Robust and Data-Driven Uncertainty Quantification Methods as Real-Time Decision Support in Data-Driven Models

Pooja Algikar

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Lamine Mili, Chair
Almuatazbellah Boker
Ming Jin
Virgilio Centeno
Konstantinos Triantis

December 2, 2024
Falls Church, VA

Keywords: Gaussian processes, Koopman operator, Huber likelihood, Robust estimation, Statistical moments.

Robust and Data-Driven Uncertainty Quantification Methods as Real-Time Decision Support in Data-Driven Models

Pooja Algikar

(ABSTRACT)

The growing complexity and data in modern engineering and physical systems require robust frameworks for real-time decision-making. Data-driven models trained on observational data enable faster predictions but face key challenges—data corruption, bias, limited interpretability, and uncertainty misrepresentation—which can compromise their reliability. Propagating uncertainties from sources like model parameters and input features is crucial in data-driven models to ensure trustworthy predictions and informed decisions. Uncertainty quantification (UQ) methods are broadly categorized into surrogate-based models, which approximate simulators for speed and efficiency, and probabilistic approaches, such as Bayesian models and Gaussian processes, that inherently capture uncertainty into predictions. For real-time UQ, leveraging recent data instead of historical records enables more accurate and efficient uncertainty characterization, making it inherently data-driven. In dynamical analysis, the Koopman operator represents nonlinear system dynamics as linear systems by lifting state functions, enabling data-driven estimation through its applied form. By analyzing its spectral properties—eigenvalues, eigenfunctions, and modes—the Koopman operator reveals key insights into system dynamics and simplifies control design. However, inherent measurement uncertainty poses challenges for efficient estimation with dynamic mode and extended dynamic mode decomposition algorithms. This dissertation develops a statistical framework to propagate measurement uncertainties in the elements of the Koopman operator. This dissertation also develops robust estimation of model parameters, considering observational

data, which is often corrupted, in Gaussian process settings. The proposed approaches adapt to evolving data and process agnostic— in which reliance on predefined source distributions is avoided.

Robust and Data-Driven Uncertainty Quantification Methods as Real-Time Decision Support in Data-Driven Models

Pooja Algikar

(GENERAL AUDIENCE ABSTRACT)

Modern engineering and scientific systems are increasingly complex and interconnected—operating in environments with significant uncertainties and dynamic changes. Traditional mathematical models and simulations often fall short in capturing the complexity of large-scale real-world, ever-evolving systems—struggling to adapt to dynamic changes and fully utilize today's data-rich environments. This is especially critical in fields like renewable integrated power systems, robotics, etc., where real-time decisions must account for uncertainties in the environment, measurements, and operations. The growing availability of observational data—enabled by advanced sensors and computational tools—has driven a shift toward data-driven approaches. Unlike traditional simulators, these models are faster and learn directly from data. However, their reliability depends on robust methods to quantify and manage uncertainties, as corrupted data, biases, and measurement noise challenge their accuracy. This dissertation focuses on characterizing uncertainties at the source using recent data, instead of relying on assumed distributions or historical data, as is common in the literature. Given that observational data is often corrupted by outliers, this dissertation also develops robust parameter estimation within the Gaussian process setting. A central focus is the Koopman operator theory—a transformative framework that converts complex, nonlinear systems into simpler, linear representations. This research integrates measurement uncertainty quantification into Koopman-based models, providing a metric to assess the reliability of the Koopman operator under measurement noise.

Dedication

To my parents,

Nanda Algikar and Basavaraj Algikar.

Acknowledgments

I would like to begin by expressing my heartfelt gratitude to my advisor, Dr. Lamine Mili, whose invaluable guidance and support have profoundly shaped my approach to this research.

I am deeply thankful to Dr. Rui Yang and my mentor, Dr. Pranav Sharma, for giving me the incredible opportunity to intern with the data analytics research group at the National Renewable Energy Laboratory in Golden, Denver. Their insights, along with engaging discussions with Dr. Marcos Netto on Koopman operator theory, strengthened my understanding and laid a solid foundation for the proposed methods. I am especially grateful to Dr. Netto for his unwavering support and willingness to provide guidance whenever I sought it.

I also extend my sincere appreciation to Professors Ming Jin, Virgilio Centeno, Almuatazbellah Boker, and Konstantinos P. Triantis for serving on my committee.

I am fortunate to have been surrounded by wonderful friends and colleagues. My deepest thanks go to Shailik Sarkar, Vasanth Reddy, and my labmates Somayeh Yarahamadi, Jaber Valinejad, and Yijun Xu for their support and camaraderie. I am also grateful to Roxanne Paul for her unwavering friendship and for making my time at Virginia Tech so memorable.

Above all, I am profoundly thankful to my husband, Padmaksha Roy, for his constant encouragement, late-night pep talks, and willingness to shoulder extra responsibilities so that I could pursue my dreams.

Finally, I am eternally grateful to my family—their love, sacrifices, and unwavering faith in me have been a source of endless inspiration. Their presence in my life is a blessing I will always cherish.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Physical, scientific, and engineering systems must effectively extract insights from observations as their scale increases, especially when running mathematical models on simulators becomes computationally demanding [31]. In such cases, using simulators for making operational and control decisions in real time is impractical. For e.g. consider a power distribution system: the network expands daily with the addition of numerous customers, the installation of photovoltaic systems (PVs), and the deployment of electric vehicle charging stations. Solving differential equations involved in dynamical analysis (such as state estimation, time domain simulations, etc.) is extremely time consuming and the results may not be even accurate when customer additions or PV installations go unrecorded [132]. In addition to that, the physical components of engineering systems are governed by natural processes; and the associated uncertainty and intermittency needs to be modeled to be prepared for unexpected events. In decisions theory, modeling uncertainty levels enables operators make trustworthy decisions. On the other hand, data-driven models—designed using observational data—are faster at making predictions compared to simulators once trained: yet, without adequately addressing uncertainty, their reliability is compromised. Data-driven models must also address misrepresentations, inadequacies, and corruptions in data, along with uncertainties in measurement processes and biases that may skew their real-world applicability.

Uncertainty quantification (UQ) establishes a probabilistic framework within data-driven models by characterizing the uncertainties of source variables—such as input variables or

system parameters—and modeling their impact on target variables, like system outputs. Uncertainty in target variables can be quantified through variability (e.g., confidence intervals), information measures such as Shannon information and probability distributions [104]. UQ enables informed decision-making by probabilistically modeling uncertainty and the occurrence of extreme events and outliers in uncertain input variables, then propagating that uncertainty through the model to estimate the variability—or ideally, the probability distribution—of uncertain output variables in physical or engineering systems, and vice versa. It can also provide estimates of uncertainty levels for both parametric models (e.g., neural networks) and non-parametric models (e.g., Gaussian processes), as well as for their parameters. System operators or decision-makers can leverage this information on uncertainty, propagated from various sources to targets, to make more uncertainty-aware decisions. For example, in aerospace design, engineers incorporate margins for fuel consumption, aerodynamic drag, and component reliability to account for uncertainties in operating conditions and material properties, thereby enhancing safety and performance in flight systems. Similarly, operators in other engineering domains can use these uncertainty estimates to ensure robust and reliable system designs [100].

