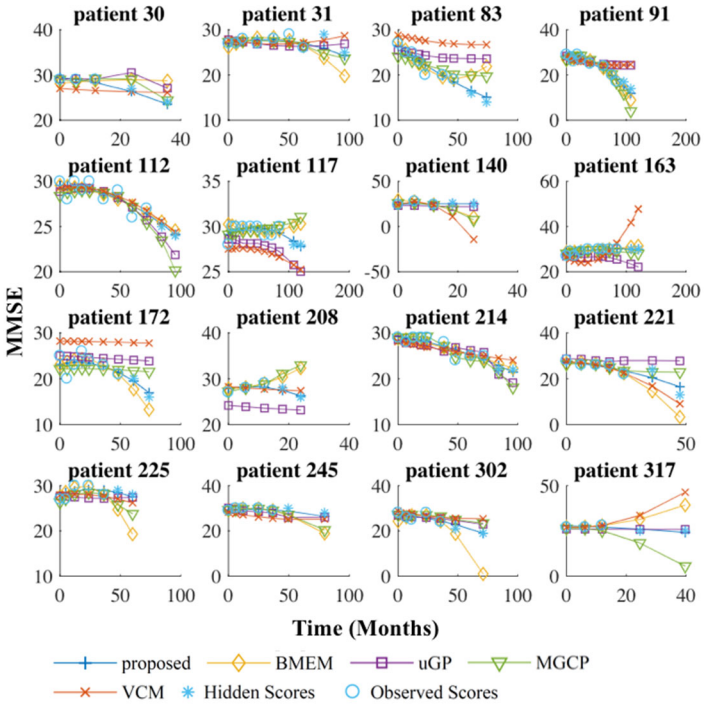


Figure 4.9 True and predicted MMSE scores from a subset of patients by using the proposed method, the BMEM, the uGP, the MGCP, and the VCM.

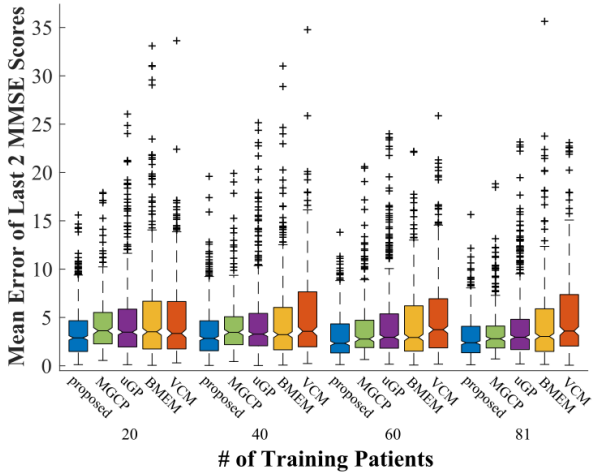


Figure 4.10 The last two MMSE score prediction errors for the testing patients when $n_{ts} = 20$ by using the proposed method, the MGCP, the uGP, the BMEM, and the VCM.

is the kth MMSE score of patient j, and $\hat{L}_j(t_{j,k})$ is the estimated value of it. For each value of n_{tr}, this procedure is repeated 50 times. The results are shown in Figure 4.10.

From Figure 4.10, we can see that 1) by effectively characterizing the similarity between patients, the proposed method obtains overall more accurate prediction results compared to the four benchmark methods, 2) as more training data are available, i.e., as n_{tr} increases, it results in fewer outliers in prediction, and 3) yet there is a weak trend showing the prognostic performance improvement with more training data. One possible reason is due to the sparsity and the high noise of the ADNI dataset. Also, there may be unobserved potential covariates which influence the disease progression and the similarities between patients, but are not measured in the examinations. The uGP is a fully nonparametric model and does not take advantage of the group-level basis functions. As a result, it shows lower accuracy and higher variance than the proposed method. The MGCP yields better prognostic performance than the uGP by sharing information among different units through the MGP. However, this model does not employ covariate information, and thus shows higher errors than the proposed method. The BMEM and the VCM both show high

prediction errors and more outliers than the proposed method, the MGCP, and the uGP. This may be due to the dissimilarities between the training and testing patients when using the BMEM and the poor estimations of kernel bandwidths when using the VCM. Interestingly, although the BMEM does not utilize covariates, it outperforms the VCM in general. This shows the importance of considering the stochastic nature of degradation processes and the risk of assuming a deterministic relationship between $\boldsymbol{\Gamma}_i$ and $\boldsymbol{x}_i$ when dealing with a heterogeneous group.

We note that computational costs based on the ADNI dataset according to the number of training patients are provided in Section 4.7.3.

4.6 Conclusion

In this study, we developed a generic framework for individualized degradation modeling and prognostics of a heterogeneous group. In particular, the degradation processes of units share the same basis functions to represent the group-level similarities, whereas the random-effect coefficients model the individual characteristics of each unit. In modeling the random-effect coefficients, the proposed method uses MGP with a kernel convolution to encode the available knowledge about intrinsic covariates into the prior distributions of the random-effect coefficients and quantify the similarities between different units. This quantification determines how much information to be transferred from one unit to another. As a result, the proposed method enables the degradation information from one unit to be shared with the entire group. The theoretical justification for the proposed framework was also investigated. The numerical study results showed that the proposed method provides a more stable degradation model and more accurate prognostic performance compared to the existing methods.

There are several potential topics for future research. First, this study assumes that covariates are static. However, in practice, there may be time-varying covariates which also dynamically

affect degradation processes. The extension to the inclusion of time-varying covariates will be of great interest in our future research. One possibility is to build an additional GP model to estimate the changes in dynamic covariates and formulate the problem into two steps: predicting the changes of covariates and then incorporating those effects into degradation modeling and prognostics. Second, this study assumes that the random-effect coefficients $\boldsymbol{\Gamma}_i$ follow a multivariate normal distribution. In future work, we will study how to relax the normality assumption for $\boldsymbol{\Gamma}_i$, e.g., using a mapping $\mathcal{H}$ such that $\mathcal{H}(\boldsymbol{\Gamma}_i)$ has a multivariate normal distribution, or using numerical methods such as Monte Carlo Markov Chain. Third, in real-world applications, there may be a negative transfer of knowledge where the unit of interest is not well-related to the historical units, so the knowledge transfer potentially deteriorates the prognostics. A systematic approach to minimize or to detect the negative transfer of knowledge is an area of future research. For example, we may reduce the negative transfer via regularizing the covariance matrix $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ by thresholding, and thus the information is transferred between different units only when their similarities are significant. Last but not least, this study targets applications where we know important covariates and aim to model the relationships between these covariates and degradation processes. However, this should be further expanded to the cases where the sparsity of covariates is high, i.e., the majority of covariates are irrelevant to degradation processes, and thus the proposed method must be preceded by a proper covariate selection procedure.

4.7 Appendix

4.7.1 The proof of the eq. (4.6)

Since $\boldsymbol{L}|\boldsymbol{\Gamma}\sim\text{MVN}(\boldsymbol{\Psi\Gamma},\boldsymbol{\Sigma_\varepsilon})$ and $\boldsymbol{\Gamma}\sim\text{MVN}\big(\boldsymbol{m}(\boldsymbol{X}),\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})\big)$, we can see that the posterior distribution $\boldsymbol{\Gamma}|\boldsymbol{L}$ also follows the multivariate normal distribution as illustrated in (4.17). The proofs of the equations (4.7) and (4.9) can be derived in a similar way and thus are omitted.

$$
\begin{aligned}
p(\boldsymbol{\Gamma}|\boldsymbol{L}) &\propto p(\boldsymbol{L}|\boldsymbol{\Gamma})p(\boldsymbol{\Gamma}) \\
&\propto \exp\left(-\frac{1}{2}(\boldsymbol{L}-\boldsymbol{\Psi\Gamma})^T\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}^{-1}(\boldsymbol{L}\right. \\
&\left.-\boldsymbol{\Psi\Gamma})\right)\exp\left(-\frac{1}{2}\big(\boldsymbol{\Gamma}-\boldsymbol{m}(\boldsymbol{X})\big)^T[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})]^{-1}\big(\boldsymbol{\Gamma}-\boldsymbol{m}(\boldsymbol{X})\big)\right) \\
&= \exp\left(-\frac{1}{2}\Big(\boldsymbol{\Gamma}^T(\boldsymbol{\Psi}^T\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{\Psi}+[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})]^{-1})\boldsymbol{\Gamma}\right. \\
&-\boldsymbol{\Gamma}^T\big(\boldsymbol{\Psi}^T\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{L}+[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})]^{-1}\boldsymbol{m}(\boldsymbol{X})\big) \\
&-\big(\boldsymbol{\Psi}^T\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{L}+[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})]^{-1}\boldsymbol{m}(\boldsymbol{X})\big)^T\boldsymbol{\Gamma}+\boldsymbol{L}^T\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{L} \\
&\left.+\boldsymbol{m}(\boldsymbol{X})^T[\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})]^{-1}\boldsymbol{m}(\boldsymbol{X})\Big)\right) \\
&\propto \exp\left(-\frac{1}{2}\big(\boldsymbol{\Gamma}-\boldsymbol{\mu}^{(1)}\big)^T{\boldsymbol{\Sigma}^{(1)}}^{-1}\big(\boldsymbol{\Gamma}-\boldsymbol{\mu}^{(1)}\big)\right),
\end{aligned}
\tag{4.17}
$$

4.7.2 The proof of the Proposition

We focus on $\boldsymbol{\Sigma}^{(1)}-\boldsymbol{\Sigma}^{(2)}$ in (4.6) and (4.7) to evaluate the variation reduction in $\boldsymbol{\Gamma}$ when a new signal is collected. Without loss of generality, we assume the new measurement is collected from unit 1. Then, we can partition $\boldsymbol{\Sigma}^{(1)}$ into four blocks:

$$
\boldsymbol{\Sigma}^{(1)}=\begin{bmatrix}\boldsymbol{\Sigma}_{1,1}^{(1)} & \boldsymbol{\Sigma}_{1,-1}^{(1)}\\ \boldsymbol{\Sigma}_{-1,1}^{(1)} & \boldsymbol{\Sigma}_{-1,-1}^{(1)}\end{bmatrix},
$$

where $\boldsymbol{\Sigma}_{1,1}^{(1)} \in \mathbb{R}^{P\times P}$ denotes the variance matrix of $\boldsymbol{\Gamma}_1$, $\boldsymbol{\Sigma}_{1,-1}^{(1)}$ and $\boldsymbol{\Sigma}_{-1,1}^{(1)}$ denote the covariance between $\boldsymbol{\Gamma}_1$ and other $\boldsymbol{\Gamma}_i$s $(i \neq 1)$, and $\boldsymbol{\Sigma}_{-1,-1}^{(1)} \in \mathbb{R}^{(I-1)P\times(I-1)P}$ is the covariance matrix between all $\boldsymbol{\Gamma}_i$s except $\boldsymbol{\Gamma}_1$ $(i \neq 1)$. By matrix inversion lemma, we can prove that

$$\boldsymbol{\Sigma}^{(1)} - \boldsymbol{\Sigma}^{(2)} = \begin{bmatrix} \boldsymbol{\Sigma}_{1,1}^{(1)}\left(\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)} + \boldsymbol{I}\right)^{-1}\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)} & \boldsymbol{\Sigma}_{1,1}^{(1)}\left(\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)} + \boldsymbol{I}\right)^{-1}\boldsymbol{A}\boldsymbol{\Sigma}_{1,-1}^{(1)} \\ \boldsymbol{\Sigma}_{-1,1}^{(1)}\left(\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)} + \boldsymbol{I}\right)^{-1}\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)} & \boldsymbol{\Sigma}_{-1,1}^{(1)}\left(\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)} + \boldsymbol{I}\right)^{-1}\boldsymbol{A}\boldsymbol{\Sigma}_{1,-1}^{(1)} \end{bmatrix},$$

where $\boldsymbol{A} = \frac{\boldsymbol{\delta}_1(1,t)^T\boldsymbol{\delta}_1(1,t)}{\sigma_1^2} \in \mathbb{R}^{P\times P}$. In other words, the posterior variance of $\boldsymbol{\Gamma}_1$ is updated to $\boldsymbol{\Sigma}_{1,1}^{(1)} - \boldsymbol{\Sigma}_{1,1}^{(1)}\left(\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)} + \boldsymbol{I}\right)^{-1}\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)}$, and that of $\boldsymbol{\Gamma}_{-1} = [\boldsymbol{\Gamma}_2; \cdots; \boldsymbol{\Gamma}_I]$ is updated to $\boldsymbol{\Sigma}_{-1,-1}^{(1)} - \boldsymbol{\Sigma}_{-1,1}^{(1)}\left(\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)} + \boldsymbol{I}\right)^{-1}\boldsymbol{A}\boldsymbol{\Sigma}_{1,-1}^{(1)}$. Recall that $\boldsymbol{\Sigma}^{(2)} = \left(\frac{\boldsymbol{\delta}(i,t)^T\boldsymbol{\delta}(i,t)}{\sigma_i^2} + \left(\boldsymbol{\Sigma}^{(1)}\right)^{-1}\right)^{-1}$, which means $\left(\boldsymbol{\Sigma}^{(2)}\right)^{-1} - \left(\boldsymbol{\Sigma}^{(1)}\right)^{-1} = \frac{\boldsymbol{\delta}(i,t)^T\boldsymbol{\delta}(i,t)}{\sigma_i^2} \succcurlyeq \boldsymbol{0}$, where $\boldsymbol{M} \succcurlyeq \boldsymbol{0}$ means that the matrix $\boldsymbol{M}$ is positive semi-definite. According to corollary 7.7.4 of [123], we can conclude that $\boldsymbol{\Sigma}^{(1)} - \boldsymbol{\Sigma}^{(2)} \succcurlyeq \boldsymbol{0}$, which represents a variance reduction for all units. In contrast, if we consider each unit independently, only the posterior variance of $\boldsymbol{\Gamma}_i$ will be updated to $\boldsymbol{\Sigma}_{1,1}^{(1)} - \boldsymbol{\Sigma}_{1,1}^{(1)}\left(\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)} + \boldsymbol{I}\right)^{-1}\boldsymbol{A}\boldsymbol{\Sigma}_{1,1}^{(1)}$, and those for other units remain the same.

4.7.3 Computational costs

We compare the computational costs for offline parameter estimation and online prognostics using the proposed method and the benchmark methods. The simulation data is generated following Section 4.4.2. The number of training and testing units is fixed to 20 and 10, and the unobserved % is set to 75%. All models are tested using Intel Core i5-6300U CPU 2.40-GHz and 16-GB RAM. The simulations are repeated 10 times. The results are shown in Table 4.4. Although the proposed method takes a longer time for parameter estimation than the existing parametric

approach (BMEM), please note that this procedure is carried out offline. In online, the RUL estimation of each testing unit takes around 0.03 seconds using the proposed method. This result shows that once the parameter estimation is finished (Section 4.3.2), the online updating and prognosis (Sections 4.3.3 and 4.3.4) can be done much faster with closed-form expressions.

Table 4.5 shows the computational costs of the proposed method on the ADNI dataset in Section 4.5 which considers a total of 7 covariates. The simulations are repeated 10 times for each value of the number of training units. The number of testing units is fixed to 10. Note that "prognostics" here means the prediction of future MMSE scores which takes a much shorter time than the prediction of RUL distribution.

Table 4.4 Average Computational Costs for Offline Parameter Estimation and Online Prognostics on Simulation Dataset

	Offline Parameter Estimation (s)	Online Prognostics (s)
Proposed	34.13	0.029
MGCP	36.56	0.032
uGP	3.61	0.034
BMEM	0.16	0.027
VCM	36.68	0.006

Table 4.5 Average Computational Costs for Offline Parameter Estimation and Online Prognostics on ADNI Dataset

n_{tr}	Offline Parameter Estimation (s)	Online Prognostics (s)
10	59.48	0.00022
20	119.42	0.00024
40	310.83	0.00021

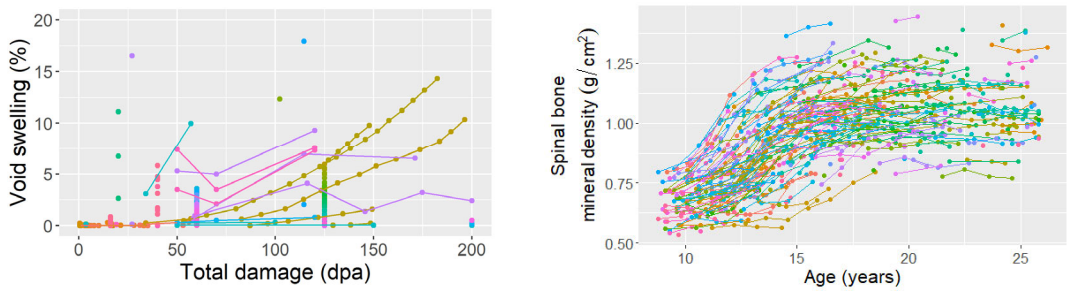
Chapter 5 Covariate-dependent functional data analysis

5.1 Introduction

5.1.1 Motivation

Void swelling is a nuclear-specific material degradation mechanism that causes an increase in the volume of components exposed to high-energy neutrons at high temperatures [124]. To mitigate the effect of swelling and ensure the continued functionality and reliability in nuclear power plants, it is crucial to (i) identify informative covariates, such as irradiation conditions, that significantly affect components' swelling trajectories and (ii) predict a swelling trajectory of a new component of interest based on the given covariates.