There are two main approaches to quantifying uncertainties in this context. The first involves constructing a surrogate model—also known as an emulator, meta-model, or response surface—that statistically approximates the simulator model. This surrogate is typically built by running simulations at carefully selected, space-filling points across the input dimensions and recording the output at each point. Surrogate models serve as inexpensive, fast approximations of the simulator, maintaining accuracy within the chosen design space of inputs. In surrogate-based UQ, Monte Carlo simulations are performed on thousands of samples drawn from an assumed probability distribution of test inputs [122]. The resulting output samples for each input test point can then be analyzed to determine variability,

distribution, or information measures, depending on the chosen metric for assessing uncertainty levels. For example, in transmission spectroscopy, distributional information can aid in estimating atmospheric composition variability [36], while in renewable energy integration in power systems, variability analysis helps operators anticipate fluctuations in generation and maintain grid stability [13]. Common surrogate models include neural networks [152], prized for their flexibility; Gaussian processes, which naturally quantify model uncertainty; and polynomial chaos expansions, known for their customizable basis function selection. For example, in autonomous navigation, NN based-based UQ assesses the reliability of sensor data (e.g., from LiDAR or cameras) in dynamic environments [37]. In power systems, polynomial chaos-based [121] and Gaussian process-based [147]. UQ models facilitate real-time voltage and frequency adjustments, stabilizing the grid in response to fluctuations from renewable energy sources and varying demand.

The second approach involves using inherently probabilistic models, such as Bayesian models, Gaussian processes (GPs), and Markov models, which aim to model the relationship f between $y \in \mathcal{Y}$ and $\boldsymbol{x} \in \mathcal{X}$ based purely on observational data $\mathcal{D} = \{y_i, \boldsymbol{x}_i\}_{i=1}^n$. These data-driven models rely on probability theory to capture uncertainty in the underlying relationships. These data-driven models inherently provide estimates for uncertainty, with parameter, model uncertainty and output uncertainty emerging as natural byproducts of the inference process. This characteristic of Bayesian models makes them highly desirable for UQ, contributing to their widespread popularity in stochastic modeling. This has led to a recent interest in probabilizing neural networks [140] by incorporating Bayesian principles, resulting in models like Bayesian neural networks, proposed by [88] and Bayesian graph neural networks [47]. These approaches make neural networks parametrically probabilistic, enabling them to produce predictive probability distributions.

A well-trained surrogate-model requires training data constituted by running the simulator

model at input space filling data points to obtain output data points as we discussed earlier. As a result, the input data points are sampled from an assumed probability distribution in the design space of stochastic input variables. The specified probability distribution for input variables may not always correspond to real-world scenario. For e.g. assumed probability distributions in constructing a power flow emulator is typically are the Gaussian distribution for the load variables, the Weibull distribution for the wind speed, and the Beta distribution for the solar irradiance, among others. However, in practice, these distributions may not represent the actual data [68, 148], yielding inaccurate uncertainty quantification results. The conventional meta-modeling methods are not designed to handle the misrepresentations (for e.g. of power curve distribution), yielding biased results.

For real-time applications, using historical data to assume a distribution for uncertain input variables can be insufficiently current. Characterizing input uncertainty based on recently observed data often provides a more accurate reflection of the present conditions. For instance, in environments like autonomous vehicles and robotics, uncertainties arise from physical and model limitations, partial observability, changing environmental dynamics, and domain shifts [110]. In these applications, data-driven UQ methods are essential in, where the uncertainties are dynamic, making online UQ essential for rapid, informed decision-making. As a result, there is increasing interest in developing fast, data-driven UQ methods that eliminate the need to specify probability distributions for uncertain inputs when building data-driven or surrogate models. Instead, there is a need to construct these models using data from the most recent time window. With the widespread deployment of sensors across the system network to enhance system observability, measurement data becomes readily available for constructing data-driven models. However, when recent observations are used to build a data-driven UQ model, they may occasionally contain erroneous data—a challenge that is often unavoidable.

In the context of nonlinear dynamical systems, data-driven models such as neural network-based approaches (e.g., neural ODEs [24], recurrent neural networks, LSTMs [120]) and Bayesian methods [30] that approximate the time-series functional relationship between system states are popular; however, they often struggle to provide detailed information specific to the system's dynamics. Koopman operator, by contrast, enables a linear approximation of nonlinear dynamics, allowing for stability assessment through the analysis of its spectral properties: Koopman mode, eigenvalues, and eigenfunctions. Koopman modes show the amplitude and spatial distribution of oscillatory or growing/decaying behavior in different regions of the system. Each eigenvalue indicates the frequency and stability of its corresponding mode. The associated eigenfunctions capture invariant structures of the dynamics.

Koopman operator theory provides a richer, linearized representation of nonlinear dynamics compared to other data-driven models. However, approximating the Koopman operator generator in finite dimensions often relies on weighted least squares based algorithms, which can be non-robust and highly sensitive to measurement quality. As a result, researchers have focused on developing robust estimation techniques for approximating the Koopman operator. For real-time decision-making, quantifying measurement uncertainties from data in the approximated Koopman operator is important to ensure reliable predictions and improve the robustness of system control and stability assessments [75]. For example: a Koopman operator based model represents the complex swing dynamics within interconnected generator networks, essential for identifying coherency [127]. Real-world power system measurements are subject to multiple uncertainties: sensor noise, communication delays, and environmental variations among them. By quantifying these uncertainties and integrating them into the Koopman model, operators can achieve a probabilistic bounds on coherence. When the model indicates high uncertainty in a coherence group's stability, operators can take proactive steps to prevent potential system instability. In many robotic systems, real-time

control solutions are essential. Linear representations enable control of nonlinear systems using linear optimal control tools, which are typically faster and easier to implement than nonlinear methods. The challenge lies in balancing dimensionality with modeling accuracy by finding the minimum number of basis functions needed for desired precision [75]. Here, model uncertainty can indicate the reliability of online control decisions.

1.1 Research objective and significance

My research aims to quantify uncertainty purely using recorded data. Traditionally, the models aiming to quantify uncertainty use a probability density functions of the standard probability distributions to describe the uncertainty in the input arguments, which isn't accurate all the time for a wide range of applications. We have data available in enormous amount due to advances in the sensing and metering technologies. I aim to leverage the data to precisely characterize the uncertainty in the input arguments, propagate it through the model, and characterize the uncertainties in the output arguments. The accuracy of uncertainty quantification models is highly influential to how accurately we are specifying the uncertainty of the input arguments. As we are aware, measured data gets corrupted with noise almost all the time, I focused on developing robust and data-driven UQ models which has the following essential components differentiating the methods from the literature:

- data-driven characterization of uncertainty of the input arguments.
- parameter estimation in the proposed uncertainty quantification model by advancing on the principles from robust estimation theory.
- time-series estimation of uncertainty bounds for system states.
- confidence levels associated with the Koopman-theoretic data-driven characterization of power systems dynamics.

• measurement uncertainty propagation characterized by analytical expressions for the second moments of the Koopman operator using random matrix theory.

1.2 Summary of achievements

To date, this research has resulted in the following publications:

• Algikar, P., Xu, Y., Yarahmadi, S., & Mili, L. (2023). A Robust Data-driven Process Modeling Applied to Time-series Stochastic Power Flow. IEEE Transactions on Power Systems.
• Algikar, P., Xu, Y., & Mili, L. (2022, July). A Measurement-Based Robust Non-Gaussian Process Emulator Applied to Data-Driven Stochastic Power Flow. In 2022 IEEE Power & Energy Society General Meeting (PESGM) (pp. 01-05). IEEE.
• Algikar, P., & Mili, L. (2023). Robust Gaussian Process Regression with Huber Likelihood. International Conference on Machine Learning (under review).
• Algikar, P., & Mili, L. (2023). Robust Hyperparameter Estimation in Gaussian Process Regression Model. Statistics and Computing (under review).
• Algikar, P., Sharma, P., Netto M., & Mili, L. Measurement Uncertainty Impact on Koopman Operator Estimation of Power System Dynamics. IEEE Transactions on Power Systems (under review).
• Algikar, P., Sharma, P., Netto M., & Mili, L. Measurement Uncertainty Impact on Koopman Operator Estimation of Power System Dynamics. IEEE Transactions on Power Systems (under review).
• Algikar, P., Sharma, P., Netto M., & Mili, L. Trustworthy Koopman Operator for Elementwise Uncertainty Analysis. IEEE Control Systems Letters (under review).

1.3 Organization

This dissertation proceeds as follows.

Chapter 2 explores the evolution of data-driven methods, emphasizing advancements in uncertainty quantification and their integration into observational data-driven approaches.

Chapter 3 focuses on robust process modeling within the framework of Gaussian Processes, with robust estimation of hyperparameters.

Chapter 4 presents Bayesian inference for Gaussian process regression, using the Huber likelihood to handle noisy data in covariate and response dimensions. It develops Gibbs sampling and Laplace approximation methods, with theoretical proof of the posterior distribution's unimodality.

Chapter 5 addresses measurement uncertainty quantification in Koopman-theoretic, data-driven approximations of nonlinear dynamics. It introduces statistical methods for computing the first and second moments of the DMD operator elements using Sawa integration techniques and moment-generating functions.

Chapter 6 delves into element-wise computation of the first and second moments of the Koopman operator using random matrix theory in DMD and EDMD approximations. The chapter highlights the benefits of zero-centered data and provides straightforward steps for calculating moments, offering analytical insights into the uncertainty quantification of diagonal and off-diagonal elements through their second moments.

Chapter 7 concludes with a discussion of future directions and key takeaways

Chapter 2

Literature review

2.1 Data-driven methods

Data-driven methods are approaches that purely use observational data to develop approximation models, make predictions, and inform decisions without being explicitly learned with field specific knowledge. These methods use statistical methods, machine learning, and AI to extract patterns, relationships, and insights from large datasets. Examples include fuzzy and rough sets for handling uncertainty [44], neural networks for approximating functions [48], global optimization and evolutionary computing [45], statistical learning theory [136], and Bayesian methods [50]. They are particularly effective for complex systems with interconnected behaviors that are difficult to predict, analyze, or control due to limited understanding or inherent variability, especially where traditional models are insufficient or challenging to construct. These models have evolved from earlier statistical models, which were based on certain assumptions about probability distributions of error in model fitting that often proved to be overly restrictive [39].

2.1.1 Statistical data-driven methods

Statistical inference approaches can be broadly categorized into model-driven and data-driven methods. Model-driven analyses rely on established probabilistic or mathematical

models, such as linear regression or analysis of variance, that assume key conditions like normality, homogeneity, or additive effects. These models provide a structured framework but can be restrictive, requiring validation of assumptions and often complex interpretation. On the other hand, data-driven analyses have three main subcategories: exploratory techniques, robust methods, and diagnostic tools. Exploratory techniques like principal component analysis [92] or clustering algorithms are less assumption-driven and aim to uncover patterns within the data. Robust methods such as robust regression with Huber estimators [124] are designed to provide reliable results even when standard distributional assumptions of error are not met, handling outliers or non-normal distributions more effectively. diagnostic tools [9] help assess model fit and assumptions; examples include residual analysis, cross-validation, or permutation tests.

The Bayesian approach to data analysis dates to the Reverend Thomas Bayes [11]. Initially, Bayesian computations were difficult except for simple examples, and applications of Bayesian methods were uncommon until Adrian F. M. Smith [116, 117] began to spearhead applications of Bayesian methods to real data. Bayesian applications to science and medicine have exploded between 1990 and 2010, thanks to the development of flexible and robust computational algorithms such as Markov chain Monte Carlo [41, 42]. More closer to data-driven way, empirical Bayesian methods take a data-driven approach: unlike classical Bayesian methods, they do not assume a fixed prior distribution. Instead, they learn the prior and its parameters directly from the data by maximizing the marginal likelihood [15, 22]. Bayesian network (BN) proposed by Pearl et.al.[91], which used graph theory and often Bayesian statistics to allow machines to make plausible hypotheses when given uncertain or fragmentary information. BNs are widely studied for uncertainty quantification, reliability analysis, and model calibration in engineering systems under uncertainty [105, 149]. These applications depend on established physics models or known causal dependency re-

lationships. Data-driven BNs found in the literature [86, 137, 153] are primarily suited for discrete variables. For continuous variables, alternative approaches have been proposed [12, 46, 52, 113]. Particularity influential in time-series analysis, observations Y dependent Markov processes on latent variables f, hidden Markov models, [1] offer an explanation on the sequential occurrence of f by observing y, since f cannot be observed directly [10].

2.1.2 Machine learning methods

Data-driven methods and machine learning are closely intertwined, and whether one is a subset of the other often depends on subjective interpretation. In this dissertation, machine learning is viewed as a subset of data-driven methods. A model that disregards field-specific feature generation and focuses solely on learning functional dependencies $f(\cdot)$ from observed input-output data $\mathcal{D} = x \subseteq \mathcal{X}, y \subseteq \mathcal{Y}$, with the primary objective of minimizing test errors, is classified as a machine learning model.