However, to effectively analyze and model swelling processes, several significant challenges need to be overcome. First, there are many covariates, such as alloy composition (e.g., a typical material has 5∼10 alloying elements in its composition), irradiation type (e.g., neutron, proton), and irradiation temperature affecting void swelling. Identifying important covariates and characterizing the effects of those covariates on swelling is a critical research challenge that existing studies have not well resolved. Second, the swelling data is in the form of a sparse dataset that often provides only a few points along the swelling with respect to the damage trajectory of a particular subject with specific covariate information as in Figure 5.1 (a) (one curve or one dot represents the swelling measurements given specific covariates). This is because the sample acquisition is particularly expensive and time-consuming in the void swelling application due to



(a) (b)

Figure 5.1 Examples of sparse functional data with covariate information: (a) void swelling and (b) spinal bone mineral density

the strict requirements of sample preparation, resource-intensive post-irradiation examination, and safety concerns [125].

While our research is primarily motivated by the void swelling application, it is common nowadays to aim for the modeling and prediction of the functional observations which are coupled with covariate information. For instance, the degradation signals (longitudinal measurements) of engineering systems are collected in addition to the manufacturing design information or operation information (covariates) of the systems [19], [25]. In healthcare applications, it is more common to observe sparse functional data than other applications since patients or participants often skip some of their examinations and drop out of the study [31], [86]. Figure 5.1 (b) describes an example of the spinal bone mineral density (BMD) according to age [126]. The dataset consists of 1~4 measurements of spinal BMD from each participant and their sex and ethnicity (covariates). We will explore the dataset in more detail in Section 5.4.2.

Note that the proposed method focuses on the static covariates that often relate to the basic nature of the subjects and do not change over time [127]. We will consider the expansion to the inclusion of dynamic covariates in our future research. It is also important to mention that the term "covariates" in this article is distinguished with some existing studies, especially on cox proportional hazards model (PHM), where "covariates" often directly refers to sensor signal

measurements [9]–[11]. For instance, for the aforementioned void swelling example, PHM-based method would consider swelling measurements as covariates instead of alloy composition or irradiation temperature.

5.1.2 Literature Review

Functional data analysis (FDA) is a powerful tool for sparse and irregularly sampled longitudinal or spatial data. It treats measurements as functions and leverages the underlying smoothness to model the data [128]. In particular, functional principal component analysis (FPCA) is one of the most widely-used FDA methods to reduce the dimensionality of functional data, which is intrinsically infinite-dimensional. Similar to conventional PCA, FPCA explores major variations of sample curves by finding principal component functions that are orthogonal and maximizes the curve variation.

Consider the random function $X(t)$ in a compact time interval $\mathcal{T}$ with mean function $\mu(t)$, covariance function $\Sigma(t,t')$, and covariance operator $\Sigma(g) = \int_{\mathcal{T}} \Sigma(s,t)g(s)ds$ for any function g satisfying $\mathbb{E}\left(\int_{\mathcal{T}} g^2(s)ds\right) < \infty$. Under mild assumptions, Mercer's theorem [129] implies that the covariance operator Σ has orthonormal eigenfunctions (principal component functions) $\phi_k(t), k = 1,2,\ldots$, with nonincreasing eigenvalues λ_k, i.e., $\lambda_1 \geq \lambda_2 \geq \cdots$, satisfying $\Sigma(\phi_k) = \lambda_k\phi_k$. This leads to the well-known Karhunen–Loève decomposition of the random function $X(t)$:

$$X(t) = \mu(t) + \sum_{k=1}^{\infty} A_k\phi_k(t)\,, \tag{5.1}$$

where the kth functional principal component (FPC) score is $A_k = \int_{\mathcal{T}} \{X(t) - \mu(t)\}\phi_k(t)dt$. A_k are random variables uncorrelated across k, i.e., $cov(A_k, A_{k'}) = 0$ if $k \neq k'$, with $\mathbb{E}(A_k) = 0$ and $var(A_k) = \lambda_k$. The top few FPCs explain most of the variability in the random curves, and thus

$X(t)$ can be approximated by a linear combination of the top K principal component functions with the corresponding FPC scores as coefficients:

$$X(t) \approx \mu(t) + \sum_{k=1}^{K} A_k \phi_k(t) .$$

The proper value of K can be selected based on the fraction of explained variance, AIC (Akaike information criterion), or BIC (Bayesian information criterion) [128]. Estimating top principal component functions is equivalent to estimating $\Sigma(t, t')$ with low-rank structure.

There are several FPCA studies incorporating covariates [130]–[134]. In general, these methods can be categorized into two classes. The first class of methods assumes that the mean and covariance functions are smooth functions of both time and covariates and uses kernel smoothers to estimate them. For instance, Jiang and Wang [133], assumed a single covariate per subject and applied a three-dimensional kernel smoother to obtain the estimation of the covariance function. The second class of methods pools all subjects to conduct a standard FPCA and fits a model between the estimated FPC scores and covariates. For instance, Li *et al.* [132] assumed that FPC scores vary linearly with covariates.

Another widely-studied FDA method is functional linear models (FLMs) [135]. FLMs are an extension of classical linear models whose responses or regressors may be functional. Here, we will focus on FLMs with functional responses $X(\cdot)$ and (multiple) scalar regressors $\boldsymbol{Z} = [Z_1, \dots, Z_M]$, also called the function-on-scalar model:

$$X(t) = \beta_0(t) + \sum_{m=1}^{M} \beta_m(t) Z_m,$$

where $\beta_m(\cdot)$ for $m = 0, \dots, M$ are smooth functions defined on $\mathcal{T}$. In the context of our problem, regressors $\boldsymbol{Z}$ can represent covariates of a subject. Various estimation methods such as a penalized least squares can be applied to estimate functional coefficients $\beta_m(\cdot)$ given $\boldsymbol{Z}$ and (noisy)

observations of $X(t)$ [135]. . Other possible extended models are the functional mixed-effects models by including random functions [136] or to the varying coefficient models (VCMs) by making the coefficients vary smoothly over covariates [137]. However, most of the existing methods based on FPCA or FLMs (i) require complete or dense observations, (ii) handle only single or low-dimensional covariates, (iii) assume all covariates are informative, or (iv) cannot decide which covariates are informative. To address these issues, this study aims to develop a unified model to incorporate the covariate information into the sparse functional data analysis. In particular, we model the covariate effects on the FPC score through the between-subject correlations and also develop an informative covariate identification algorithm.

5.2 Methodology

Figure 5.2 illustrates the proposed covariate dependent spares functional data analysis. The proposed method first models the variation within each subject through pooled FPCA (Section 5.2.1), and then models the variation between different subjects (Section 5.2.2). Through these two types of variations, we will be able to make predictions of a new subject (Section 5.2.3) and identify important covariates as well (Section 5.2.4).

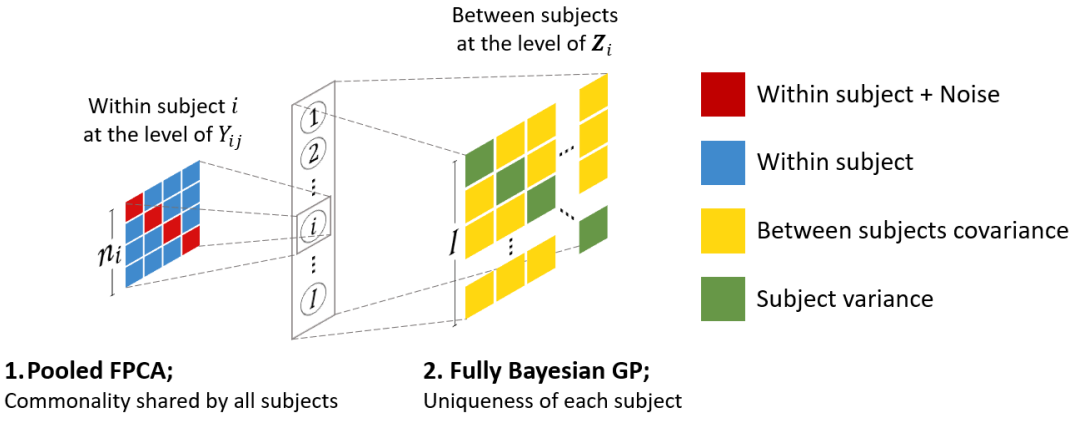


Figure 5.2 Illustration of the proposed framework

5.2.1 Base Model – Pooled FPCA

In this subsection, we will first review the classic pooled FPCA discussed in Section 5.1.2 in more details, which will lay a foundation of the proposed method. Suppose there are I subjects and n_i measurements for the ith subject. Let Y_{ij} be the jth measurement of the ith subject at a random point t_{ij}:

$$Y_{ij} = X_i(t_{ij}) + \varepsilon_{ij} \approx \mu(t_{ij}) + \sum_{k=1}^{K} A_{ik}\phi_k(t_{ij}) + \varepsilon_{ij} = \mu(t_{ij}) + \boldsymbol{A}_{(i)}^T \boldsymbol{\phi}(t_{ij}) + \varepsilon_{ij}, \qquad (5.2)$$

where $\mathbb{E}(\varepsilon_{ij}) = 0$, $var(\varepsilon_{ij}) = \sigma_\varepsilon^2$, $\boldsymbol{\phi}(t) = [\phi_1(t), \dots, \phi_K(t)]^T$ is a set of vectors derived from the eigenfunctions and $\boldsymbol{A}_{(i)} = [A_{i1}, \dots, A_{iK}]^T$ is a vector of the FPC scores unique to the ith subject.

As described in Section 5.1.2, the estimation of the eigencomponents (λ_k and $\phi_k(t)$) is straightforward once the mean and covariance of the functional data have been estimated. Motivated by Yao *et al*. [138], we pool data across subjects to overcome the sparse measurement issue and apply smoothing methods to estimate the mean function $\hat{\mu}(t)$ and covariance surfaces $\hat{\Sigma}(t, t')$. In particular, local linear smoothing proposed by [139] is adopted to estimate $\hat{\mu}(t)$; other smoothing methods can also be used. Fitting a local linear smoothing means minimizing

$$\sum_{i=1}^{I}\sum_{j=1}^{n_i} K_{LLS}(t, t_{ij})\{Y_{ij} - \mathcal{C}_0 - \mathcal{C}_1(t - t_{ij})\}^2$$

with respect to the coefficients $\mathcal{C}_0$ and $\mathcal{C}_1$, where K_{LLS} is a kernel function (often a Gaussian kernel function). The estimated mean function at t is $\hat{\mu}(t) = \hat{\mathcal{C}}_0$. Similarly, two-dimensional weighted least squares smoother is adopted to obtain the estimated covariance surfaces $\hat{\Sigma}(t, t')$. In particular, consider the covariance *within* subject i. Using eq. (5.2), $cov(Y_{ij}, Y_{il}) = cov\left(X_i(t_{ij}), X_i(t_{il})\right) + \sigma_\varepsilon^2\delta_{jl}$, where $\delta_{jl} = 1$ if $j = l$ and 0 otherwise. The "raw" covariances $G_i(t_{ij}, t_{il}) = \left(Y_{ij} -\right.$

$\hat{\mu}(t_{ij})\big)\big(Y_{il} - \hat{\mu}(t_{il})\big)$ is smoothed against (t_{ij}, t_{il}) using a nonparametric smoother, such as a local polynomial estimate [139]. Since $\mathbb{E}\left(G_i(t_{ij}, t_{il})\right) = cov\left(X_i(t_{ij}), X_i(t_{il})\right) + \sigma_\varepsilon^2 \delta_{jl}$, the diagonal raw covariances where $j = l$ are excluded in the estimation of the covariance function since these include an additional term σ_ε^2 due to the variance of the measurement errors. In fact, once $\hat{\Sigma}(t, t')$ is obtained, σ_ε^2 can be easily calculated by smoothing $Y_{ij} - \hat{\mu}(t_{ij})^2 - \hat{\Sigma}(t_{ij}, t_{ij})$ against t_{ij}, for instance, through a widely-used local linear smoother [139]. Estimation of the eigencomponents are then obtained using the eigen-equations,

$$\int_{\mathcal{T}} \hat{\Sigma}(t, t')\hat{\phi}_k(t)dt = \hat{\lambda}_k \hat{\phi}_k(t), \tag{5.3}$$

where $\hat{\phi}_k$ is subject to $\int_{\mathcal{T}} \hat{\phi}_k(t)^2 dt = 1$ and $\int_{\mathcal{T}} \hat{\phi}_k(t) \times \hat{\phi}_{k'}(t)dt = 0$ for $k < k'$. Following the existing literature, we solve the eigen-equations by discretizing the smoothed covariance [140]. Let $\boldsymbol{\Omega}_{FPCA}$ be the set of components of FPCA, i.e., $\boldsymbol{\Omega}_{FPCA} = \{\mu(t), \phi_1(t), \dots, \phi_K(t), \lambda_1, \dots, \lambda_K, \sigma_\varepsilon^2\}$.

In general, the FPC scores of each subject are estimated by numerical integration based on the definition $A_k = \int_{\mathcal{T}} \{X(t) - \mu(t)\}\phi_k(t)dt$. Nevertheless, this approach does not work well with sparse functional data. One of the most widely-used methods to overcome this is by assuming that A_{ik} follows a Gaussian distribution and estimating the FPC scores using the conditional expectation given that there is at least one measurement from the subject of interest [138]: $\hat{A}_{ik} = \mathbb{E}[A_{ik}|\boldsymbol{Y}_i] = \hat{\lambda}_k \widehat{\boldsymbol{\phi}}_{ik}^T \widehat{\boldsymbol{\Sigma}}_{Y_i}^{-1}(\boldsymbol{Y}_i - \widehat{\boldsymbol{\mu}}_i)$, where $\widehat{\boldsymbol{\phi}}_{ik} = \left[\hat{\phi}_k(t_{i1}), \dots, \hat{\phi}_k(t_{in_i})\right]^T$, $\boldsymbol{Y}_i = \left[Y_{i1}, \dots, Y_{in_i}\right]^T$, $\widehat{\boldsymbol{\mu}}_i = \left[\hat{\mu}(t_{i1}), \dots, \hat{\mu}(t_{in_i})\right]^T$, and the (j, l)th entry of $\widehat{\boldsymbol{\Sigma}}_{Y_i}$ is $\hat{\Sigma}(t_{ij}, t_{il})$. However, in this existing approach, the covariate information is not leveraged to estimate FPC scores of each subject, and the estimation is not accurate enough when the subject of interest does not collect any

measurements. In the next subsection, we propose a novel approach to overcome this limitation by incorporating the covariate information into the estimation of FPC scores.

5.2.2 Encoding Covariate Information into Kernel Design

The FPC scores A_{ik} and A_{rk} for the ith and rth subjects are commonly assumed to be independent for $i \neq r$. However, in many applications, the variables $\{A_{ik}\}_{i=1}^{I}$ may have an underlying relationship related to subjects' covariate information. For example, if two ferritic steel components have similar alloy compositions, they are likely to show similar swelling processes. On the contrary, if two ferritic steel components have significantly different alloy compositions, their swelling processes are likely to vary. The covariates of the ith subject are denoted by $\boldsymbol{Z}_i = [Z_{i1}, \dots, Z_{iM}]^T$, where M is the total number of covariates of each subject. In such models, it is reasonable to assume that $cov(A_{ik}, A_{rk})$ depends on the covariate difference between the ith and rth subjects, $\|\boldsymbol{Z}_i - \boldsymbol{Z}_r\|$, where $\|\boldsymbol{u}\|$ is the magnitude of a vector $\boldsymbol{u}$ (given that the covariates have been normalized to the same range). Such a priori knowledge is not taken into account in the conventional FPC score estimation described in Section 5.2.1.

The proposed framework encodes such covariate information into the estimation of FPC scores by imposing K independent zero-mean Gaussian processes (GPs) on $A_k, k = 1, \dots, K$:

$$A_k(\boldsymbol{Z}) \sim \mathcal{GP}\big(0, K_k(\boldsymbol{Z}, \boldsymbol{Z}')\big), \tag{5.4}$$

where $K_k(\boldsymbol{Z}, \boldsymbol{Z}')$ is the covariance function of the GP on A_k. Here, we choose the GP due to its great flexibility to establish the non-parametric relation, the interpolation capability at any covariate $\boldsymbol{Z}$, and the ability to quantify uncertainties. For a single subject i with covariates $\boldsymbol{Z}_i$, the prior distribution of A_{ik} follows a Gaussian distribution with variance $K_k(\boldsymbol{Z}_i, \boldsymbol{Z}_i)$. Note that this approach is similar to the existing studies relying on Gaussian assumption except that the prior

distribution is tailored to each subject based on its covariates. For any two subjects $i, r \in \{1, \cdots, I\}$, the covariance between A_{ik} and A_{rk} is now quantified by their covariate similarities, i.e., $cov(A_{ik}, A_{rk}) = K_k(\boldsymbol{Z}_i, \boldsymbol{Z}_r)$. Similar ideas of using GP to model FPC scores can be found in Chung *et al*. [141] for extrapolating longitudinal data and in Kim *et al*. [127] for modeling heterogeneous degrading subjects. Yet, these existing methods cannot be applied to our case. Although Chung *et al*. [141] can handle multi-sensor signals, it does not incorporate covariate information. Kim et al., [8] considers covariate information of heterogeneous subjects, yet cannot identify significant covariates. In addition, both methods rely on maximum likelihood estimation, which is unreliable in the cases with small number of subjects , highly sparse measurements per subject, and high-dimensional covariates. Our framework differs in that an explicit kernel design is considered to enable informative covariate identification, a fully Bayesian scheme is established to obtain a more robust and stable estimation, and the estimated eigenvalues are incorporated such that FPCA and GP are conducted in a unified manner.

Note that $cov(A_{ik}, A_{rk'})$ is always 0 when $k \neq k'$. In fact, this is the reason why we can use K independent univariate GPs instead of a single multivariate GP to model $\boldsymbol{A}_{(i)}$ together, which will greatly speed up the computations of the proposed method. It is also worth mentioning that unlike typical GPs modeling noisy observations, the additive noise term is not included in eq. (5.4) to encourage the variation of covariates $\boldsymbol{Z}_i$ rather than the noise variance to account for the variation of $\boldsymbol{A}_{(i)}$, and further enhance the informative covariate identification procedure which will be explained in Section 5.2.4. However, we can add a noise term to eq. (4) in applications where some important covariates are not recorded, and the extension is straightforward.