Recently, we see a surge of data-driven methods on solving ill-posed inverse problems, meaning that small errors in data may lead to large errors in the model parameter, or there are several possible model parameter values that are consistent with observations [6]. Data-driven methods, particularly those from machine learning, offer alternative approaches to address the limitations of analytical models in inverse problems. Deep learning, known for its transformative impact on tasks like computer vision and speech recognition [66], adapts generic models to specific problems through training data without relying on prior knowledge. However, an entirely data-driven approach face challenges in scientific applications due to insufficient training data and the need for robustness, which seriously limits their effectiveness in solving complex inverse problems. For example, in inverse power flow within power distribution systems, traditional methods rely on accurate forward models to esti-

[1] A nice representation of Bayesian networks and Hidden Markov Models is available in [86]

mate system states like voltages and power flows. However, these models often simplify real-world complexities or rely on incomplete system information, leading to approximations that may not fully capture the behavior of distributed energy resources or grid dynamics. Additionally, real-world inputs, such as sensor measurements, are often noisy, incomplete, or biased, creating further challenges. Techniques leveraging sparsity or statistical assumptions improve estimation, yet they too struggle with highly dynamic, irregular, or site-specific variations. These factors necessitate more adaptable, data-driven, or hybrid approaches to handle evolving complexities in modern power grids effectively.

In the context of state estimation in engineering systems, data-driven methods often act as "black-box" models, meaning they focus on learning relationships between input and output variables directly from measurements without explicitly incorporating the underlying physical or mathematical rules of the system. While these models—like neural networks or deep learning algorithms—can effectively capture complex patterns, their interpretability is limited compared to traditional physics-based models [101]. This trade-off between flexibility and interpretability is a key consideration when using data-driven approaches for state estimation, particularly in critical applications. For example, in power systems, a critical application of state estimation is real-time monitoring for fault detection and isolation. State estimation algorithms continuously assess the voltage and current levels throughout the grid. In case of abnormalities, like a sudden voltage drop or current spike, these algorithms can quickly identify the affected lines or components. Accurate state estimation enables operators to detect faults and isolate them to prevent cascading failures, which could otherwise lead to large-scale blackouts.

Machine learning methods—particularly deep neural networks—are highly effective at minimizing loss and achieving a broad fit across data. They manage overfitting and underfitting with various regularization techniques embedded in the loss functions for regression and clas-

sification tasks. As a result, these models have advanced significantly in machine learning, focusing on data-driven fits. However, in applying them to real-world scientific or engineering systems, domain experts must carefully assess which tasks can be assigned to machine learning, given its capability for rapid approximation. Some tasks can be managed using observed data alone, while others still depend on system-specific solvers. Ultimately, these models offer the potential to approximate almost any function, provided there is sufficient data.

To incorporate system-specific constraints and achieve parameter optimization that aligns with the limitations imposed by physical systems, an emerging branch called physics-informed machine learning combines data-driven approaches with knowledge of physical laws [59]. This approach leverages computer models or simulators to fill data gaps and aims to approximate functions efficiently. When data is insufficient or as part of the model's function, it integrates simulator runs to maintain accuracy while respecting system constraints.

2.1.3 Physics informed ML

Machine learning has emerged as a promising alternative in solving partial differential equations (PDEs) for multiphysics, but training deep neural networks requires big data, not always available for scientific problems. Instead, such networks can be trained from additional information obtained by enforcing the physical laws (for example, at random points in the continuous space-time domain), thus emerging a branch of ML called physics informed ML is widely looked up for scientific applications. Such physics-informed learning integrates (noisy) data and mathematical models, and implements them through neural networks or other kernel-based regression networks. Moreover, it may be possible to design specialized network architectures that automatically satisfy some of the physical invariants for better ac-

curacy, faster training and improved generalization [59]. There is a need for developing new frameworks and standardized benchmarks as well as new mathematics for scalable, robust and rigorous next-generation physics-informed learning machines.

Traditionally, an inverse problem is formalized as solving an equation of the form

$$y = \mathcal{A}(f) + \epsilon. \tag{2.1}$$

Here $y \in \mathcal{Y}$ is the measured data, assumed to be given, and $f \in \mathcal{X}$ is the model parameter we aim to reconstruct. In many applications, both y and f are elements in appropriate function spaces $\mathcal{Y}$ and $\mathcal{X}$, respectively. The mapping $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is the forward operator, which describes how the model parameter gives rise to data in the absence of noise and measurement errors, and $\epsilon \in \mathcal{Y}$ is the observational noise that constitutes random corruptions in the data g. The above view constitutes a knowledge-driven approach, where the forward operator and the probability distribution of the observational noise are derived from first principles. Focusing on , the forward model is always an approximate description of reality, and extending it might be challenging due to a limited understanding of the underlying physical or technical setting. Ideally one uses explicit knowledge-driven models when there are such available, and learns models from example data using data-driven methods only when this is necessary.

2.1.4 Gaussian processes

Gaussian Processes (GPs) occupy a unique space within the intersection of data-driven statistical methods and machine learning. As probabilistic models, GPs rely on a prior over a set of functions $f_1, f_2, \ldots, f_n$ (can be indexed by space or time) making them more aligned with model-driven approaches theoretically. Therefore, they can be used to deduce

estimations of latent functions at test points conditional on training points as well as to learn the joint distributions of parameters. GPs are universal in a sense that they can be defined over an input domain indexed by time (in the case of temporal processes) or space (in the case of spatial processes). The covariance functions (or kernels) in GPs are adapted based on data, which helps GPs capture dependencies without needing strict assumptions about the underlying system's functional form.

In a machine learning context [142], GPs are often used for regression and classification tasks, where their adaptability and non-parametric nature allow them to model complex relationships. For example, in regression, GPs use training data to learn a flexible mapping between input variables X and target variables Y, with minimal domain-specific assumptions. The training process involves learning hyperparameters by maximizing the marginal likelihood, which makes GPs effective at adjusting their covariance structures based on empirical data.

At the same time, GPs provide uncertainty estimates through their probabilistic framework, making them very useful in applications where confidence in predictions is critical. This combination of data adaptability and probabilistic structure allows GPs to function well in both data-driven and machine learning paradigms. For the purposes of this dissertation, Gaussian Processes are one of the primary topics of focus. A detailed framework for GPs is provided in Appendix A.

2.2 Data-driven methods in applied Koopman operator analysis

Real-world systems are complex and nonlinear. Identifying unknown dynamics from data and learning intrinsic coordinates that enable a linear representation of the underlying nonlinear

dynamics are two of the most pressing goals of modern dynamical systems [18]. The Koopman operator approach has found tremendous success in these goals [80]. In simple words, the Koopman operator transforms a finite-dimensional nonlinear dynamical system into an infinite-dimensional linear system. This transformation enables the use of well-developed principles and tools in linear algebra to study nonlinear dynamical systems without neglecting nonlinearities [17]. Dynamic mode decomposition (DMD) [107], one of the numerical methods associated with the Koopman operator [98], has been widely adopted. Indeed, DMD has been applied to robotic systems [16], traffic control systems [7], power grids [112], and other engineering systems where sensor measurements are used for real-time operations and control. For example, in power grids, data obtained from instruments called phasor measurement units can be used to assess the stability of the system without having to solve nonlinear differential equations [126], identify groups of synchronous generators oscillating coherently after a disturbance [125], and quantify the interplay between state variables and oscillatory modes [89].

This chapter briefly introduces Koopman operator theory, followed by the presentation of DMD, extended DMD (EDMD), and the kernel variant of EDMD formulations. The impact of data uncertainty on (E)DMD approximations of the Koopman operator is discussed, along with a review of existing methods in the literature that address these challenges.

2.2.1 Koopman operator theory

Let us consider the following dynamical system as an example:

$$f(\boldsymbol{x}) = \dot{\boldsymbol{x}}. \tag{2.2}$$

Here, we assume that the state space $\boldsymbol{x} \in \mathcal{M} \subseteq \mathbb{R}^n$ of the dynamical system forms a linear vector space with certain geometrical properties and that the rule of evolution has some degree of regularity, which is denoted as $f(\cdot)$. Let us assume that the solution to (2.2) exists. Let us define the flow map, $F^t(\boldsymbol{x})$, which takes $\boldsymbol{x}$ from its initial state, $\boldsymbol{x}_0$, to the state $\boldsymbol{x}^t$ at time t given by

$$F^t(\boldsymbol{x}_0) = \boldsymbol{x}_0 + \int_{\boldsymbol{x}=\boldsymbol{x}_0, t'=t_0}^{t'=t} f(\boldsymbol{x}(t_0))\, dt'. \tag{2.3}$$

The Koopman operator lifts the dynamics of the system from a finite-dimensional nonlinear state space to an infinite-dimensional observable linear space. Consequently, we project the dynamics from the state space $\mathcal{M}$ by $g(F^t(\boldsymbol{x}))$ to an infinite observable space to get the linear rule of evolution defined by the Koopman operator, $\mathbf{U}^t$. Formally, we have an expression for the evolution of the system from time t_0 to t given by:

$$g(\boldsymbol{x}_t) = g(F^t(\boldsymbol{x}_0)) = \boldsymbol{U}^t g(\boldsymbol{x}_0). \tag{2.4}$$

The eigenfunctions, ϕ_j, associated with the j^{th} eigenvalue, λ_j, of the Koopman operator, $\mathbf{U}^t$, are defined as:

$$\boldsymbol{U}^t \phi_j = e^{\lambda_j t} \phi_j. \tag{2.5}$$

Now, we assume that all the observables lie in the linear span of the Koopman eigenfunctions, which are the right eigenvectors of the Koopman operator; therefore, the observables can be constructed as a linear combination of eigenfunctions given by

$$g(\mathbf{x}) = \sum_{k=0}^{\infty} c_k \phi_k(\mathbf{x}), \tag{2.6}$$

yielding $\boldsymbol{U}^t g(\boldsymbol{x}) = \sum_{k=0}^{\infty} c_k e^{\lambda_k t} \phi_k(\mathbf{x})$. The coefficients c_k, of these linear expansions are called Koopman modes, which inform us of the shape or structure within the data that evolves

with the eigenvalues of the Koopman operator, λ_k. Note that the principal eigenfunctions are not all the right eigenvectors of the Koopman operator but only those that are linear combinations of observables.

The infinite-dimensional Koopman operator, $\boldsymbol{U}^t$, can nonetheless capture the full nonlinear dynamics, but applied analysis demands a finite approximation of $\boldsymbol{U}^t$, hereafter denoted by $\boldsymbol{A}$.

2.2.2 Applied Koopman operator

Numerous algorithms have been proposed to approximate the Koopman operator from data [20, 99, 134], with dynamic mode decomposition (DMD) [108] and extended DMD (EDMD) [145] being prominent examples. This chapter focuses on the algorithmic steps of DMD, a method that analyzes system dynamics by extracting dominant modes and their associated temporal dynamics from snapshot data.

DMD

DMD [107] is a class of numerical methods with connections to the Koopman operator formalism [98] widely used to perform data-driven analysis of nonlinear dynamical systems [8]. Suppose we have $m+1$ data snapshots $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{m+1}$ sampled from a continuous-time system at instances $t_1, \ldots, t_{m+1}$. Let us organize $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m]$ and $\boldsymbol{Y} = [\boldsymbol{x}_2, \ldots, \boldsymbol{x}_{m+1}]$. An estimate of the DMD operator, assuming a full-state observable, is given by

$$\boldsymbol{A} \approx \boldsymbol{X}^{\dagger}\boldsymbol{Y}, \tag{2.7}$$

where $\boldsymbol{X}^{\dagger}$ is the Moore-Penrose pseudoinverse of $\boldsymbol{X}$.

EDMD

EDMD, a generalized version of DMD, broadens its applicability to nonlinear systems by introducing a set of nonlinear functions or observables, $\boldsymbol{\psi} : \mathbb{R}^n \to \mathbb{R}^{N_k}$. Unlike DMD, which operates directly on the data, EDMD uses these observables to provide a more flexible representation of the system's dynamics. The EDMD operator is given by:

$$\boldsymbol{A} = \boldsymbol{G}^{\dagger}\boldsymbol{L}. \tag{2.8}$$

where $\boldsymbol{G} = \sum_{i=1}^{T} \boldsymbol{\psi}(\boldsymbol{x}_i)\boldsymbol{\psi}(\boldsymbol{x}_i)^T$ and $\boldsymbol{L} = \sum_{i=1}^{T} \boldsymbol{\psi}(\boldsymbol{x}_i)\boldsymbol{\psi}(\boldsymbol{y}_i)^T$.

Kernel EDMD

The kernel trick is a common technique for efficiently computing inner products in an implicitly defined reproducing kernel Hilbert space. A kernel function has the form $k : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$, and is associated with a *feature map* $\boldsymbol{\psi}$, which maps from state space to feature space. The relationship between k and $\boldsymbol{\psi}$ is that $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\psi}(\boldsymbol{x}_i)^{\top}\boldsymbol{\psi}(\boldsymbol{x}_j)$ for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{M}$. Gaussian kernels have the form:

$$k(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{z}\|^2}{\sigma^2}\right), \tag{2.9}$$

which is commonly used in Gaussian processes for machine learning applications [109]. σ is a scaling parameter that is dependent on the characteristic length scales of the underlying problem.