For the kth FPC score A_k, $K_k(\boldsymbol{Z}, \boldsymbol{Z}')$ is specified as the squared exponential covariance function with a *separate* scale parameter ρ_{km} for each covariate Z_{im}:

$$K_k(\boldsymbol{Z}_i, \boldsymbol{Z}_r) = \lambda_k \exp\left(-\frac{1}{2}\sum_{m=1}^{M}\frac{1}{\rho_{km}}(Z_{im} - Z_{rm})^2\right). \quad (5.5)$$

This kernel design is also well known as the automatic relevance determination (ARD) kernel [142], where the *characteristic length-scale* for the mth covariate is given by $\rho_{km}^{1/2}$. Ideally, if the mth covariate is irrelevant, the estimation of ρ_{km} should be large enough in order for the model to ignore this covariate, i.e., the difference between Z_{im} and Z_{rm} has negligible effects on the covariance between A_{ik} and A_{rk}. On the other hand, if ρ_{km} is small, A_{ik} will vary rapidly along the corresponding covariate, implying the high "relevance" of the mth covariate. Section 5.2.3 will discuss in detail how the fully Bayesian estimation approach is adopted to obtain the robust estimation of the unknown parameters ρ_{km} in eq. (5.5).

In the cases where subjects also have categorical covariates (e.g., irradiation type), we may apply one of the conventional transformations from categorical values to numerical values. Without loss of generality, suppose there are Q categorical covariates $\boldsymbol{U} = [U_1, \dots, U_Q]$, where the qth categorical covariate U_q has u_q levels, in addition to the continuous covariates $\boldsymbol{Z}$. Using the one-hot encoding, the qth categorical covariate of the ith subject is converted to a one-hot vector $\widetilde{\boldsymbol{U}}_{iq} = \left[U_{iq}^{(1)}, \dots, U_{iq}^{(u_q-1)}\right]$. If the ith subject belongs to the u_qth level of the qth categorical covariate, $\widetilde{\boldsymbol{U}}_{iq}$ is set to a zero vector; otherwise, all entries are zeros except the one corresponding to the qth categorical covariate of the ith subject. As a result, a transformed covariate vector of the ith subject will be $[\boldsymbol{Z}_i; \widetilde{\boldsymbol{U}}_{i1}; \dots; \widetilde{\boldsymbol{U}}_{iQ}]$. In applications where the differences across different categorical levels are assumed to be consistent, it is possible to further tie the length scales to be the same for different levels for ease of computation and extend the covariance function in eq. (5.5) as follows:

$$cov\left(A_k(\boldsymbol{Z}_i, \boldsymbol{U}_i), A_k(\boldsymbol{Z}_r, \boldsymbol{U}_r)\right)$$

$$= \lambda_k \exp\left(-\frac{1}{2}\sum_{m=1}^{M}\frac{(Z_{im}-Z_{rm})^2}{\rho_{km}} - \frac{1}{2}\sum_{q=1}^{Q}\frac{\sum_{u=1}^{u_q-1}\left(U_{iq}^{(u)}-U_{rq}^{(u)}\right)^2}{\rho'_{kq}}\right).$$

Another key part of the kernel design in eq. (5) is that the kth largest eigenvalues λ_k derived from FPCA acts as a *scale factor*. For the ith subject with covariates $\boldsymbol{Z}_i$, the variance of the prior distribution of the kth FPC score is reduced to λ_k, i.e., $var(A_{ik}) = K_k(\boldsymbol{Z}_i, \boldsymbol{Z}_i) = \lambda_k$, and thus the proposed design bridges the gap between FPCA in Section 5.2.1 and GP modeling in this subsection. This further resolves the unidentifiability issue in the ARD kernel that the length-scale parameters $\boldsymbol{\rho} = \{\boldsymbol{\rho}_k\}_{k=1}^K$, where $\boldsymbol{\rho}_k = [\rho_{k1}, \dots, \rho_{kM}]$, alone are not well identified, yet only the ratios of $\boldsymbol{\rho}_k$ to λ_k are identifiable [143].

Please note that, in practice, multiple covariates may be collinear. The collinearity makes the kernel matrix of the GP more ill-conditioned and leads to non-identifiability. Thus, we should remove the redundant or highly correlated covariates in data preprocessing to avoid the collinearity between different covariates.

To summarize, we first pool the covariance within a subject at the level of Y_{ij} to characterize the commonalities shared by all subjects and estimate $\mu(t)$, $\phi_k(t)$, λ_k and σ_ε^2. We then consider the covariance between subjects at the level of $\boldsymbol{Z}_i$ to characterize the uniqueness of each subject through its FPC scores $\boldsymbol{A}_{(i)}$ and model the covariate importance through $\boldsymbol{\rho}_k$. This is illustrated in Figure 5.2. It is important to note that while these two procedures are conducted separately, they are under the integrated structure where we use the variation of Y_{ij} to derive the variation from $\boldsymbol{Z}$ (between subjects) and the variation left conditioned on $\boldsymbol{Z}$ (within a subject). This is possible because each subject's trajectory is summarized through a set of FPC scores, and the covariate information only contributes to the between-subjects covariance of these scores.

5.2.3 Fully Bayesian Estimation and Prediction

In this subsection, we will discuss how to estimate the set of length-scale parameters $\boldsymbol{\rho}$ and make predictions of the function trajectories of the subjects of interest, i.e., estimate new subjects' FPC scores based on their covariates and measurements. The classical approach to estimate the parameters of the mean and covariance functions in GPs is by maximizing the marginal likelihood (also called Type II maximum likelihood) and yielding fixed point estimates [101], [127]. This method has several advantages. For example, the marginal likelihood is tractable for Gaussian noise models, and once the point estimation has been obtained, the predictions of a new data point can be computed analytically. However, this approach also has several major drawbacks when it is used in the proposed framework. First, it is unstable, i.e., the resulting point estimates of ρ_{km} may significantly vary for different parameter initializations, especially in the cases of a small number of subjects, a high sparsity, or high-dimensional covariates. It is also well known that the non-convexity of the marginal likelihood surface can make Type II maximum likelihood estimation tend to overfit [144]. Second, unlike most of the existing studies using the ARD kernel, the point estimates of ρ_{km} of irrelevant covariates often do not go to infinity or a significantly large value in the proposed framework which makes it much more challenging to identify important covariates. This is because we do not have direct realizations of the FPC scores $\boldsymbol{A}_{(i)}$; instead we only indirectly observe the linear combination of FPC scores of each subject through the sparse and noisy measurements as in eq. (2).

To address these issues and obtain more robust results, our method proposes the fully Bayesian hierarchical scheme as follows:

Prior over hyperparameters	$\rho_{km} \sim p(\rho_{km}),\ k = 1, \dots, K$ and $m = 1, \dots, M$
Prior over parameters	$\boldsymbol{A}_k \mid \boldsymbol{Z}, \boldsymbol{\rho}_k \sim N(0, \boldsymbol{K}_k), \quad k = 1, \dots, K$

Likelihood $\boldsymbol{Y}|\boldsymbol{A}, \boldsymbol{\Omega}_{FPCA} \sim N(\boldsymbol{\mu} + \boldsymbol{\Phi}\boldsymbol{A}, \sigma_\varepsilon^2 \mathbb{I})$

Specifically, following Neal [142], an inverse Gamma prior $p(\rho_{km})$ whose mean scales with the number of covariates M is imposed on ρ_{km} with the shape parameter α_ρ and the scale parameter $\beta_\rho M^{2/\alpha_\rho}$. An inverse gamma prior is chosen since it has a sharp left tail that assigns negligible mass on very small values and a heavy right tail that allows large values. Also, when the number of covariates M increases, the proportion of informative covariates are often expected to decrease, and thus, inverse Gamma prior whose mean scales with M is widely used [142], [145]. Let $\boldsymbol{A}_k = [A_{1k}, \dots, A_{Ik}]^T \in \mathbb{R}^{I\times 1}$ be a vector of the kth FPC scores, $\boldsymbol{A} = \left[\boldsymbol{A}_{(1)}; \cdots; \boldsymbol{A}_{(I)}\right] \in \mathbb{R}^{IK\times 1}$ be all the FPC scores, and $\boldsymbol{Z} = [\boldsymbol{Z}_1; \cdots; \boldsymbol{Z}_I] \in \mathbb{R}^{IM\times 1}$ be the covariates of all subjects. Denote $\boldsymbol{K}_k = \begin{bmatrix} K_k(\boldsymbol{Z}_1, \boldsymbol{Z}_1) & \cdots & K_k(\boldsymbol{Z}_1, \boldsymbol{Z}_I) \\ \vdots & \ddots & \vdots \\ K_k(\boldsymbol{Z}_I, \boldsymbol{Z}_1) & \cdots & K_k(\boldsymbol{Z}_I, \boldsymbol{Z}_I) \end{bmatrix} \in \mathbb{R}^{I\times I}$ to be the Gram matrix, $\boldsymbol{Y} = [\boldsymbol{Y}_1; \dots; \boldsymbol{Y}_I] \in \mathbb{R}^{(\Sigma n_i)\times 1}$ to be all measurements, and $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{\phi}_I \end{bmatrix} \in \mathbb{R}^{(\Sigma n_i)\times IK}$ to be the eigenfunction design matrix, where $\boldsymbol{\phi}_i = [\boldsymbol{\phi}_{i1}, \dots, \boldsymbol{\phi}_{iK}] \in \mathbb{R}^{n_i\times K}$.

Recall that our two main objectives are (i) identify informative covariates and (ii) model and predict the trajectory of a new subject. The first step to achieve both goals is the estimation of length-scale parameters $\boldsymbol{\rho}$. Under such Bayesian hierarchical scheme, we use a self-tuning variant of the Hamiltonian Monte Carlo (HMC) called the No-U-Turn-Sampler (NUTS) to estimate the posterior distribution of the unknown parameters $\boldsymbol{\rho}$ [146]. The performance of a standard HMC is highly sensitive to two tuning parameters: a step size and a number of steps. Several preliminary runs are required to find the proper values of these parameters. Alternatively, the NUTS develops a recursive algorithm that adaptively tunes the parameters. Empirically, the NUTS is shown to

work at least as good as a well-tuned standard HMC. The implementation of the NUTS in this study uses a probabilistic programming language (Stan).

The xext step is to make the predictions for a new subject with covariate information $\boldsymbol{Z}^{\star}$ based on the parameter estimation results. To achieve this, we integrate over the joint posterior:

$$p(\boldsymbol{A}^{\star}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{Z}^{\star}) = \int\int p(\boldsymbol{A}^{\star}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{Z}^{\star},\boldsymbol{A},\boldsymbol{\rho})p(\boldsymbol{A}|\boldsymbol{\rho},\boldsymbol{Y},\boldsymbol{Z})p(\boldsymbol{\rho}|\boldsymbol{Y},\boldsymbol{Z})d\boldsymbol{A}d\boldsymbol{\rho}\,,$$

where we have suppressed the conditioning over $\boldsymbol{\Omega}_{FPCA}$ for brevity. Given that the joint prior distribution of $\boldsymbol{A}^{\star}$ and $\boldsymbol{A}$ is Gaussian distribution and $p(\boldsymbol{A}^{\star}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{Z}^{\star},\boldsymbol{A},\boldsymbol{\rho})$ and $p(\boldsymbol{A}|\boldsymbol{\rho},\boldsymbol{Y},\boldsymbol{Z})$ are Gaussian distributions as well, the above can be further simplified to

$$p(\boldsymbol{A}^{\star}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{Z}^{*}) = \int p(\boldsymbol{A}^{\star}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{Z}^{\star},\boldsymbol{\rho})p(\boldsymbol{\rho}|\boldsymbol{Y},\boldsymbol{Z})d\boldsymbol{\rho}\,, \tag{5.6}$$

where $\boldsymbol{A}^{\star}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{Z}^{\star},\boldsymbol{\rho} \sim N(\boldsymbol{\mu}^{\star},\boldsymbol{\Sigma}^{\star})$ with parameters,

$$\begin{aligned}\boldsymbol{\mu}^{\star} &= \boldsymbol{K}^{\star}\boldsymbol{K}^{-1}\left(\frac{1}{\sigma_{\varepsilon}^{2}}\boldsymbol{\Phi}^{T}\boldsymbol{\Phi}+\boldsymbol{K}^{-1}\right)^{-1}\frac{1}{\sigma_{\varepsilon}^{2}}\boldsymbol{\Phi}^{T}(\boldsymbol{Y}-\boldsymbol{\mu}),\text{and}\\ \boldsymbol{\Sigma}^{\star} &= \boldsymbol{K}^{\star}\boldsymbol{K}^{-1}\left(\frac{1}{\sigma_{\varepsilon}^{2}}\boldsymbol{\Phi}^{T}\boldsymbol{\Phi}+\boldsymbol{K}^{-1}\right)^{-1}\boldsymbol{K}^{-1}\boldsymbol{K}^{\star T}+\boldsymbol{K}^{\star\star}-\boldsymbol{K}^{\star}\boldsymbol{K}^{-1}\boldsymbol{K}^{\star T}.\end{aligned} \tag{5.7}$$

$\boldsymbol{K}^{\star\star} = \boldsymbol{K}(\boldsymbol{Z}^{\star},\boldsymbol{Z}^{\star})$ denotes the covariance matrix of $\boldsymbol{Z}^{\star}$ and $\boldsymbol{K}^{\star} = [\boldsymbol{K}(\boldsymbol{Z}_1,\boldsymbol{Z}^{\star}),\dots,\boldsymbol{K}(\boldsymbol{Z}_I,\boldsymbol{Z}^{\star})]$ denotes that between $\boldsymbol{Z}^{\star}$ and $\boldsymbol{Z}$, where $\boldsymbol{K}(\boldsymbol{Z},\boldsymbol{Z}')$ is a diagonal matrix with diagonal entries $K_k(\boldsymbol{Z},\boldsymbol{Z}'), k = 1,\dots,K$. The proof of (7) is given in the Appendix. The predictive distribution in eq. (6) can be approximated with Monte Carlo integration as follows.

$$\begin{aligned}p(\boldsymbol{A}^{\star}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{Z}^{\star}) &= \int p(\boldsymbol{A}^{\star}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{Z}^{\star},\boldsymbol{\rho})p(\boldsymbol{\rho}|\boldsymbol{Y},\boldsymbol{Z})d\boldsymbol{\rho}\\ &\approx \frac{1}{H}\sum\nolimits_{h=1}^{H} p\left(\boldsymbol{A}^{\star}\middle|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{Z}^{\star},\boldsymbol{\rho}_{(h)}\right), \qquad \boldsymbol{\rho}_{(h)}\sim p(\boldsymbol{\rho}|\boldsymbol{Y},\boldsymbol{Z}),\end{aligned} \tag{5.8}$$

where $\boldsymbol{A}$ is integrated out analytically and $\boldsymbol{\rho}_{(h)}$ is a random draw $\boldsymbol{\rho}_{(h)}$ from the posterior distribution $p(\boldsymbol{\rho}|\boldsymbol{Y},\boldsymbol{Z})$ obtained through the NUTS. We can then obtain the approximate posterior distribution of $Y^\star$ using $Y^\star = \hat{\mu}(t^\star) + \sum_{k=1}^{K} A_k^\star \hat{\phi}_k(t^\star)$.

When the subjects of interest have collected at least one measurement, i.e., for the estimation of $\boldsymbol{A}$, a similar procedure can be carried out by using $p(\boldsymbol{A}|\boldsymbol{Y},\boldsymbol{Z}) = \int p(\boldsymbol{A}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{\rho})p(\boldsymbol{\rho}|\boldsymbol{Y},\boldsymbol{Z})d\boldsymbol{\rho}$, where $p(\boldsymbol{A}|\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{\rho}) = N\left(\boldsymbol{\mu}^{\star(2)},\boldsymbol{\Sigma}^{\star(2)}\right)$ with fixed parameters,

$$\boldsymbol{\mu}^{\star(2)} = \frac{1}{\sigma_\varepsilon^2}\boldsymbol{\Sigma}^{\star(2)}\boldsymbol{\Phi}^T(\boldsymbol{Y}-\boldsymbol{\mu}), \text{and}$$

$$\boldsymbol{\Sigma}^{\star(2)} = \left(\frac{1}{\sigma_\varepsilon^2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{K}^{-1}\right)^{-1}. \tag{5.9}$$

The details are given in the Appendix. Using the marginalization property of Gaussian distribution, the posterior distribution of the ith subject can be obtained by extracting the ith subvector of the mean vector and the ith submatrix of the covariance matrix of the posterior distribution of $\boldsymbol{A}$.

5.2.4 Informative Covariate Identification

The proposed method assumes that there might be non-informative covariates that do not affect the functional trajectories. In fact, in many applications, it may be unknown which covariate significantly affects a subject's trajectory. For instance, in void swelling, identifying important alloy composition is still an area of active research. Section 5.2.2 introduces how the ARD kernel can be used to measure the importance of each covariate. In this section, we will explain the limitation of the naïve implementation of the ARD kernel in our context and discuss a more systematic approach to identify informative covariates.