For kernel variant of EDMD [143], using the data set of snapshot pairs and the kernel

function, k, compute the elements of $\hat{\boldsymbol{G}}$ and $\hat{\boldsymbol{L}}$ using:

$$\hat{\boldsymbol{G}}^{(ij)} \triangleq k(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad \hat{\boldsymbol{L}}^{(ij)} \triangleq k(\boldsymbol{y}_i, \boldsymbol{x}_j). \tag{2.10}$$

2.2.3 UQ in applied Koopman operator

Like any weighted least squares method, (E)DMD is susceptible to the quality of the measurements [34]. The noise and uncertainty associated with the measurements directly impact the accuracy of analytical conclusions and control decisions. Therefore, when the DMD method is applied to experimental or field data, one must account for the inherent uncertainty associated with measurements. Nüske et al. [90] derived probabilistic bounds on the extended dynamic mode decomposition [144]. They examine systems described by ordinary and stochastic differential equations from an ergodic and independent identically distributed sample perspective. To formulate theoretical constraints on error, the authors split the variance of the random matrix that forms the basis functions into the terms associated with the asymptotic contribution and the number of data points. Interestingly, the approximation error is quantified using the Frobenius norm between the Galerkin projection of the Koopman generator onto the space of observables and the operator computed from finite samples. The Frobenius norm measures the overall difference between the two representations. However, its insensitivity to structural errors may limit the ability to understand local uncertainty.

Colbrook et al. [25] enhanced the reliability of dynamic mode decomposition by showing a rigorous convergence on the spectral information of Koopman operators from trajectory data for chaotic systems. Zhang et. al. [150] set boundaries on the approximation error of the extended DMD method but do not account for measurement noise. They determine convergence in terms of the largest and smallest eigenvalues of the stiffness matrix. They

approximate the stiffness matrix that constitutes the dictionary functions with a normal probability distribution. Dawson et al. [29] introduced a formulation for sensor noise as a random matrix. The bias induced by additive noise levels is numerically quantified through the expectation of the DMD operator. This approximation holds only for small noise levels. Finally, the bias correction is proposed by downweighting the expectation of the DMD operator based on the variance of the noise levels.

Duke et al. [34] explore the impact of factors such as the quantity and quality of the data, the signal-to-noise ratio, and the quantity of the ensemble on the accuracy of DMD. Dawson et al. [29] quantify the bias in the estimation of the Koopman operator by the DMD method resulting from additive noise levels in the measurements and propose corrective measures Nüske et al. [90] derive probabilistic bounds on how well the extended DMD algorithm [144] captures the underlying system dynamics. They consider both ergodic and independent identically distributed (i.i.d.) samples for systems described by ordinary and stochastic differential equations. They quantify the approximation error using the Frobenius norm between the Koopman operator estimated from finite samples and the Galerkin projection of the Koopman generator onto the space of observables. In formulating theoretical bounds on error, the authors decompose the variance of the random matrix that constitutes the basis functions into terms related to the asymptotic contribution and the impact of the number of data points. The authors in [150] analyze the convergence of a variant of the generator-extended DMD algorithm. The authors establish bounds for the approximation errors but do not consider measurement noise. Based on the convergence of the coefficients of the approximate stiffness matrix of the dictionary functions to a normal probability distribution, the authors arrive at the convergence of the Koopman operator in the matrix norm in terms of the largest and smallest eigenvalues of the stiffness matrix.

2.3 Uncertainty quantification in data-driven models

Uncertainty Quantification (UQ) is a crucial technique for enhancing the trustworthiness of data-driven models in decision making. The recorded data is often noisy, incomplete, or heterogeneous. By explicit quantification of the variability or confidence in predictions, UQ provides more informed judgments in scientific and physical systems, preventing catastrophic outcomes. For example, in autonomous vehicles, sensor readings from LiDAR or cameras can be uncertain due to environmental factors like fog or glare. Incorporating uncertainty estimates enables the vehicle to act cautiously when confidence in obstacle detection is low, preventing potential collisions. In aerospace, aircraft navigation systems can adjust algorithms based on uncertainties in GPS or inertial measurement readings, improving flight safety during conditions like sensor malfunctions or adverse weather. Similarly, in robotics, UQ allows robots to adaptively plan their movements or grip strength based on uncertain sensor inputs, reducing the likelihood of errors in object manipulation.

2.3.1 Types of uncertianties

It is common to divide uncertainty into two types, *aleatoric* and *epistemic* uncertainty. Aleatoric uncertainty — from the Latin *alea*, meaning a die — refers to uncertainty about an inherently variable phenomenon. Epistemic uncertainty — from the Greek *epistēmē*, meaning knowledge — refers to uncertainty arising from a lack of knowledge [123]. If one has at hand a model for some system of interest, then epistemic uncertainty is often further subdivided into model form uncertainty, in which one has significant doubts that the model is even 'structurally correct', and parametric uncertainty, in which one believes that the form of the model reflects reality well, but one is uncertain about the correct values to use for particular parameters in the model. To a certain extent, the distinction between

epistemic and aleatoric uncertainty is an imprecise one and and repeats the old debate between frequentist and subjectivist (e.g. Bayesian) statisticians.

Aleatoric and epistemic uncertainties can occur together within the same context. If experimental parameters show aleatoric uncertainty (inherent variability) and these parameters are then used in a computer simulation, the surrogate model—like a GP or polynomial chaos expansion—will also exhibit epistemic uncertainty (due to limited data) [94]. In this case, the resulting uncertainty involves a mix of both types, creating a more complex form of uncertainty that can't be purely classified as aleatoric or epistemic, but rather as a broader "inferential uncertainty." For example, in a smart grid or power distribution system integrated with renewable energy, an example of experimental parameters could be the solar irradiance levels or wind speed measurements at different times of the day. These parameters are crucial because they directly impact the power output of solar panels or wind turbines. When used in simulations for grid stability analysis, any variability or errors in these measurements introduce aleatoric uncertainty. If these measurements are then used to train a surrogate model (like a GP) for forecasting or optimization, the model's incomplete understanding or data limitations add epistemic uncertainty.

Measurement errors generally consists of two parts: systematic error (the consistent, repeatable error due to sensor bias, calibration issues, or environmental factors) and random error (drifts, environmental conditions). If one aims to quantify measurement uncertainty, they are ideally quantifying aleatoric uncertainty, while systematic errors are typically managed through sensor calibration and bias correction of the data-driven model.

2.3.2 Forward and inverse uncertainty quantification

In forward UQ (also known as uncertainty propagation), the goal is to quantify the uncertainties in output variable(s) y that arise from uncertainties in input variable(s) $\boldsymbol{x}$ and/or parameters $\boldsymbol{\eta}$. Model uncertainty, $\mathcal{U}(h)$, is quantified in probabilistic models, as described above. Aleatoric uncertainty—associated with measurement error—is captured within the model and its parameters. Surrogate-based methods provide flexibility: they allow specifying any desired distribution for inputs or parameters, enabling UQ in the output variables. Though not fixed, forward UQ generally focuses on computing $\mathcal{U}(y) \mid \mathcal{U}(\boldsymbol{x})$ or $\mathcal{U}(\boldsymbol{\eta})$.

In the inverse UQ problem, we aim to quantify $\mathcal{U}(\eta)$, $\mathcal{U}(f) \mid \mathcal{U}(x)$ or $\mathcal{U}(y)$. This involves assessing the uncertainty of a data-driven model and its parameters given the uncertainty in input and output variable(s).

For the purposes of this dissertation, calibration is omitted from inverse uncertainty propagation to focus on developing methods that support uncertainty-aware decision-making. Model parameters can be calibrated based on uncertainty measures at a later stage, which falls outside the scope of this work.

We refer to this as reverse UQ—quantifying $\mathcal{U}(\eta), \mathcal{U}(f) \mid \mathcal{U}(x)$ or $\mathcal{U}(y)$—where the objective is to deduce the distribution of the model and its parameters given data uncertainty, primarily measurement uncertainty.

2.3.3 Prior research

Monte Carlo sampling methods are the oldest and most widely used approach for forward UQ capable of handling high dimensionality [58]. The idea is to sample from the assumed input distributions multiple times and run the computer simulation to obtain a distribution

of output variables. Quasi-Monte Carlo methods make sure that all regions of the input space are covered as evenly as possible instead of purely random sampling to achieve faster convergence and better coverage of the input space [21]. However, these methods can be computationally demanding. Modern Monte Carlo sampling techniques, such as multilevel, multifidelity, and multimodel approaches, offer more efficient solutions for forward UQ [151].

Latin hypercube sampling achieves similar accuracy to Monte Carlo sampling with fewer samples, reducing computational costs in complex models with many uncertain input variables [81]. It is particularly valuable for constructing training datasets in surrogate-based UQ. Sequential Latin hypercube sampling methods incrementally add sampling points—improving allocation efficiency with each addition [70].

Polynomial Chaos expansions, proposed by Nobert Wiener (with Hermite polynomials) [141] constructs an expansion of the output variable(s) as a sum of orthogonal polynomials with coefficients determined by the input distributions. Input uncertainties can be described using known probability distributions (not necessarily Gaussian [35]). This method directly relates input distributions to output distributions, and the coefficients of the polynomial provide information on how each input affects the output. Recent advances include multi-element polynomial chaos expansions [139] divide the input domain into multiple elements, applying expansion within each element separately. This local approach helps handle strong nonlinearities and discontinuities in the model response. The use of sparse grids for polynomial chaos expansions and stochastic collocation help alleviate the curse of dimensionality by efficiently distributing collocation points [71].

Bayesian neural networks incorporate uncertainty of the weights and inputs of the neural network by placing probability distributions over them [62, 138, 146]. Non-normal distributions can be accommodated with variants: mixture density networks [14], multi-modality in neural processes [4], and mixture models [57]. Regression-based neural networks (NNs) are

increasingly used in safety-critical domains such as computer vision [93], robotics, and control systems [115], where accurately inferring model uncertainty is essential for their broader adoption. Hybrid variational inference techniques where researchers are combining variational inference for more expressive approximations model like mixture models [72, 128, 129] or hierarchical variational inference [95] improve the quality of uncertainty estimates while maintaining scalability. Variational inference with normalizing flows [97] approximates a complex posterior distribution with a simple initial density, which is transformed into a more complex one by applying a sequence of invertible transformations until a desired level of complexity is attained.

Unlike BNNs, each model in deep ensembles [38] is typically initialized with random weight values drawn from a common initialization scheme (like Xavier or He initialization). Therefore, multiple instances of the model (e.g., neural networks) are trained with different random initializations or subsets of the training data aimed to learn slightly different relationships, capturing different aspects of the input-output relationship. The key idea is that this variability reflects the range of possible model outcomes, indicating the uncertainty due to the model itself. A combined predictive mean and variance representing the epistemic uncertainty is the outcome. Bayesian Deep Ensembles combine ideas from Bayesian methods and deep ensembles to better capture epistemic uncertainty. For example is to use Bayesian calibration techniques to refine the ensemble predictions [49]. [82] propose to explicitly regularize ensemble diversity (choosing fewer weak learners for an ensemble) to improve robustness and calibration on in-distribution data as well as under dataset shift for classification tasks.

Bootstrapping [130] is a strongly non-parametric technique that accounts for input uncertainties by creating multiple resampled datasets. The key assumption in bootstrapping is that the original dataset is representative of the population, allowing it to be used as a basis for generating new samples.

Hybrid methods [28, 67, 85]—which integrate deep learning with probabilistic models like GPs—can reduce interpretability due to the complexity of combining non-parametric and parametric representations [140]. These methods often carry the computational burdens of both components (e.g., deep NNs and GPs), making them costly for real-time applications or large datasets.

Surrogate modeling is crucial in forward UQ: it offers simplified, computationally efficient representations of complex simulations, enabling repeated evaluations for sensitivity analyses—key for propagating uncertainties and quantifying their impact on outputs. Traditionally, surrogate models were built on simplified fundamental equations. Over time, data-driven methods like GP regression gained prominence [103]. With advances in computation and vast training datasets, modern deep learning models are increasingly used in surrogate modeling, often outperforming traditional models like linear regression and decision trees [27].

Key challenges include high dimensionality [3], data scarcity, and computational costs, which adaptive sampling techniques help address [135]. Robust UQ remains a priority, with Bayesian methods—such as Bayesian Neural Networks [73, 88]—boosting reliability. Conformal predictions provide a robust, distribution-free, and model-agnostic UQ framework, offering valid prediction intervals without assuming precise data distributions [111]. Additionally, neural network-based surrogate models are gaining traction [133].

2.4 Summary of research gaps

To quantify uncertainty from a specific source within a system, there are two main approaches: one assumes a probability distribution for the uncertain parameter or input variable, while the other, known as data-driven UQ, learns this distribution from historical data.

The first approach often relies on surrogates, or metamodels: by assuming a probability distribution for uncertain inputs, samples are drawn and evaluated on an emulator trained with a limited set of simulator runs, which represents the real-world process. When the prior probability density used to generate input samples does not accurately reflect real-world conditions, surrogate-based approaches are limited in their ability to produce reliable time-series forecasts of prediction uncertainties.

In the machine learning community, UQ is often divided into two main types: aleatoric uncertainty, which is data-driven and captures inherent variability in the data, and epistemic uncertainty, which is model-driven and reflects uncertainty due to limited knowledge about the model. Effective UQ is essential for tasks where both aleatoric and epistemic uncertainties need to be accounted for in predictions.

Many UQ methods in ML rely on assuming a probability distribution for uncertain inputs, especially those rooted in Bayesian and probabilistic frameworks. Bayesian neural networks are sensitive to the choice of priors, learning rates, and other hyperparameters, making them challenging to tune effectively. However, approaches that don't require distributional assumption—such as bootstrapping and deep ensembles—have limitations. While deep ensembles can capture input uncertainties to some extent, they generally only provide the predictive mean and variance, lacking a full representation of the output distribution. Consequently, the ML community continues to explore advanced methods to more fully capture and quantify uncertainties across both data and model components.