In the existing literature on the ARD kernel, it often sets a threshold value ξ and decides that the mth covariate is informative if $\rho_{km} < \xi$. In other words, we test the null hypothesis H_0 that a

covariate is non-informative and reject H_0 if ρ_{km} is in the critical region, i.e., $\rho_{km} < \xi$. Nevertheless, as mentioned in Section 5.2.3, the estimation of ρ_{km} of irrelevant covariates in the proposed method does not always tend to infinity or very large values due to the sparse and noisy measurements. This makes it much more difficult to choose a proper value of ξ. In addition, the distribution of ρ_{km} under the null hypothesis cannot be analytically derived. To tackle this problem, the proposed method first introduces a new inert covariate that has no effect on the response. The posterior draws of the inert covariate obtained in Section 5.2.3 are then used to estimate the shape of the probability density function (pdf) of a covariate under H_0 and numerically obtain the critical region. Similar idea of adding an inert factor (also called "pseudo-variable") and using its posterior distribution as a reference has been explored in different contexts [147]–[149], yet not in the context of FPCA and the explicit kernel design in Section 5.2.2.

In particular, an additional covariate $\tilde{Z}_{i(M+1)}$ that has no impact on the response is appended, i.e., $\tilde{\boldsymbol{Z}}_i = \left[Z_{i1}, \dots, Z_{iM}, \tilde{Z}_{i(M+1)}\right]^T$. To mimic the existing covariates, $\tilde{Z}_{i(M+1)}$ also ranges from 0 to 1 (given that the original covariates have been normalized to this range). An additional binary inert covariate can be used to accommodate one-hot encoded categorical covariates. The ideal choice of $\tilde{Z}_{i(M+1)}$ is that the added column $\boldsymbol{Z}_{M+1} = \left[\tilde{Z}_{1(M+1)}, \dots, \tilde{Z}_{I(M+1)}\right]^T$ is orthogonal to existing columns $\boldsymbol{Z}_m = [Z_{1m}, \dots, Z_{Im}]^T, m = 1, \dots, M$. Yet, in practice, such a column $\boldsymbol{Z}_{M+1}$ may not exist. Instead, we randomly sample $\boldsymbol{Z}_{M+1}$ and perform the following informative covariate identification procedure multiple times to obtain robust results. Ideally, $\boldsymbol{Z}_{M+1}$ should be sampled to mimic the existing covariates' behaviors. In cases where the true distributions of existing covariates are unknown, an inert covariate can be sampled from $Uniform(0,1)$ given that each covariate is standardized to range between 0 and 1 (or sampled from the Bernoulli distribution for one-hot encoded categorical covariates), or by empirically estimating the probability functions of existing

covariates (e.g., kernel density estimation) [150]. In each trial, $\boldsymbol{Z}_{M+1}$ is added, and the NUTS is implemented to calculate the posterior distributions of the following significance score s_m of all covariates:

$$s_m = \sum_{k=1}^{K} \lambda_k \exp\left(\frac{1}{\rho_{km}}\right), \quad m = 1, \dots, M+1.$$

In general, the importance of a covariate is measured using the inverse of ρ_{km} as explained in Section 5.2.2. In the proposed framework, the covariate is important when the corresponding principal component function accounts for a larger variation (larger λ_k) or the function varies more rapidly according to the corresponding covariate changes (smaller ρ_{km}). Thus, the above significance score is proposed which is the weighted sum of $\exp\left(\frac{1}{\rho_{km}}\right), k = 1, \dots, K$ with the eigenvalue λ_k as the weight. As a result, a covariate with a high significance score s_m is considered as an important covariate. In each trial of the augmentation of an inert covariate and the NUTS, we record the posterior median of s_{M+1} and every realization of $s_m, m = 1, \dots, M$. As a result, posterior medians of s_{M+1} represent the behaviors of s_m under the null hypothesis. The posterior median over all realizations of s_m can be compared to the reference distribution of the posterior median of s_{M+1}. Note that we can run multiple trials in parallel if needed to reduce computational time. One major advantage of adding a new inert covariate as a reference is that it allows us to control the false positive rate and removes the need to find a proper value of a threshold value ξ. For instance, if s_m is compared to the 95$^{\text{th}}$ percentile of the reference distribution of s_{M+1}, it can be said that there is a 5% chance of falsely identifying a non-informative covariate as informative. Alternatively, we can also use the Monte Carlo estimation of significance and calculate the p-value as the relative ranking of the posterior median of s_m among the samples of the posterior median of s_{M+1}. It is also worth pointing out that a sufficient number of the trials of

the augmentation of an inert covariate and the NUTS is crucial, especially in the estimation of the high quantile values [151]. In the following numerical studies, we observed that more than 200 trials are enough.

In practice, it is possible that we also want to screen out covariates with negligible effects. In such case, we set $\tilde{Z}_{i(M+1)}$ to have small effects on the response instead of no effects. For instance, we can use the adjusted observation as $\tilde{Y}_{ij} = Y_{ij} + \eta \times \lambda_1 \phi_1(t_{ij}) \tilde{Z}_{i(M+1)}$ with a small incremental positive value η.

5.3 Simulation Study

In this section, a series of numerical studies are performed to demonstrate the effectiveness and the sensitivity of the proposed method using simulated covariate-dependent sparse functional data. Section 5.3.1 introduces how we generate the simulated dataset. Section 5.3.2 evaluates the informative covariate identification and prediction performance of the proposed method. In Section 5.3.3, we confirm that the addition of an inert covariate does not significantly change the posterior medians of the existing covariates. From Section 5.3.4 to Section 5.3.7, we evaluate how the proposed method performs under different scenarios, including a covariate with small effects, low heterogeneity between different subjects, a categorical covariate with additive/multiplicative effects, and different prior distributions over the hyperparameters. Additional simulation study results analyzing the sensitivity of the proposed method to the mis-specification of the inert covariate are also presented in the supplemental materials.

5.3.1 Data Generation

The simulated trajectories have mean function $\mu(t) = t + 10\exp(-(t-5)^2)$ and covariance function derived from two eigenfunctions, $\phi_1(t) = \cos(\pi t/5)/\sqrt{5}$ and $\phi_2(t) = -\sin(\pi t/5)/\sqrt{5}$ for $0 \le t \le 10$. Each subject has a total of $M = 20$ covariates. The covariates of all subjects, $[\boldsymbol{Z}_1, \cdots, \boldsymbol{Z}_I]$, is drawn as a $I \times M$ Latin hypercube sample (LHS) matrix from a set of $Uniform(0,1)$. The FPC scores of the ith subject are generated using $A_{i1} = 0.5 \times Z_{i1}^2 + 0.5 \times Z_{i2}$ and $A_{i2} = 0.5 \times \sin(2 \times Z_{i1}) + 0.5 \times Z_{i2}^2$, i.e., only the first two covariates out of a total of 20 are informative. The calculated FPC scores are then normalized to have mean 0 and variance (eigenvalues) $\lambda_1 = 5$ and $\lambda_2 = 2$. The variance of the measurement error is $\sigma_\varepsilon^2 = 0.5^2$. For an equally spaced grid $\{c_0, \ldots, c_{50}\}$ on $[0,10]$ with $c_0 = 0$ and $c_{50} = 10$, let $d_l = c_l + e_l$, where $e_l \overset{\text{i.i.d}}{\sim} N(0,1)$, $d_l = 0$ if $d_l < 0$ and $d_l = 10$ if $d_l > 10$ to create "jittered" grid. In the ith subject, the number of measurements n_i is randomly sampled from a discrete uniform distribution on $\{2, \ldots, 5\}$, and the locations of measurements are randomly sampled from $\{d_1, \ldots, d_{50}\}$ without replacement. Figure 5.3 shows an example of the trajectories of 50 randomly generated subjects.

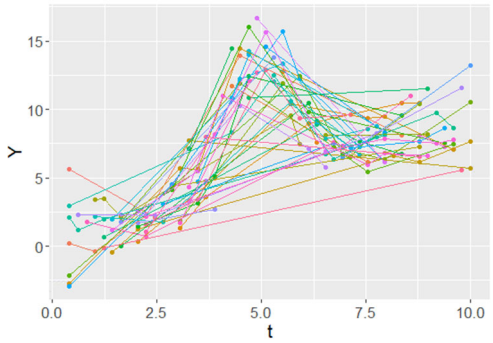


Figure 5.3 Realizations of 50 subjects

5.3.2 Baseline scenario

First, the trajectories of $I = 50$ subjects are generated to verify the informative covariate identification performance. Throughout Sections 5.3.2 to 5.3.6, the parameters of the prior distribution of $\boldsymbol{\rho}$ are set to $\alpha_\rho = 4$ and $\beta_\rho = 1$. The 600 iterations of the NUTS are run to generate posterior realizations of $\boldsymbol{\rho}$, with the first 500 discarded as burn-in. The augmentation of an inert covariate and the NUTS are repeated 300 times. The resulting 300 posterior medians of s_{M+1} form the reference distribution. Figure 5.4 illustrates the boxplots of the posterior realizations of s_m from a random simulation where the horizontal solid (dashed) line is the 95th (90th) percentile of the reference distribution. The figure shows that the first two covariates are correctly identified as informative and clearly have higher significance scores than the other non-informative covariates. The above simulation is repeated 100 times, and Table 5.1 summarizes the results. The covariates are identified as informative using either the 95th or 90th percentiles of the reference distribution as a criterion. The table shows that our method correctly identifies the informative covariates with very high probability. Note that although it is possible to approximately control the false positive rate through the percentiles of the inert covariate, a limited number of subjects or the high sparsity of the measurements per subject can result in a higher false positive rate as illustrated in Table 5.1. We further conduct the informative covariate identification with a smaller number of the total subjects *I*. The simulation is repeated 50 times for each value of *I*. Table 5.2 summarizes that with more subjects, i.e., as *I* increases, the probability of correctly identifying informative covariates

Table 5.1 Proportion of simulations that each covariate is identified as informative (Truly informative covariates are highlighted in bold)

	Covariate																			
Percentile	**1**	**2**	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
90th	**1**	**1**	0.17	0.18	0.20	0.16	0.09	0.16	0.11	0.07	0.13	0.05	0.13	0.13	0.13	0.12	0.14	0.17	0.08	0.15
95th	**0.99**	**0.98**	0.10	0.06	0.13	0.12	0.05	0.11	0.05	0.08	0.07	0.02	0.04	0.05	0.04	0.03	0.10	0.13	0.08	0.09

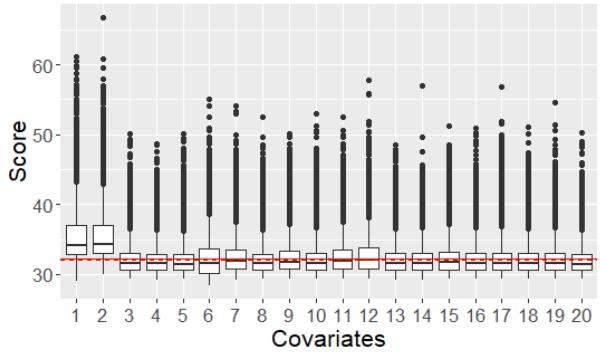


Figure 5.4 Posterior distributions of s_m for a randomly selected simulation. Horizontal solid (dashed) line: the 95th (90th) percentile of the reference distribution

Table 5.2 Proportion of simulations that an informative covariate is correctly identified with small I

	I			
	10		20	
	Covariate			
Percentile	1	2	1	2
90th	0.50	0.42	1	1
95th	0.48	0.36	0.98	1

(the power of the hypothesis test) increases. With more than or equal to only 20 subjects, two informative covariates are always correctly identified as informative.

To assess the prediction performance, we generate I training subjects and 10 testing subjects. For the training subjects, both covariates and sparse measurements are available where n_i is randomly sampled from a discrete uniform distribution on $\{5, \dots, 10\}$. Our goal is to accurately estimate their trajectories. We calculate the mean squared fitting errors (MSFE) of testing subjects: $\sum_{r=1}^{10}\sum_{l=1}^{50}\left(X_r(c_l) - \hat{X}_r(c_l)\right)^2 /50 /10$. The following benchmark methods are used:

(1) Conventional GP with the ARD kernel, which takes Y_{ij} as outputs and $[t_{ij};\ \boldsymbol{Z}_i]$ as inputs

(2) Varying-coefficient model (VCM) that semi-parametrically approximates the effects of the covariates [152]

(3) CD-FPCA which is the FPCA with covariate-dependent mean and covariance structure (assuming informative covariates are known as a *priori*) [134]

(4) PCA through conditional expectation (PACE) for sparse longitudinal data [138]

While the proposed method uses the GP to model the relation between the covariates and the FPCA scores, another possible approach is by locally fitting a line to estimate $\boldsymbol{A}_{(i)}$. This corresponds to the VCM whose coefficient is the FPC score $\boldsymbol{A}_{(i)}$ which varies according to the covariates $\boldsymbol{Z}_i$ [31]. We also consider two FPCA-based methods: the PACE [138] and the CD-FPCA [134]. As discussed in Section 5.2.1, the PACE is a widely-used pooled FPCA method that pools measurements from all subjects to address the sparsity issue and uses smoothing methods to estimate the mean and covariance functions. Note that the PACE does not consider the covariates of subjects, i.e., all subjects are assumed to be identical. Alternatively, the CD-FPCA can handle covariate information and assumes that mean and covariance functions vary according to the covariates. It is important to point out that the CD-FPCA cannot identify informative covariates and is developed for univariate or low-dimensional covariates scenarios. Thus, the following results for the CD-FPCA is the ones assuming informative covariates are known as priori. The existing methods requiring complete or dense observations are not included. For instance, Supervised Sparse and Functional PCA (SupSFPC) method proposed by Li, Shen, and Huang [132] handles the sparsity in a principal component loading vector and cannot handle sparse and irregularly spaced measurements, and thus not considered as a benchmark.

The simulations are repeated 50 times for each value of I. We first assume only the covariates are available for the testing subjects (i.e., no measurements available). Since both the PACE and the CD-FPCA require at least one measurement from a testing subject to make estimation of its FPC scores, the estimated (covariate-adjusted) mean function is used as the predictions of the PACE and the CD-FPCA. The results are shown in Table 5.4 that the proposed method yields much lower fitting errors than the benchmark methods. Unsurprisingly, the prediction accuracy

Table 5.3 Mean and standard deviation (in parentheses) of mean squared fitting errors when no observations are available for testing subjects

I	Proposed	GP	VCM	CD-FPCA	PACE
20	0.258 (0.158)	0.734 (1.068)	2.018 (0.981)	18.72 (1.570)	3.116 (1.084)
40	0.213 (0.109)	0.415 (0.398)	0.914 (0.603)	18.76 (1.967)	3.009 (0.998)
60	0.167 (0.085)	0.391 (0.299)	0.324 (0.315)	18.38 (0.568)	3.054 (1.028)

Table 5.4 Mean and standard deviation (in parentheses) of mean squared fitting errors when two observations are available for each testing subject

I	Proposed	GP	VCM	CD-FPCA	PACE
20	0.241 (0.144)	0.401 (0.198)	1.574 (1.420)	11.94 (2.971)	23.50 (12.67)
40	0.197 (0.114)	0.321 (0.214)	0.710 (0.813)	9.885 (1.609)	2.257 (4.916)
60	0.158 (0.089)	0.299 (0.201)	0.229 (0.282)	9.451 (1.567)	1.038 (0.368)

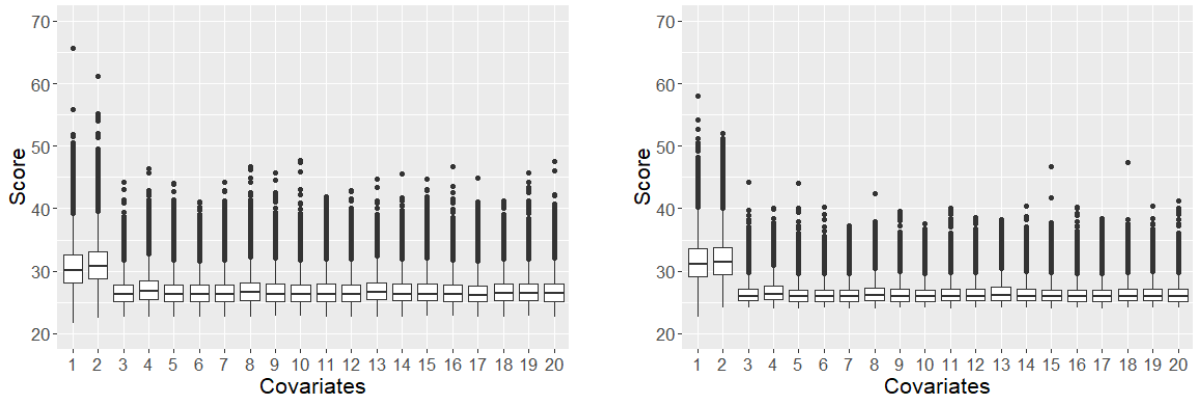
improves as more training subjects are available. One interesting result is the fitting errors of the CD-FPCA. Although the CD-FPCA considers covariate information unlike the PACE, the CD-FPCA results in much higher errors than the PACE. This is mainly because the available dataset is extremely limited, i.e., only 20~60 subjects with very sparse measurements, and the CD-FPCA largely overfitted the data. We further simulate the case where each testing subject has collected two measurements and Table 5.4 illustrates the results. Similar to Table 5.3, Table 5.4 shows that the proposed method outperforms the benchmark methods.

5.3.3 Addition of inert covariates

Recall that the posterior medians are used in identifying informative covariates. Thus, it is crucial to assess how the posterior medians of significance scores change as we add an inert covariate. To illustrate this, the NUTS is repeated on the same dataset with and without the addition of an inert covariate. Figure 5.5 (a) and (b) illustrate the boxplots of the posterior realizations of s_m of existing covariates (a) when an inert covariate is added and (b) when it is not. Two figures are very similar, indicating that the addition of an inert covariate does not significantly change the

Table 5.5 Proportion of simulations that each covariate is identified as informative when the third informative covariate has much smaller effects than the first and second informative covariates (Truly informative covariates are highlighted in bold)

	Covariate																			
Percentile	**1**	**2**	**3**	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
90th	**1**	**1**	**0.12**	0.10	0.08	0.08	0.14	0.08	0.10	0.12	0.12	0.14	0.12	0.12	0.08	0.06	0.12	0.08	0.16	0.08
95th	**1**	**1**	**0.06**	0.08	0.06	0.08	0.12	0.06	0.08	0.10	0.10	0.10	0.08	0.10	0.06	0.02	0.04	0.02	0.12	0.04



(a) (b)

Figure 5.5 Posterior distributions of s_m (a) when an inert covariate is added and (b) when it is not for a randomly selected simulation.

posterior medians of s_m. The average percentage errors of the posterior medians between two scenarios are 1.58%, indicating that the addition of an inert covariate has negligible effects on the posterior estimations of existing covariates.

5.3.4 Covariates with small effects

In this sub subsection, a new informative covariate is added, i.e., the FPC scores of the ith subject are generated using $A_{i1} = 0.5 \times Z_{i1}^2 + 0.5 \times Z_{i2}$ and $A_{i2} = 0.5 \times \sin(2 \times Z_{i1}) + 0.5 \times Z_{i2}^2 + 0.05 \times Z_{i3}$, i.e., the first three covariates are informative. Compared to Section 5.3.2, the third covariate now has very small effects on the second FPC score, i.e., insignificant effects on the measurements. The covariate identification simulation is repeated 50 times, where each simulation follows the same procedure as in Section 5.3.2. Table 5.5 summarizes the results. The

table shows that compared to the first and second covariates with much more significant effects, the probability of correctly identifying the third covariate as informative is much smaller. In other words, when the large variations of a covariate merely result in negligible heterogeneity of the trajectories, the probability of identifying this covariate as informative is low.

5.3.5 Low heterogeneity

In this subsection, a group of less heterogenous subjects is generated. Specifically, eigenvalues are set to $\lambda_1 = 1^2$ and $\lambda_2 = 0.5^2$. The simulation is repeated 50 times where each simulation follows the same procedure as in Section 5.3.2. The informative covariate identification results are in

Table 5.6. The table shows that the probability of correctly identifying informative covariates (the power of the hypothesis test) decreases, and that of incorrectly identifying non-informative covariates as informative increases as the heterogeneity decreases. This is because as the eigenvalue decreases, the trajectories of different subjects become more similar given the same covariate differences and make it more difficult to detect the covariate effects. We also observe that in more extreme cases where the eigenvalues get similar to or smaller than the noise variance σ_ε^2, the probability of correctly identifying an informative covariate rapidly decreases, close to the probability of falsely identifying a non-informative covariate as informative, which indicates the

Table 5.6 Proportion of simulations that each covariate is identified as informative when $\lambda_1 = 1^2$ and $\lambda_2 = 0.5^2$ (Truly informative covariates are highlighted in bold)

	Covariate																			
Percentile	**1**	**2**	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
90th	**0.86**	**0.88**	0.18	0.20	0.24	0.16	0.12	0.16	0.14	0.14	0.16	0.18	0.16	0.20	0.22	0.14	0.20	0.16	0.16	0.18
95th	**0.86**	**0.88**	0.14	0.16	0.22	0.16	0.10	0.12	0.10	0.12	0.16	0.12	0.12	0.20	0.20	0.12	0.18	0.14	0.12	0.14

application condition of the proposed method. However, due to the page limit, we omit the results here.

5.3.6 Categorical covariates

In this subsection, a new informative categorical covariate is added. Specifically, two cases are designed: a categorical covariate with an additive effect on FPC scores and a categorical covariate with a multiplicative effect on FPC scores. We assume there are 19 continuous covariates and one categorical covariate with two categories, resulting in $M = 20$ after applying the one-hot encoding. First, the categorical covariate has an additive effect on FPC scores, i.e., the FPC scores of the ith subject are generated using $A_{i1} = 0.5 \times Z_{i1}^2 + 0.5 \times Z_{i2} + 0.5 \times \mathbb{I}(U_{i1})$ and $A_{i2} = 0.5 \times \sin(2 \times Z_{i1}) + 0.5 \times Z_{i2}^2 + 0.1 \times \mathbb{I}(U_{i1})$, where $\mathbb{I}(U_{i1}) = 1$ if the ith subject belongs to the first category and 0 otherwise, resulting $\mathbb{E}\big(\mathbb{I}(U_{i1})\big) = 0.5$. The covariate identification simulation is repeated 50 times, where each simulation follows the same procedure as in Section 5.3.2, except that two inert covariates—continuous (generated from $Uniform(0,1)$) and categorical (generated from $Bernoulli(1/2)$)—are augmented. Table 5.7 summarizes that the two informative continuous covariates (Covariates 1 and 2) and one binary covariate corresponding to the

Table 5.7 Proportion of simulations that each covariate is identified as informative when a categorical covariate with an additive effect is added (Truly informative covariates are highlighted in bold)

	Covariate																			
Percentile	**1**	**2**	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	**20**
90th	**1**	**1**	0.16	0.12	0.12	0.10	0.14	0.12	0.08	0.10	0.12	0.10	0.16	0.12	0.14	0.16	0.16	0.12	0.14	**1**
95th	**1**	**1**	0.10	0.04	0.06	0.02	0.06	0.08	0.02	0.06	0.08	0.06	0.10	0.06	0.08	0.10	0.10	0.04	0.06	**1**

Table 5.8 Proportion of simulations that each covariate is identified as informative when a categorical covariate with a multiplicative effect is added (Truly informative covariates are highlighted in bold)

	Covariate																			
Percentile	**1**	**2**	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	**20**
90th	**1**	**1**	0.14	0.16	0.12	0.14	0.10	0.08	0.12	0.14	0.12	0.12	0.10	0.10	0.14	0.16	0.08	0.14	0.12	**1**
95th	**1**	**1**	0.06	0.10	0.08	0.10	0.06	0.02	0.06	0.06	0.08	0.06	0.02	0.04	0.06	0.10	0.04	0.06	0.06	**1**

informative categorical covariate (Covariate 20) are correctly identified as informative with very high probability. Second, the categorical covariate has a multiplicative effect on FPC scores, i.e., the FPC scores of the ith subject are $A_{i1} = (0.5 \times Z_{i1}^2 + 0.5 \times Z_{i2}) \times (\mathbb{I}(U_{i1}) + 1)$ and $A_{i2} = 0.5 \times \sin(2 \times Z_{i1}) + 0.5 \times Z_{i2}^2$. The results are shown in Table 5.8 that the proposed method successfully identifies all the informative continuous and categorical covariates with high probability.

5.3.7 Sensitivity to prior distribution over $\boldsymbol{\rho}$

In this subsection, we assess the sensitivity of the proposed method against the choice of a prior distribution over $\boldsymbol{\rho}$. In literature, there is no standard approach to choose the parameters α_ρ and β_ρ of the inverse gamma distribution other than using the plot of the probability density function [145]. We conduct similar informative covariate identification procedures as in Section 5.3.2, except using different scale and shape parameters of the inverse gamma prior distribution over $\boldsymbol{\rho}$. In Section 5.3.2, α_ρ and β_ρ are set to 4 and 1, respectively. Table 5.9 and Table 5.10 show the covariate identification results when $\alpha_\rho = 2$ and $\beta_\rho = 1$, and when $\alpha_\rho = 4$ and $\beta_\rho = 2$,

Table 5.9 Proportion of simulations that each covariate is identified as informative when $\alpha_\rho = 2$ and $\beta_\rho = 1$ (Truly informative covariates are highlighted in bold)

	Covariate																			
Percentile	**1**	**2**	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
90th	**1**	**1**	0.18	0.14	0.26	0.10	0.08	0.22	0.10	0.08	0.14	0.02	0.04	0.10	0.14	0.08	0.14	0.16	0.06	0.10
95th	**1**	**0.98**	0.14	0.14	0.20	0.08	0.08	0.16	0.08	0.04	0.10	0.02	0.04	0.08	0.10	0.06	0.12	0.08	0.04	0.06

Table 5.10 Proportion of simulations that each covariate is identified as informative when $\alpha_\rho = 4$ and $\beta_\rho = 2$ (Truly informative covariates are highlighted in bold)

	Covariate																			
Percentile	**1**	**2**	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
90th	**1**	**1**	0.12	0.20	0.14	0.14	0.06	0.18	0.10	0.10	0.12	0.14	0.10	0.10	0.18	0.14	0.06	0.22	0.08	0.16
95th	**1**	**1**	0.10	0.10	0.12	0.12	0.03	0.16	0.08	0.08	0.10	0.12	0.00	0.08	0.20	0.08	0.02	0.18	0.04	0.10

respectively. With a different prior distribution over $\boldsymbol{\rho}$, the significance scores of the existing covariates are expected to change. Nevertheless, as the significance score of the inert covariate also changes accordingly, the results repeating 50 simulations show that the proposed method still identifies the important covariates with high probability. Compared to Table 5.1, the results in Table 5.9 and Table 5.10 slightly change, indicating the robustness of the proposed method to the choice of the parameters of the prior distribution over $\boldsymbol{\rho}$.

5.4 Case Study

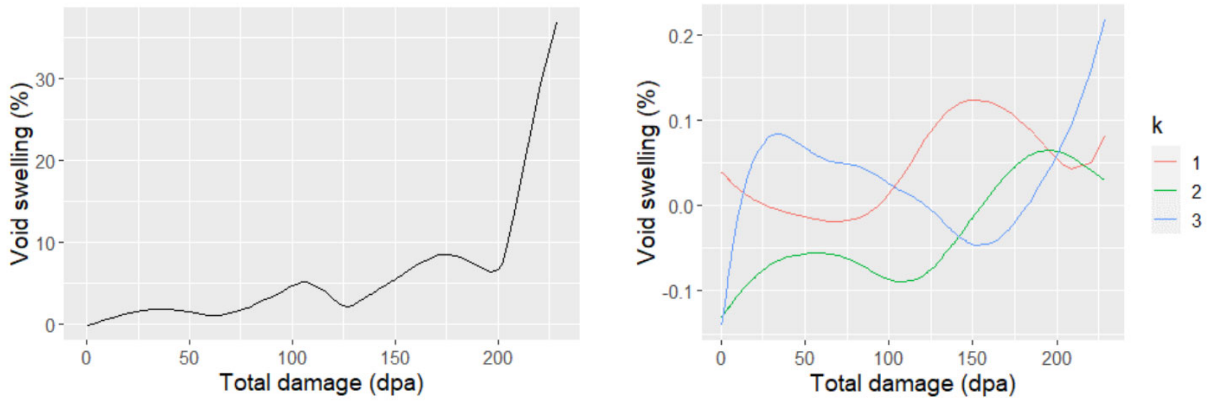
5.4.1 Void Swelling

In this subsection, we apply the proposed method to investigate which covariates affect swelling processes described in Section 5.1.1, and estimate the trajectory of the swelling process given the covariate information.

We collected the data from a large amount of existing literature on void swelling in austenitic stainless steel. The created dataset consists of 238 subjects with a total number of 317 measurements, as illustrated in Figure 5.1 (a). Note that out of a total of 238 subjects, 219 subjects have only one measurement. The full list of covariates is summarized in Table 5.11. As demonstrated in Section 5.2.2, we create four dummy variables applying the one-hot encoding to "Irradiation Type," resulting in $M = 21$. Pooling all subjects together, the mean function $\mu(t)$ and the principal component functions $\phi_1(t), \dots, \phi_K(t)$ are extracted. The value of K is set to 3 which explains 92.16% of total variations. Figure 5.6 illustrates the mean function and the top 3 principal component functions. The 600 iterations of the NUTS are then run to generate posterior

Table 5.11 Summary of all covariates in the void swelling dataset

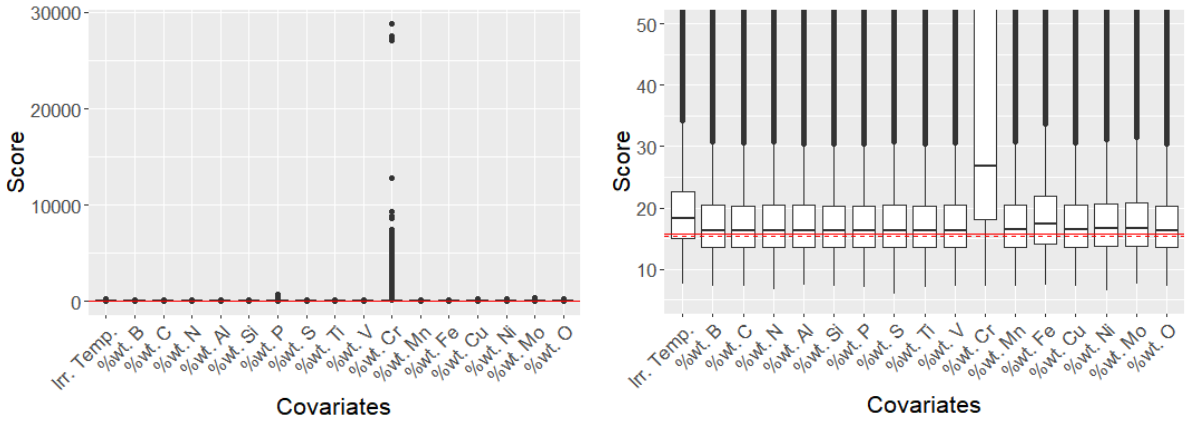
Continuous Covariate		Unit
Irradiation Temperature		°C
Alloy Composition	B C N Al Si P S Ti V Cr Mn Fe Cu Ni Mo O	wt. %
Categorical Covariate		**Level**
Irradiation Type		Ni6+ Fe2+ Neutron Proton Electron



(a) (b)

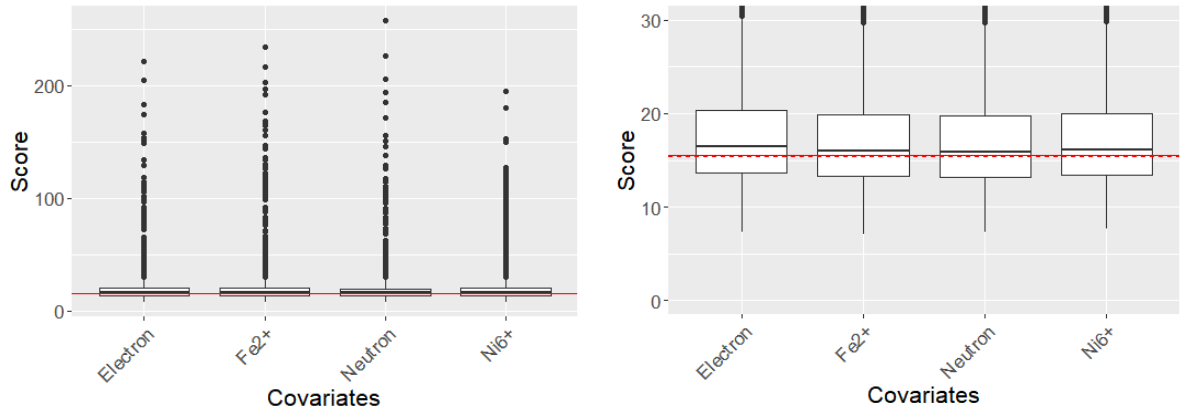
Figure 5.6 Pooled FPCA results of the void swelling: (a) mean function and (b) top 3 principal component functions

realizations of $\boldsymbol{\rho}$, with the first 500 discarded as burn-in. The augmentation of two inert covariates—continuous (generated from $Uniform(0,1)$) and categorical (generated from



(a) (b)

Figure 5.7 Posterior distributions of s_m of the continuous covariates in the void swelling dataset ((a) and (b) have different y-axis scales). Horizontal solid (dashed) line: the 95th (90th) percentile of the reference distribution



(a) (b)

Figure 5.8 Posterior distributions of s_m of the categorical covariates in the void swelling dataset ((a) and (b) have different y-axis scales). Horizontal solid (dashed) line: the 95th (90th) percentile of the reference distribution

$Bernoulli(1/5)$)—and the NUTS are repeated 300 times to construct the reference distribution. Figure 5.7 visualizes the boxplots of the posterior realizations of continuous covariates where the horizontal solid (dashed) line is the 95th (90th) percentile of the reference distribution. Similarly, Figure 5.8 shows the results of categorical covariates. The results show that all covariates are identified as informative using the 95th or 90th percentiles of the reference distribution as a criterion, i.e., all covariates have some effects on the swelling processes. In fact, the alloy

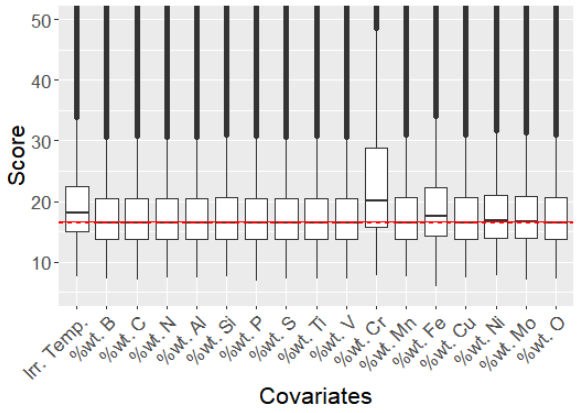


Figure 5.9 Posterior distributions of s_m of the continuous covariates in the void swelling dataset when the inert covariate has a small effect. Horizontal solid (dashed) line: the 95th (90th) percentile of the reference distribution. The y-axis range is set to [5, 50] for better visualization.

Table 5.12 MLE results of length-scale parameters of conventional ARD

Covariate	Value	Covariate	Value
Total Damage	0.094	Irradiation Temperature	0.086
B	$> 10^3$	Mn	$> 10^3$
C	0.562	Fe	$> 10^3$
N	$> 10^3$	Cu	0.120
Al	0.022	Ni	4.067
Si	0.642	Mo	34.10
P	0.152	Ni6	$> 10^3$
S	558.1	Fe2	0.015
Ti	7.155	Neutron	121.3
V	0.042	Proton	0.153
Cr	$> 10^3$	Electron	0.126

composition affects the duration of the transient regime of alloys' swelling processes, and different irradiation types are also known to cause different morphologies of damage cascades and long-term microstructure development [153]. The irradiation temperature is another factor that significantly affects swelling processes, although the exact cause and relations are not fully investigated. Please refer to [124], [153] for more detailed explanations. We further adjust the inert

covariates to have small effects ($\eta = 0.2$) and repeat the informative covariate identification procedure. As a result, three continuous covariates are found to have significant effects on void swelling: Irradiation temperature, %wt. Fe, and %wt. Cr, as visualized in Figure 5.9. This also agrees with the domain knowledge on void swelling that the first-order major element composition (e.g., Fe, Cr, and Ni) has more significant effects than other minor element compositions. The average values of the alloy composition of Fe, Cr, and Ni are 62.72%, 16.11%, and 16.65%, respectively. The informative covariate identification results can be further used to design an experiment to investigate the optimal condition to extend the swelling period to the maximum (e.g., informed alloy design).

In comparison, the point estimation of ρ_{km} using MLE in the conventional GP is also provided in Table 5.12. We can see that it is not straightforward to set an appropriate value of ξ and decide which covariates to screen out. More importantly, the conventional GP fails to identify the important covariates such as %wt. Cr and %wt. Fe, yet concludes other covariates as informative whose effects on void swelling are less clear.

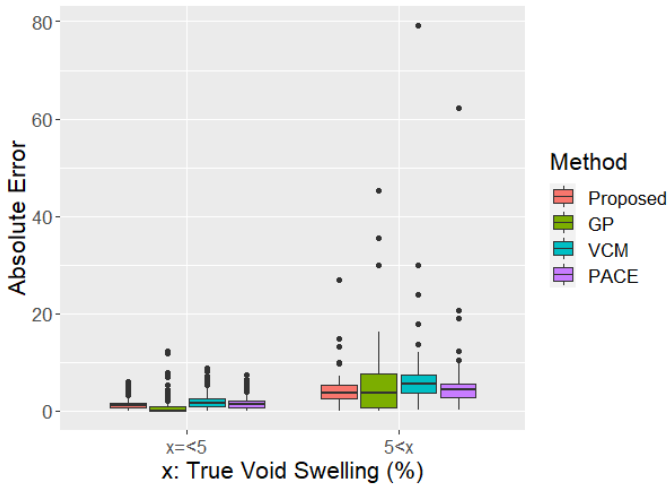


Figure 5.10 Comparison results of the absolute errors by using the benchmark methods and the proposed method according to the true void swelling values

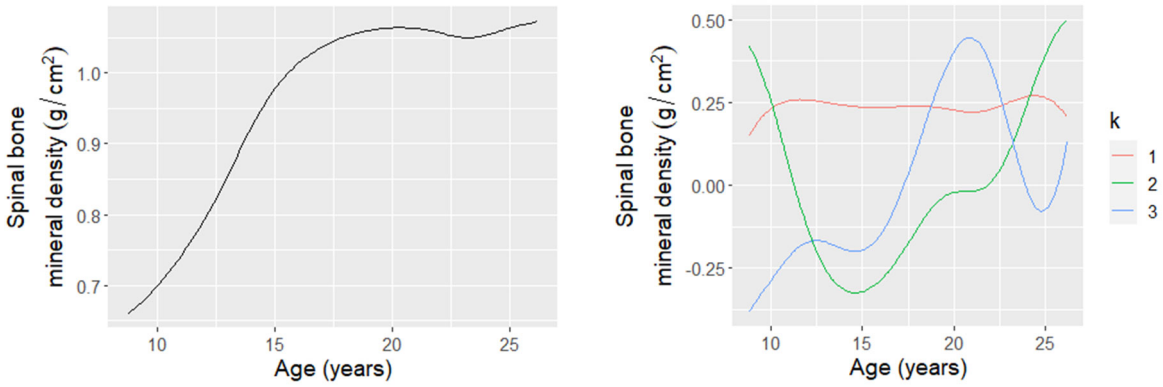
To evaluate the prediction performance of the proposed method, we conduct leave-one-out cross validation (i.e., considering each subject as a testing set and the rest of 237 subjects as a training set). The absolute errors, i.e., $|(True\ Value) - (Estimated\ Value)|$, are calculated and compared with the benchmarks. The results are shown in Figure 5.10, indicating that the proposed method results in fewer outliers and overall outperforms the benchmark methods. Especially, the advantage of the proposed method seems to be more significant when the true values of void swelling are large, which is important for practical applications. Although the GP shows comparable prediction performance with the proposed method, the GP fails to identify informative covariates and thus results in more outliers with larger errors than the proposed method. Note that the CD-FPCA is not included in the results as it cannot handle high-dimensional covariates and even with the pre-selected covariates, it largely overfits the data and results in much higher errors than others.

5.4.2 Spinal Bone Mineral Density

In this subsection, the proposed method is applied to evaluate whether sex or ethnicity is an important factor affecting the spinal BMD. We use the publicly available dataset from the R package loon. As illustrated in Figure 5.1 (b), the dataset includes 423 young (aged 9-26 years) subjects, where each subject collects 1~4 measurements of the spinal BMD [126]. Here, we focus on 280 subjects with more than one measurement. There are two categorical covariates: sex and

Table 5.13 Covariates in Spinal BMD dataset

Covariate	Level	# of Subjects
Sex	Female	153
	Male	127
Ethnicity	Asian	71
	Black	67
	Hispanic	52
	White	90



(a) (b)

Figure 5.12 Pooled FPCA results of the spinal BMD: (a) mean function and (b) top 3 principal component functions

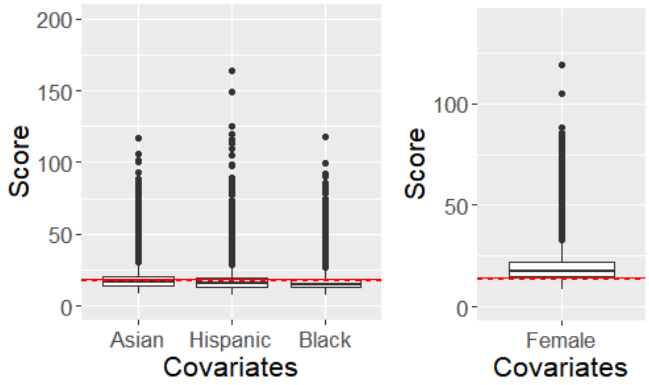


Figure 5.11 Posterior distributions of s_m of the spinal BMD dataset. Horizontal solid (dashed) line: the 95th (90th) percentile of the reference distribution

ethnicity, as summarized in Table 5.13. Using the one-hot encoding, four binary covariates are designed. We follow the same detailed setting described in Section 5.4.1. In the pooled FPCA, the value of K is set to 3 explaining 93.56% of total variations. Figure 5.12 shows the mean function and the top 3 principal component functions.

In terms of the inert covariate, two binary covariates are added: one sampled from $Bernoulli(1/2)$ and another sampled from $Bernoulli(1/4)$. Each corresponds to sex and ethnicity. Figure 5.11 (a) and (b) visualize the boxplots of the posterior realizations of ethnicity and sex, respectively, where the horizontal solid (dashed) line is the 95th (90th) percentile of the

reference distribution. The results show that sex has significant effects on spinal BMD, but not ethnicity. This agrees with the existing analysis that has consistently reported the sex differences in spinal BMD, yet observed discrepant results for the ethnic differences in spinal BMD depending on different sample size or experimental design [126], [154]–[156].

5.5 Conclusion

This work aims at cases where each subject has sparse measurements and records subject-level covariates. This problem can be found in a wide range of applications such as nuclear engineering, manufacturing, and healthcare. To address this, the proposed method models the between-subjects variation coming from covariates and the within-subject variation conditioned on covariates in a systematic manner. The numerical study results demonstrate that the proposed method outperforms existing methods in model fitting, prediction, and informative covariate identification accuracy.

There are several potential topics for future research. First, this paper assumes covariates are static. The extension to the inclusion of time-varying covariates will be of great interest in our future research, e.g., through function-on-function models. Second, the ARD is known to overestimate the relevance of nonlinear variables (covariates) compared to the relevance of linear variables of equal relevance [157]. A systematic approach to address this issue is an area of future research. Third, it is possible that multiple functional data streams are observed from each subject. For instance, bone mineral densities in different locations (spine, hip, and forearm) may be simultaneously measured from each patient. It would be interesting to extend the proposed method to handle such cases, especially through multivariate FPCA [158]. Last but not least, it is worth studying how to extend the proposed method to handle missing covariate values using data imputation.

5.6 Appendix

5.6.1 The proof of eq. (5.7)

In this appendix, we have suppressed the conditioning over $\boldsymbol{\rho}, \boldsymbol{Z}$ and $\boldsymbol{Z}^*$ for brevity. Since $\boldsymbol{Y}|\boldsymbol{A} \sim N(\boldsymbol{\mu} + \boldsymbol{\Phi A}, \sigma_\varepsilon^2 \mathbb{I})$ and $\boldsymbol{A} \sim N\big(\boldsymbol{0}, \boldsymbol{K}(\boldsymbol{Z}, \boldsymbol{Z})\big)$, we can show that the posterior distribution $\boldsymbol{A}|\boldsymbol{Y}$ also follows the multivariate Gaussian distribution:

$$\boldsymbol{A}|\boldsymbol{Y} \sim N\big(\boldsymbol{\mu}^{\star(2)}, \boldsymbol{\Sigma}^{\star(2)}\big), \tag{\&1}$$

where $\boldsymbol{\mu}^{\star(2)} = \boldsymbol{\Sigma}^{\star(2)} \left(\frac{1}{\sigma_\varepsilon^2} \boldsymbol{\Phi}^T (\boldsymbol{Y} - \boldsymbol{\mu}) \right)$ and $\boldsymbol{\Sigma}^{\star(2)} = \left(\frac{1}{\sigma_\varepsilon^2} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \boldsymbol{K}^{-1} \right)^{-1}$. Following the definition of a GP, FPC scores of any finite number of subjects have a joint Gaussian distribution. Specifically, the FPC scores of the historical subjects $\boldsymbol{A}$ and those of the new subject $\boldsymbol{A}^\star$ have a joint Gaussian distribution:

$$\begin{pmatrix} \boldsymbol{A}^\star \\ \boldsymbol{A} \end{pmatrix} \sim N\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}^{\star\star} & \boldsymbol{K}^\star \\ \boldsymbol{K}^{\star T} & \boldsymbol{K} \end{bmatrix} \right), \tag{\&2}$$

Since $p(\boldsymbol{A}^\star|\boldsymbol{A}) = p(\boldsymbol{A}^\star, \boldsymbol{A})/p(\boldsymbol{A})$, we can show that $\boldsymbol{A}^\star|\boldsymbol{A} \sim N(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$, where $\boldsymbol{\mu}' = \boldsymbol{K}^\star \boldsymbol{K}^{-1} \boldsymbol{A}$ and $\boldsymbol{\Sigma}' = \boldsymbol{K}^{\star\star} - \boldsymbol{K}^\star \boldsymbol{K}^{-1} \boldsymbol{K}^{\star T}$. Marginalizing as $p(\boldsymbol{A}^\star|\boldsymbol{Y}) = \int p(\boldsymbol{A}^\star|\boldsymbol{A}) p(\boldsymbol{A}|\boldsymbol{Y})\, d\boldsymbol{A}$, we can further prove that

$$\boldsymbol{A}^\star|\boldsymbol{Y} \sim N(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}^\star), \tag{9}$$

with parameters, $\boldsymbol{\mu}^\star = \boldsymbol{K}^\star \boldsymbol{K}^{-1} \boldsymbol{\mu}^{\star(2)}$ and $\boldsymbol{\Sigma}^\star = \boldsymbol{K}^\star \boldsymbol{K}^{-1} \boldsymbol{\Sigma}^{\star(2)} \boldsymbol{K}^{-1} \boldsymbol{K}^{\star T} + \boldsymbol{\Sigma}'$.

5.6.2 Sensitivity to Mis-specification of Inert Covariate

In this subsection, we consider the cases, where the distribution of the inert covariate is mis-specified due to the lack of prior knowledge or limited amount of historical dataset. In such scenarios, the proposed method may provide additional errors in terms of informative covariate

identification. In this simulation, we follow detailed settings described in Section 5.3.2, except that the inert covariate is sampled from the symmetric beta distribution that has been widely used to model the behavior of random variables limited to a finite interval. Note that $Beta(1,1)$ is equivalent to $Uniform(0,1)$. Three distributions are used to sample inert covariates: $Beta(0.8, 0.8)$, $Beta(1,1)$ and $Beta(1.2, 1.2)$. Figure 5.13 shows the probability density functions of the corresponding beta distributions.

The simulation is repeated 50 times for each of the Beta distributions. The covariate selection results are in Table 5.14.

The results in Table 5.14 show that the proposed method is quite robust with respect to the mis-specification of the inert covariate. In particular, the proposed method was able to correctly identify

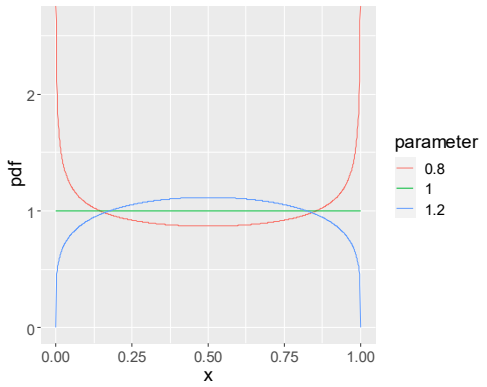


Figure 5.13 Probability density functions of the beta distributions of mis-specified inert covariates

Table 5.14 Average proportion of simulations that a covariate is identified as informative/irrelevant when the inert covariate is mis-specified (correct distribution of the inert covariate is $Beta(1.0, 1.0)$)

Percentile	Inert covariate	Covariate	
		Informative	Irrelevant
90th	*Beta(0.8, 0.8)*	1	0.132
	Beta(1.0, 1.0)	1	0.123
	Beta(1.2, 1.2)	1	0.128
95th	*Beta(0.8, 0.8)*	1	0.083
	Beta(1.0, 1.0)	1	0.068
	Beta(1.2, 1.2)	1	0.075

informative covariates despite of mis-specified inert covariate. Yet, when the distribution of the inert covariate is mis-specified (i.e., $Beta(0.8, 0.8)$ or $Beta(1.2, 1.2)$), there is slightly higher chance of falsely identifying non-informative covariates as informative. Recall that the proposed method identifies informative covariates by measuring how rapidly FPC score changes according to the covariate change. Roughly speaking, when the distribution of the inert distribution is mis-specified, the differences of the inert covariates between different subjects will be distorted, while the corresponding differences of the FPC score estimations between subjects remain the same. This may lead to "less" rapid changes of FPC scores, i.e., the significance of the inert covariate is underestimated, and increases the false positive rate (decreases the false negative rate).

Chapter 6 Summary

Degradation modeling and prognostics are crucial to avoid unanticipated system failures, associated economic losses, and even safety issues. This thesis aims to address major challenges in exploring novel applications of degradation modeling and prognostics. The contributions of this thesis can be summarized as follows: (1) established a novel health index-based method which selects informative sensors and combines multiple sensor signals to achieve more accurate degradation modeling and prognostics and better interpretability; (2) developed a novel Bayesian deep learning framework that performs degradation modeling and prognostics in complex systems involving multiple sensors, multiple failure modes and multiple operational conditions from a probabilistic point of view; (3) established an individualized degradation modeling and prognostics of a heterogeneous group which encodes the available information about intrinsic covariates into the random-effect coefficients and quantifies the similarities between different units; (4) proposed a covariate-dependent functional data analysis method to identify informative covariates when each system has sparse and irregular longitudinal measurements and there is no physical knowledge available for a functional form of the degradation process.

Chapter 7 References

[1] C. J. Lu and W. Q. Meeker, "Using Degradation Measures to Estimate a Time-to-Failure Distribution," *Technometrics*, vol. 35, no. 2, pp. 161–174, 1993, doi: 10.1080/00401706.1993.10485038.

[2] N. Gebraeel, "Sensory-Updated Residual Life Distributions for Components With Exponential Degradation Patterns," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 382–393, 2006, doi: 10.1109/TASE.2006.876609.

[3] X. S. Si, W. Wang, C. H. Hu, D. H. Zhou, and M. G. Pecht, "Remaining Useful Life Estimation Based on a Nonlinear Diffusion Degradation Process," *IEEE Trans. Reliab.*, vol. 61, no. 1, pp. 50–67, Mar. 2012, doi: 10.1109/TR.2011.2182221.

[4] J. Guo, Z. Li, and M. Li, "A Review on Prognostics Methods for Engineering Systems," *IEEE Trans. Reliab.*, vol. 69, no. 3, pp. 1110–1129, 2019.

[5] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proc. IEEE*, vol. 85, no. 1, pp. 6–23, 1997, doi: 10.1109/5.554205.

[6] W. Wang and A. H. Christer, "Towards a general condition based maintenance model for a stochastic dynamic system," *J. Oper. Res. Soc.*, vol. 51, no. 2, pp. 145–155, Feb. 2000, doi: 10.1057/palgrave.jors.2600863.

[7] M. Rausand and A. Høyland, *System reliability theory : models, statistical methods, and applications*. Wiley-Interscience, 2004.

[8] Y. Lei, N. Li, S. Gontarz, J. Lin, S. Radkowski, and J. Dybala, "A Model-Based Method for Remaining Useful Life Prediction of Machinery," *IEEE Trans. Reliab.*, vol. 65, no. 3, 2016, doi: 10.1109/TR.2016.2570568.

[9] D. An, N. H. Kim, and J. H. Choi, "Practical options for selecting data-driven or physics-based prognostics algorithms with reviews," *Reliab. Eng. Syst. Saf.*, vol. 133, pp. 223–236, 2015, doi: 10.1016/j.ress.2014.09.014.

[10] Z. S. Ye, N. Chen, and Y. Shen, "A new class of Wiener process models for degradation analysis," *Reliab. Eng. Syst. Saf.*, vol. 139, pp. 58–67, 2015, doi: 10.1016/j.ress.2015.02.005.

[11] X. Wang, N. Balakrishnan, and B. Guo, "Residual life estimation based on a generalized Wiener degradation process," *Reliab. Eng. Syst. Saf.*, vol. 124, pp. 13–23, 2014, doi: 10.1016/j.ress.2013.11.011.

[12] Z. S. Ye and N. Chen, "The Inverse Gaussian Process as a Degradation Model," *Technometrics*, vol. 56, no. 3, pp. 302–311, Jul. 2014, doi: 10.1080/00401706.2013.830074.

[13] J. Lawless and M. Crowder, "Covariates and random effects in a gamma process model with application to degradation and failure," *Lifetime Data Anal.*, vol. 10, no. 3, pp. 213–227, 2004, doi: 10.1023/B:LIDA.0000036389.14073.dd.

[14] V. Bagdonavičius and M. S. Nikulin, "Estimation in Degradation Models with Explanatory Variables," *Lifetime Data Anal.*, vol. 7, no. 1, pp. 85–103, 2001, doi: 10.1023/A:1009629311100.

[15] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mech. Syst. Signal Process.*, vol. 21, no. 6, pp. 2560–2574, Aug. 2007, doi: 10.1016/J.YMSSP.2006.12.007.

[16] Y. Lei and M. J. Zuo, "Gear crack level identification based on weighted K nearest neighbor classification algorithm," *Mech. Syst. Signal Process.*, vol. 23, no. 5, pp. 1535–1547, Jul. 2009, doi: 10.1016/J.YMSSP.2009.01.009.

[17] V. T. Tran, B.-S. Yang, M.-S. Oh, and A. C. C. Tan, "Fault diagnosis of induction motor based on decision trees and adaptive neuro-fuzzy inference," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 1840–1849, Mar. 2009, doi: 10.1016/J.ESWA.2007.12.010.

[18] C. G. Huang, H. Z. Huang, and Y. F. Li, "A Bi-Directional LSTM prognostics method under multiple operational conditions," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8792–8802, 2019,

doi: 10.1109/TIE.2019.2891463.

[19] M. Kim and K. Liu, "A Bayesian Deep Learning Framework for Interval Estimation of Remaining Useful Life in Complex Systems by Incorporating General Degradation Characteristics," *IISE Trans.*, vol. 53, no. 3, pp. 326–340, 2020, doi: 10.1080/24725854.2020.1766729.

[20] K. Liu, N. Z. Gebraeel, and J. Shi, "A Data-Level Fusion Model for Developing Composite Health Indices for Degradation Modeling and Prognostic Analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 652–664, Jul. 2013, doi: 10.1109/TASE.2013.2250282.

[21] K. Liu and S. Huang, "Integration of Data Fusion Methodology and Degradation Modeling Process to Improve Prognostics," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 344–354, Jan. 2016, doi: 10.1109/TASE.2014.2349733.

[22] K. Liu, A. Chehade, and C. Song, "Optimize the Signal Quality of the Composite Health Index via Data Fusion for Degradation Modeling and Prognostic Analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1504–1514, Jul. 2017, doi: 10.1109/TASE.2015.2446752.

[23] L. Bian and N. Gebraeel, "Stochastic modeling and real-time prognostics for multi-component systems with degradation rate interactions," *IIE Trans.*, vol. 46, no. 5, pp. 470–482, May 2014, doi: 10.1080/0740817X.2013.812269.

[24] A. Chehade, C. Song, K. Liu, A. Saxena, and X. Zhang, "A data-level fusion approach for degradation modeling and prognostic analysis under multiple failure modes," *J. Qual. Technol.*, vol. 50, no. 2, pp. 150–165, Apr. 2018, doi: 10.1080/00224065.2018.1436829.

[25] H. Yan, K. Liu, X. Zhang, and J. Shi, "Multiple Sensor Data Fusion for Degradation Modeling and Prognostics under Multiple Operational Conditions," *IEEE Trans. Reliab.*, vol. 65, no. 3, pp. 1416–1426, 2016, doi: 10.1109/TR.2016.2575449.

[26] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, 2017, doi: 10.1016/j.neucom.2017.02.045.

[27] Y. Hong, Y. Duan, W. Q. Meeker, D. L. Stanley, and X. Gu, "Statistical methods for degradation data with dynamic covariates information and an application to outdoor weathering data," *Technometrics*, vol. 57, no. 2, pp. 180–193, 2015, doi: 10.1080/00401706.2014.915891.

[28] D. R. Cox, "Regression Models and Life-Tables," *J. R. Stat. Soc.*, vol. 34, no. 2, pp. 527–541, 1972, doi: 10.1007/978-1-4612-4380-9_37.

[29] V. T. Tran, H. T. Phama, B.-S. Yang, and T. T. Nguyen, "Machine performance degradation assessment and remaining useful life prediction using proportional hazard model and support vector machine," *Mech. Syst. Signal Process.*, vol. 32, pp. 320–330, Oct. 2012, doi: 10.1016/J.YMSSP.2012.02.015.

[30] Haitao Liao, Wenbiao Zhao, and Huairui Guo, "Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model," in *RAMS '06. Annual Reliability and Maintainability Symposium, 2006.*, 2006, pp. 127–132, doi: 10.1109/RAMS.2006.1677362.

[31] Y. Lin, K. Liu, E. Byon, X. Qian, S. Liu, and S. Huang, "A Collaborative Learning Framework for Estimating Many Individualized Regression Models in a Heterogeneous Population," *IEEE Trans. Reliab.*, vol. 67, no. 1, pp. 328–341, 2018, doi: 10.1109/TR.2017.2767941.

[32] F. Garner, "Irradiation Performance of Cladding and Structural Steels in Liquid Metal Reactors," in *Materials Science and Technology*, Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA, 2006.

[33] X. S. Si, W. Wang, C. Hu, and D. Zhou, "Remaining useful life estimation – A review on the statistical data driven approaches," *Eur. J. Oper. Res.*, vol. 213, no. 1, pp. 1–14, 2011, doi: 10.1016/j.ejor.2010.11.018.

[34] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mech. Syst. Signal Process.*, vol. 20, no. 7, pp. 1483–1510, 2006, doi: 10.1016/j.ymssp.2005.09.012.

[35] C. Hu, B. D. Youn, P. Wang, and J. Taek Yoon, "Ensemble of data-driven prognostic algorithms

for robust prediction of remaining useful life," *Reliab. Eng. Syst. Saf.*, vol. 103, pp. 120–135, 2012, doi: 10.1016/j.ress.2012.03.008.

[36] Z. Tian, "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring," *J. Intell. Manuf.*, vol. 23, no. 2, pp. 227–237, Apr. 2012, doi: 10.1007/s10845-009-0356-9.

[37] T. H. Loutas, D. Roulias, and G. Georgoulas, "Remaining Useful Life Estimation in Rolling Bearings Utilizing Data-Driven Probabilistic E-Support Vectors Regression," *IEEE Trans. Reliab.*, vol. 62, no. 4, pp. 821–832, Dec. 2013, doi: 10.1109/TR.2013.2285318.

[38] Zhengguo Xu, Yindong Ji, and Donghua Zhou, "Real-time Reliability Prediction for a Dynamic System Based on the Hidden Degradation Process Identification," *IEEE Trans. Reliab.*, vol. 57, no. 2, pp. 230–242, Jun. 2008, doi: 10.1109/TR.2008.916882.

[39] Jianbo Yu, "State-of-Health Monitoring and Prediction of Lithium-Ion Battery Using Probabilistic Indication and State-Space Model," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 11, pp. 2937–2949, Nov. 2015, doi: 10.1109/TIM.2015.2444237.

[40] X. Fang, K. Paynabar, and N. Gebraeel, "Multistream sensor fusion-based prognostics model for systems with single failure modes," *Reliab. Eng. Syst. Saf.*, vol. 159, pp. 322–331, Mar. 2017, doi: 10.1016/j.ress.2016.11.008.

[41] C. Song, K. Liu, and X. Zhang, "Integration of Data-Level Fusion Model and Kernel Methods for Degradation Modeling and Prognostic Analysis," *IEEE Trans. Reliab.*, vol. 67, no. 2, pp. 640–650, 2017.

[42] C. Song and K. Liu, "Statistical degradation modeling and prognostics of multiple sensor signals via data fusion: A composite health index approach," *IISE Trans.*, vol. 50, no. 10, pp. 853–867, Oct. 2018, doi: 10.1080/24725854.2018.1440673.

[43] F. Yang, M. S. Habibullah, T. Zhang, Z. Xu, P. Lim, and S. Nadarajan, "Health index-based prognostics for remaining useful life predictions in electrical machines," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2633–2644, Apr. 2016, doi: 10.1109/TIE.2016.2515054.

[44] N. Chen and K. L. Tsui, "Condition monitoring and remaining useful life prediction using degradation signals: revisited," *IIE Trans.*, vol. 45, no. 9, pp. 939–952, Sep. 2013, doi: 10.1080/0740817X.2012.706376.

[45] Z. S. Ye, Y. Wang, K. L. Tsui, and M. Pecht, "Degradation data analysis using wiener processes with measurement errors," *IEEE Trans. Reliab.*, 2013, doi: 10.1109/TR.2013.2284733.

[46] Z.-S. Ye and M. Xie, "Stochastic modelling and analysis of degradation for highly reliable products," *Appl. Stoch. Model. Bus. Ind.*, vol. 31, no. 1, pp. 16–32, Jan. 2015, doi: 10.1002/asmb.2063.

[47] G. Wang, J. Jiao, and S. Yin, "A Kernel Direct Decomposition-Based Monitoring Approach for Nonlinear Quality-Related Fault Detection," *IEEE Trans. Ind. Informatics*, vol. 13, no. 4, pp. 1565–1574, Aug. 2017, doi: 10.1109/TII.2016.2633989.

[48] G. Wang and J. Jiao, "A Kernel Least Squares Based Approach for Nonlinear Quality-Related Fault Detection," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3195–3204, Apr. 2017, doi: 10.1109/TIE.2016.2637886.

[49] R. C. Hill, W. E. Griffiths, and G. C. Lim, *Principles of Econometrics*, 4th ed. Wiley, 2011.

[50] H. Zou, "The Adaptive Lasso and Its Oracle Properties," *J. Am. Stat. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006, doi: 10.1198/016214506000000735.

[51] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58. WileyRoyal Statistical Society, pp. 267–288, 1996, doi: 10.2307/2346178.

[52] Y. Liu, D. K. Frederick, J. A. DeCastro, J. S. Litt, and W. W. Chan, *User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)*, 2nd ed. National Aeronautics and Space Administration, Glenn Research Center, 2012.

[53] W. Peng, L. Hong, and Z. Ye, "Degradation-Based Reliability Modeling of Complex Systems in Dynamic Environments," in *Statistical Modeling for Degradation Data*, 2017, pp. 81–103.

[54] S. Khan and T. Yairi, “A review on the application of deep learning in system health management,” *Mech. Syst. Signal Process.*, vol. 107, pp. 241–265, Jul. 2018, doi: 10.1016/J.YMSSP.2017.11.024.

[55] Z. Tian, L. Wong, and N. Safaei, “A neural network approach for remaining useful life prediction utilizing both failure and suspension histories,” *Mech. Syst. Signal Process.*, vol. 24, no. 5, pp. 1542–1555, Jul. 2010, doi: 10.1016/J.YMSSP.2009.11.005.

[56] X. Li, Q. Ding, and J. Sun, “Remaining useful life estimation in prognostics using deep convolution neural networks,” *Reliab. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018, doi: 10.1016/J.RESS.2017.11.021.

[57] Y. LeCun *et al.*, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.

[58] J. Zhu, N. Chen, and W. Peng, “Estimation of Bearing Remaining Useful Life based on Multiscale Convolutional Neural Network,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3208–3216, 2018, doi: 10.1109/TIE.2018.2844856.

[59] A. Graves, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012.

[60] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, “Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, 2017, doi: 10.1109/TNNLS.2016.2582798.

[61] L. Liao, W. Jin, and R. Pavel, “Enhanced Restricted Boltzmann Machine With Prognosability Regularization for Prognostics and Health Assessment,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7076–7083, Nov. 2016, doi: 10.1109/TIE.2016.2586442.

[62] N. Z. Gebraeel and M. A. Lawley, “A Neural Network Degradation Model for Computing and Updating Residual Life Distributions,” *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 154–163, Jan. 2008, doi: 10.1109/TASE.2007.910302.

[63] C. G. Huang, X. Yin, H. Z. Huang, and Y. F. Li, “An Enhanced Deep Learning-Based Fusion Prognostic Method for RUL Prediction,” *IEEE Trans. Reliab.*, p. in-press, 2019, doi: 10.1109/TR.2019.2948705.

[64] A. K. Garga *et al.*, “Hybrid reasoning for prognostic learning in CBM systems,” *IEEE Aerosp. Conf. Proc.*, vol. 6, pp. 2957–2969, 2001, doi: 10.1109/AERO.2001.931316.

[65] W. Peng, Z. S. Ye, and N. Chen, “Bayesian Deep-Learning-Based Health Prognostics Toward Prognostics Uncertainty,” *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2283–2293, 2020, doi: 10.1109/TIE.2019.2907440.

[66] D. J. C. MacKay, “Bayesian methods for adaptive models,” *Ph.D. thesis, Calif. Inst. Technol.*, 1992, doi: Bayesian methods for adaptive models,.

[67] A. Graves, “Practical Variational Inference for Neural Networks,” in *Advances in Neural Information Processing Systems 24*, 2011, pp. 2348–2356, Accessed: Mar. 25, 2019. [Online]. Available: http://papers.nips.cc/paper/4329-practical-variational-inference-for-neural-networks.pdf.

[68] 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 - Volume 48*, 2016, pp. 1050–1059, doi: 10.1109/TKDE.2015.2507132.

[69] D. Soudry, I. Hubara, and R. Meir, “Expectation Backpropagation: Parameter-Free Training of Multilayer Neural Networks with Continuous or Discrete Weights,” in *Advances in Neural Information Processing Systems 27*, 2014, pp. 963–971.

[70] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, “Bayesian Optimization with Robust Bayesian Neural Networks,” in *Advances in Neural Information Processing Systems 29*, 2016, pp. 4134–4142.

[71] I. Osband, C. Blundell, A. Pritzel, B. Van Roy, and B. Van Roy, “Deep Exploration via Bootstrapped DQN,” in *Advances in Neural Information Processing Systems 29*, 2016, pp. 4026–4034.

[72] Y. Gal, “Uncertainty in Deep Learning,” *PhD Thesis, Univ. Cambridge*, 2016, doi: 10.1371/journal.pcbi.1005062.

[73] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving

neural networks by preventing co-adaptation of feature detectors," *arXiv Prepr. arXiv1207.0580*, Jul. 2012, Accessed: Apr. 08, 2019. [Online]. Available: http://arxiv.org/abs/1207.0580.

[74] A. Chehade, S. Bonk, and K. Liu, "Sensory-Based Failure Threshold Estimation for Remaining Useful Life Prediction," *IEEE Trans. Reliab.*, vol. 66, no. 3, pp. 939–949, Sep. 2017, doi: 10.1109/TR.2017.2695119.

[75] N. Gebraeel and J. Pan, "Prognostic degradation models for computing and updating residual life distributions in a time-varying environment," *IEEE Trans. Reliab.*, vol. 57, no. 4, pp. 539–550, 2008, doi: 10.1109/TR.2008.928245.

[76] C. Song, K. Liu, and X. Zhang, "A Generic Framework for Multisensor Degradation Modeling based on Supervised Classification and Failure Surface," *IISE Trans.*, vol. 51, no. 11, pp. 1288–1302, Jan. 2019, doi: 10.1080/24725854.2018.1555384.

[77] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.

[78] S. Hochreiter and J. Schmidhuber, "LONG SHORT-TERM MEMORY," *Neural Comput.*, vol. 9, no. 8, pp. 1–32, 1997, doi: 10.1162/neco.1997.9.8.1735.

[79] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.

[80] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data Augmentation for Time Series Classification using Convolutional Neural Networks," Sep. 2016, [Online]. Available: https://halshs.archives-ouvertes.fr/halshs-01357973.

[81] A. Saxena and K. Goebel, "Turbofan Engine Degradation Simulation Data Set. NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA," 2008.

[82] T. Tieleman and G. Hinton., "RmsProp: Divide the gradient by a running average of its recent magnitude," 2012.

[83] G. Sateesh Babu, P. Zhao, and X.-L. Li, "Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life," in *International Conference on Database Systems for Advanced Applications*, 2016, pp. 214–228, Accessed: Oct. 16, 2018. [Online]. Available: http://www.i2r.a-star.edu.sg.

[84] M. Kim, C. Song, and K. Liu, "A Generic Health Index Approach for Multisensor Degradation Modeling and Sensor Selection," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1426–1437, 2019, doi: 10.1109/TASE.2018.2890608.

[85] B. Bole, C. S. Kulkarni, and M. Daigle, "Adaptation of an electrochemistry-based Li-ion battery model to account for deterioration observed under randomized use," 2014.

[86] A. Chehade and K. Liu, "Structural Degradation Modeling Framework for Sparse Datasets with an application on Alzheimer ' s Disease," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 192–205, 2019, doi: 10.1109/TASE.2018.2829770.

[87] L. Hao, K. Liu, N. Gebraeel, and J. Shi, "Controlling the Residual Life Distribution of Parallel Unit Systems Through Workload Adjustment," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1042–1052, Apr. 2017, doi: 10.1109/TASE.2015.2481703.

[88] M. Short *et al.*, "Examination of issues involved when using ion irradiation to simulate void swelling and microstructural stability of ferritic-martensitic alloys in spallation environments," 2014, [Online]. Available: https://indico.psi.ch/event/3052/images/639-O9-1_M._Short__IWSMT-12_09-1.pdf.

[89] S. C. Chow and J. Shao, "Estimating Drug Shelf-Life with Random Batches," *Biometrics*, vol. 47, no. 3, p. 1071, Sep. 1991, doi: 10.2307/2532659.

[90] W. Peng, Y.-F. Li, J. Mi, L. Yu, and H.-Z. Huang, "Reliability of complex systems under dynamic conditions: A Bayesian multivariate degradation perspective," *Reliab. Eng. Syst. Saf.*, vol. 153, pp. 75–87, Sep. 2016, doi: 10.1016/J.RESS.2016.04.005.

[91] Z. Xu, Y. Hong, and R. Jin, "Nonlinear general path models for degradation data with dynamic covariates," *Appl. Stoch. Model. Bus. Ind.*, vol. 32, no. 2, pp. 153–167, 2016, doi: 10.1002/asmb.2129.

[92] R. Thapa, H. E. Burkhart, J. Li, and Y. Hong, "Modeling Clustered Survival Times of Loblolly Pine with Time-dependent Covariates and Shared Frailties," *J. Agric. Biol. Environ. Stat.*, vol. 21, pp. 92–110, 2016, doi: 10.1007/s13253-015-0217-2.

[93] Y. Hong and W. Q. Meeker, "Field-Failure Predictions Based on Failure-Time Data With Dynamic Covariate Information," *Technometrics*, vol. 55, no. 2, pp. 135–149, May 2013, doi: 10.1080/00401706.2013.765324.

[94] D. Lugtigheid, X. Jiang, and A. K. S. Jardine, "A finite horizon model for repairable systems with repair restrictions," *J. Oper. Res. Soc.*, vol. 59, no. 10, pp. 1321–1331, Oct. 2008, doi: 10.1057/palgrave.jors.2602471.

[95] X. Zhao, M. Fouladirad, C. Bérenguer, and L. Bordes, "Condition-based inspection/replacement policies for non-monotone deteriorating systems with environmental covariates," *Reliab. Eng. Syst. Saf.*, vol. 95, no. 8, pp. 921–934, 2010, doi: 10.1016/j.ress.2010.04.005.

[96] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, 2016, doi: 10.1186/s40537-016-0043-6.

[97] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.

[98] K. B. Goode, J. Moore, and B. J. Roylance, "Plant machinery working life prediction method utilizing reliability and condition-monitoring data," *Proc. Inst. Mech. Eng. Part E J. Process Mech. Eng.*, vol. 214, no. 2, pp. 109–122, May 2000, doi: 10.1243/0954408001530146.

[99] Y. Li, L. Wang, J. Wang, J. Ye, and C. K. Reddy, "Transfer learning for survival analysis via efficient L2,1-Norm regularized cox regression," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 231–240, doi: 10.1109/ICDM.2016.129.

[100] N. Zou, Y. Zhu, J. Zhu, M. Baydogan, W. Wang, and J. Li, "A Transfer Learning Approach for Predictive Modeling of Degenerate Biological Systems," *Technometrics*, vol. 57, no. 3, pp. 362–373, 2015, doi: 10.1080/00401706.2015.1044117.

[101] R. Kontar, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, "Nonparametric Modeling and Prognosis of Condition Monitoring Signals Using Multivariate Gaussian Convolution Processes," *Technometrics*, vol. 60, no. 4, pp. 484–496, 2018, doi: 10.1080/00401706.2017.1383310.

[102] J. Q. Shi, B. Wang, and D. M. Titterington, "Gaussian Process Functional Regression Modeling for Batch Data," *Biometrics*, vol. 63, pp. 714–723, 2007, doi: 10.1111/j.1541-0420.2007.00758.x.

[103] J. C. Biesanz, N. Deeb-Sossa, A. A. Papadakis, K. A. Bollen, and P. J. Curran, "The Role of Coding Time in Estimating and Interpreting Growth Curve Models.," *Psychol. Methods*, vol. 9, no. 1, pp. 30–52, Mar. 2004, doi: 10.1037/1082-989X.9.1.30.

[104] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*, 2nd ed. Springer-Verlag New York, 2005.

[105] J. Guo and Z. Li, "Prognostics of Lithium ion battery using functional principal component analysis," *2017 IEEE Int. Conf. Progn. Heal. Manag. ICPHM 2017*, pp. 14–17, 2017, doi: 10.1109/ICPHM.2017.7998299.

[106] Y. Cheng, C. Lu, T. Li, and L. Tao, "Residual lifetime prediction for lithium-ion battery based on functional principal component analysis and Bayesian approach," *Energy*, vol. 90, pp. 1983–1993, 2015, doi: 10.1016/j.energy.2015.07.022.

[107] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[108] S. Conti, A. O'Hagan, and A. O 'hagan, "Bayesian emulation of complex multi-output and dynamic computer models," *J. Stat. Plan. Inference*, vol. 140, pp. 640–651, 2010, doi: 10.1016/j.jspi.2009.08.006.

[109] T. E. Fricker, J. E. Oakley, and N. M. Urban, "Multivariate Gaussian Process Emulators With Nonseparable Covariance Structures," *Technometrics*, vol. 55, no. 1, pp. 47–56, Feb. 2013, doi: 10.1080/00401706.2012.715835.

[110] P. Boyle, P. Boyle, and M. Frean, “Dependent Gaussian processes,” *Adv. Neural Inf. Process. Syst.*, vol. 17, pp. 217–224, 2005, Accessed: Sep. 25, 2018. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.70.2912.

[111] M. Alvarez and N. D. Lawrence, “Sparse Convolved Gaussian Processes for Multi-output Regression,” *Adv. Neural Inf. Process. Syst.*, vol. 21, pp. 57–64, 2009, Accessed: Sep. 25, 2018. [Online]. Available: https://papers.nips.cc/paper/3553-sparse-convolved-gaussian-processes-for-multi-output-regression.

[112] P. Z. G. Qian, H. Wu, and C. F. J. Wu, “Gaussian Process Models for Computer Experiments With Qualitative and Quantitative Factors,” *Technometrics*, vol. 50, no. 3, pp. 383–396, Aug. 2008, doi: 10.1198/004017008000000262.

[113] J. Fan and W. Zhang, “Statistical methods with varying coefficient models,” *Stat. Interface*, vol. 1, no. 1, pp. 179–195, 2008, doi: 10.1016/j.bbi.2008.05.010.

[114] Q. Li and J. S. Racine, “Smooth Varying-Coefficient Nonparametric Models for Qualitative and Quantitative Data,” *Econom. Theory*, vol. 26, pp. 1607–1637, 2010.

[115] D. Nguyen-Tuong, M. Seeger, and J. Peters, “Model Learning with Local Gaussian Process Regression,” *Adv. Robot.*, vol. 23, no. 15, pp. 2015–2034, Jan. 2009, doi: 10.1163/016918609X12529286896877.

[116] S. G. Mueller *et al.*, “The Alzheimer’s Disease Neuroimaging Initiative,” *Neuroimaging Clin. N. Am.*, vol. 15, no. 4, pp. 869–877, Nov. 2005, doi: 10.1016/j.nic.2005.09.008.

[117] J. Zhou, J. Liu, V. A. Narayan, and J. Ye, “Modeling Disease Progression via Fused Sparse Group Lasso,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’12*, 2012, pp. 1095–1103.

[118] M. S. Mendiondo, J. Wesson Ashford, R. J. Kryscio, and F. A. Schmitt, “Modelling Mini Mental State Examination changes in Alzheimer’s disease,” *Stat. Med.*, vol. 19, no. 11–12, pp. 1607–1616, 2000.

[119] M. F. Folstein, S. E. Folstein, and P. R. McHugh, “‘Mini-mental state’. A practical method for grading the cognitive state of patients for the clinician,” *J. Psychiatr. Res.*, vol. 12, no. 3, pp. 189–198, 1975, doi: 10.1016/0022-3956(75)90026-6.

[120] M. J. Sliwinski, S. M. Hofer, C. Hall, H. Buschke, and R. B. Lipton, “Modeling Memory Decline in Older Adults: The Importance of Preclinical Dementia, Attrition, and Chronological Age.,” *Psychol. Aging*, vol. 18, no. 4, pp. 658–671, 2003, doi: 10.1037/0882-7974.18.4.658.

[121] K. Ito *et al.*, “Disease progression model for cognitive deterioration from Alzheimer’s Disease Neuroimaging Initiative database,” *Alzheimer’s Dement.*, vol. 7, no. 2, pp. 151–160, Mar. 2011, doi: 10.1016/J.JALZ.2010.03.018.

[122] C. R. Jack *et al.*, “The Alzheimer’s disease neuroimaging initiative (ADNI): MRI methods,” *J. Magn. Reson. Imaging*, vol. 27, no. 4, pp. 685–691, Apr. 2008, doi: 10.1002/jmri.21049.

[123] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2013.

[124] F. A. Garner, “Radiation-Induced Damage in Austenitic Structural Steels Used in Nuclear Reactors,” in *Comprehensive Nuclear Materials*, 2nd ed., vol. 3, Elsevier, 2020, pp. 57–168.

[125] M. Jin, P. Cao, and M. P. Short, “Predicting the onset of void swelling in irradiated metals with machine learning,” *J. Nucl. Mater.*, vol. 523, pp. 189–197, Sep. 2019, doi: 10.1016/J.JNUCMAT.2019.05.054.

[126] L. K. Bachrach, T. Hastie, M.-C. Wang, B. Narasimhan, and R. Marcus, “Bone Mineral Acquisition in Healthy Asian, Hispanic, Black, and Caucasian Youth: A Longitudinal Study,” *J. Clin. Endocrinol. Metab.*, vol. 84, no. 12, pp. 4702–4712, Dec. 1999, doi: 10.1210/JCEM.84.12.6182.

[127] M. Kim, C. Song, and K. Liu, “Individualized Degradation Modeling and Prognostics in a Heterogeneous Group via Incorporating Intrinsic Covariate Information,” *IEEE Trans. Autom. Sci. Eng.*, vol. In press, pp. 1–16, 2021, doi: 10.1109/TASE.2021.3070532.

[128] J.-L. Wang, J.-M. Chiou, and H.-G. Müller, “Review of Functional Data Analysis,” *Annu. Rev. Stat. Its Appl.*, vol. 3, no. 1, pp. 257–295, 2016, doi: 10.1146/)).

[129] J. Mercer, “Functions of positive and negative type, and their connection with the theory of integral

equations," *Proc. R. Soc. London. Ser. A, Contain. Pap. a Math. Phys. Character*, vol. 83, no. 559, pp. 69–70, Nov. 1909, doi: 10.1098/RSPA.1909.0075.

[130] H. Cardot, "Conditional functional principal components analysis," *Scand. J. Stat.*, vol. 34, no. 2, pp. 317–335, Jun. 2007, [Online]. Available: http://www.jstor.org/stable/41548554.

[131] F. Ding, S. He, D. E. Jones, and J. Z. Huang, "Supervised Functional PCA with Covariate Dependent Mean and Covariance Structure," *arXiv:2001.11425*, pp. 1–24, 2020, [Online]. Available: http://arxiv.org/abs/2001.11425.

[132] G. Li, H. Shen, and J. Z. Huang, "Supervised Sparse and Functional Principal Component Analysis," *J. Comput. Graph. Stat.*, vol. 25, no. 3, pp. 859–878, 2016, doi: 10.1080/10618600.2015.1064434.

[133] C. R. Jiang and J. L. Wang, "Covariate adjusted functional principal components analysis for longitudinal data," *Ann. Stat.*, vol. 38, no. 2, pp. 1194–1226, 2010, doi: 10.1214/09-AOS742.

[134] F. Ding, S. He, D. E. Jones, and J. Z. Huang, "Functional PCA With Covariate-Dependent Mean and Covariance Structure," *Technometrics*, vol. 0, no. 0, pp. 1–29, 2022, doi: 10.1080/00401706.2021.2008502.

[135] J. O. Ramsay and C. Dalzell, "Some tools for functional data analysis," *J. R. Stat. Soc. Ser. B*, vol. 53, no. 3, pp. 539–561, 1991.

[136] W. Guo, "Functional mixed effects models," *Biometrics*, vol. 58, pp. 121–128, 2002.

[137] J. Li, C. Huang, and Z. Hongtu, "A Functional Varying-Coefficient Single-Index Model for Functional Response Data," *J. Am. Stat. Assoc.*, vol. 112, no. 519, pp. 1169–1181, 2017, doi: 10.1080/01621459.2016.1195742.

[138] F. Yao, H. G. Müller, and J. L. Wang, "Functional data analysis for sparse longitudinal data," *J. Am. Stat. Assoc.*, vol. 100, no. 470, pp. 577–590, 2005, doi: 10.1198/016214504000001745.

[139] J. Fan and I. (Irène) Gijbels, "Local polynomial modelling and its applications," p. 341, 1996.

[140] J. A. Rice and B. W. Silverman, "Estimating the Mean and Covariance Structure Nonparametrically When the Data are Curves," *J. R. Stat. Soc. Ser. B*, vol. 53, no. 1, pp. 233–243, 1991, [Online]. Available: http://www.jstor.org/stable/2345738.

[141] S. Chung and R. Kontar, "Functional Principal Component Analysis for Extrapolating Multi-stream Longitudinal Data," *IEEE Trans. Reliab.*, pp. 1–11, 2020, doi: 10.1109/TR.2020.3035084.

[142] R. M. Neal, "Bayesian learning for neural networks," Ph.D. Thesis, University of Toronto, 1995.

[143] H. Zhang, "Inconsistent Estimation and Asymptotically Equal Interpolations in Model-Based Geostatistics," *J. Am. Stat. Assoc.*, vol. 99, no. 465, pp. 250–261, 2004, doi: 10.1198/016214504000000241.

[144] R. M. Neal, "Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification," no. 9702, pp. 1–24, 1997.

[145] C. E. Rasmussen, "Evaluation of Gaussian Processes and other Methods for Non-Linear Regression," Ph.D. Thesis, Department of Computer Science, University of Toronto, 1996.

[146] M. D. Hoffman and A. Gelman, "The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1593–1623, 2014.

[147] C. Linkletter, D. Bingham, N. Hengartner, D. Higdon, and K. Q. Ye, "Variable selection for Gaussian process models in computer experiments," *Technometrics*, vol. 48, no. 4, pp. 478–490, 2006, doi: 10.1198/004017006000000228.

[148] H. Moon, A. M. Dean, and T. J. Santner, "Two-stage sensitivity-based group screening in computer experiments," *Technometrics*, vol. 54, no. 4, pp. 376–387, 2012, doi: 10.1080/00401706.2012.725994.

[149] Y. Wu, D. D. Boos, and L. A. Stefanski, "Controlling Variable Selection by the Addition of Pseudovariables," *J. Am. Stat. Assoc.*, vol. 102, no. 477, pp. 235–243, Jun. 2007, [Online]. Available: http://www.jstor.org/stable/27639835.

[150] B. W. Silverman, "Density Estimation for Statistics and Data Analysis," *Density Estim. Stat. Data Anal.*, pp. 1–175, Feb. 1998, doi: 10.1201/9781315140919.

[151] A. L. M. Dekkers, J. H. J. Einmahl, and L. De Haan, "A moment estimator for the index of an extreme-value distribution," *Ann. Stat.*, pp. 1833–1855, 1989.

[152] H. Wang and Y. Xia, “Shrinkage Estimation of the Varying Coefficient Model,” *J. Am. Stat. Assoc.*, vol. 13, no. 2, pp. 76–87, 2017.
[153] G. S. Was, “Fundamentals of radiation materials science : metals and alloys,” p. 827, 2007.
[154] D. P. Mccormick, S. W. Ponder, H. D. Fawcett, and J. L. Palmer, “Spinal bone mineral density in 335 normal and obese children and adolescents: Evidence for ethnic and sex differences,” *J. Bone Miner. Res.*, vol. 6, no. 5, pp. 507–513, May 1991, doi: 10.1002/JBMR.5650060513.
[155] D. N. Patel, J. M. Pettifor, P. J. Becker, C. Grieve, and K. Leschner, “The effect of ethnic group on appendicular bone mass in children,” *J. Bone Miner. Res.*, vol. 7, no. 3, pp. 263–272, Mar. 1992, doi: 10.1002/JBMR.5650070304.
[156] A. C. Looker, L. J. Melton, T. Harris, L. Borrud, J. Shepherd, and J. McGowan, “Age, gender, and race/ethnic differences in total body and subregional bone density,” *Osteoporos. Int.*, vol. 20, no. 7, pp. 1141–1149, Jul. 2009, doi: 10.1007/S00198-008-0809-6.
[157] J. Piironen and A. Vehtari, “PROJECTION PREDICTIVE MODEL SELECTION FOR GAUSSIAN PROCESSES,” 2016.
[158] J.-M. Chiou, Y.-T. Chen, and Y.-F. Yang, “MULTIVARIATE FUNCTIONAL PRINCIPAL COMPONENT ANALYSIS: A NORMALIZATION APPROACH,” *Stat. Sin.*, vol. 24, no. 4, pp. 1571–1596, 2014, [Online]. Available: http://www.jstor.org/stable/24310959.