Recent advancements (2020–2024) include several innovative approaches:

- Stochastic weight averaging: This method forms an approximate posterior distribution over neural network weights and samples from this Gaussian distribution to perform Bayesian model averaging [74]. It aims to provide a simple, scalable, and general-

purpose approach for uncertainty representation and calibration in deep learning.

- Deep evidential regression: This framework captures both aleatoric and epistemic uncertainties without requiring multiple predictions or sampling-based methods. It places evidence-based priors over the original Gaussian likelihood function and trains the neural network to estimate the hyperparameters of this evidential distribution. A regularization term is added during training to penalize deviations in the model's predicted evidence from the correct output [4].

- Multi-fidelity GPs: Recent advances in GPs combine information from low- and high-fidelity models, enhancing expressiveness with neural network kernels. By training a deep neural network alongside GPs, this approach effectively represents complex input spaces and models multi-scale data efficiently.

- Neural processes (NPs): NPs integrate neural networks with probabilistic processes to model distributions over functions, allowing data-driven learning of both mean and variance functions and efficiently capturing prediction uncertainty [40]. Attentive Neural Processes [61] extend this method by incorporating attention mechanisms to capture spatial and temporal dependencies."

Recent efforts (2020–2024), as discussed above and in Section 2.4, aim to enhance UQ methods in ML by improving computational efficiency, scalability, outlier robustness, and high-dimensional handling. By integrating sparse methods, principal component analysis, adversarial training, normalizing flows, and multi-fidelity hierarchical modeling, these approaches are becoming more practical and accurate for applications in machine learning, engineering, and science—though some fundamental challenges remain:

- Sampling-based methods: Monte Carlo methods often require an impractically large number of samples to converge, particularly for rare events or complex uncertainties.

The assumption of independent input distributions, typical in Monte Carlo sampling, may not hold in real-world cases where input variables are correlated. Furthermore, Latin Hypercube Sampling (LHS) becomes less effective in high-dimensional spaces, compromising coverage and efficiency, and it assumes well-defined input distributions, limiting its utility in cases with poorly understood inputs.

- Polynomial chaos expansion: As the number of input variables increases, the number of polynomial terms grows rapidly, raising computational costs and the risk of overfitting. For highly non-linear or discontinuous input-output relationships, they may require high-order polynomials, further increasing complexity and reducing interpretability.

- Bayesian neural networks (BNNs): BNNs are highly sensitive to the choice of priors, learning rates, and other hyperparameters, making them challenging to tune effectively.

- Deep ensembles: While deep ensembles capture some epistemic uncertainty, they can still be overconfident, especially with out-of-distribution inputs or rare events. Additionally, they lack a principled probabilistic interpretation of uncertainty, as they do not define distributions over weights or predictions.

- Bootstrapping: Bootstrapping, based on resampling observed data, cannot generate samples outside the original dataset's range, limiting its effectiveness for unseen scenarios or out-of-distribution inputs. It can also be computationally expensive for large, complex neural network models.

Chapter 3

Robust data-driven process model

Let us represent a simulator function: $f(\cdot)$, is assumed to be a function of $\mathbf{x}_{t_i} \in \mathcal{X}$ in training interval $\boldsymbol{t} = [1, \ldots, n]$. The output variables are $y_{t_i} = f(\mathbf{x}_{t_i}) + \epsilon_{t_i}$ measured with $\epsilon_{t_i} \sim \mathcal{N}(0, \sigma_0^2)$. In the Gaussian process framework, uncertainty about f is encoded using a Gaussian process prior, which is defined by a mean function $m(\cdot)$ and a covariance (kernel) function $k(\cdot, \cdot)$. As a Gaussian process distribution, $f(\mathbf{x}_{t_1}), f(\mathbf{x}_{t_2}), \ldots, f(\mathbf{x}_{t_n})$ follow a multivariate normal distribution. Formally, we have

$$f(\cdot)|\, \boldsymbol{\beta}, \boldsymbol{l}, \tau^2 \sim \mathrm{GP}(m(\cdot), k(\cdot, \cdot)), \tag{3.1}$$

where the mean function $m(\cdot)$ takes the form

$$m(\mathbf{x}) = \boldsymbol{h}(\mathbf{x})^\top \boldsymbol{\beta}. \tag{3.2}$$

Here, $\boldsymbol{h}(\mathbf{x}_{t_i}) : \mathbb{R}^p \to \mathbb{R}^q$ denotes the basis function that can be chosen to model the assumed degree of non-linearity about the underlying process. For example, the quadratic basis functions are expressed as $\boldsymbol{h}(\mathbf{x}) = [1, x_{t_{i1}}, \ldots, x_{t_{ip}}, x_{t_{i1}}^2, \ldots, x_{t_{ip}}^2]$.

The kernel function $k(\mathbf{x}_{t_i}, \mathbf{x}_{t_j})$ denotes the covariance between corresponding output points

(y_{t_i}, y_{t_j}). A commonly used covariance function is the radial basis function given by

$$k(\mathbf{x}_{t_i}, \mathbf{x}_{t_j}|\boldsymbol{l}, \tau^2) = \tau^2 \exp\left(-\sum_{k=1}^{p} \frac{(\mathrm{x}_{t_{ik}} - \mathrm{x}_{t_{jk}})^2}{2{l_k}^2}\right), \tag{3.3}$$

where $\boldsymbol{l} = (l_1, \ldots, l_p)$ denotes the characteristic length-scale, which models the rapidity of the process. Some other covariance functions are listed in Table 3.1.

Table 3.1: Commonly used Kernel Functions

		Kernel function
Exponential $k_E(\mathbf{x}_{t_i}, \mathbf{x}_{t_j})$		$\tau^2 \exp\left(-\sum_{k=1}^{p} \frac{\lvert \mathrm{x}_{t_{ik}} - \mathrm{x}_{t_{jk}} \rvert}{l_k}\right)$
Matern $k_M(\mathbf{x}_{t_i}, \mathbf{x}_{t_j})$	3/2	$\tau^2 \left(1 + \sum_{k=1}^{p} \frac{\sqrt{3}(\mathrm{x}_{t_{ik}} - \mathrm{x}_{t_{jk}})}{l_k}\right) \exp\left(-\sum_{k=1}^{2p} \frac{\sqrt{3}\lvert \mathrm{x}_{t_{ik}} - \mathrm{x}_{t_{jk}} \rvert}{l_k}\right)$
Rational Quadratic $k_{RQ}(\mathbf{x}_{t_i}, \mathbf{x}_{t_j})$		$\tau^2 \left(1 + \exp\left(-\sum_{k=1}^{p} \frac{(\mathrm{x}_{t_{ik}} - \mathrm{x}_{t_{jk}})^2}{2l_k^2 \alpha}\right)\right)^{-\alpha}$

Let us gather n input and output measurements into the matrix $\mathbf{X} = [\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_n}]^\top$ and the vector $\mathbf{y} = [y_{t_1}, \ldots, y_{t_n}]^\top$, respectively. The matrix of of basis functions is then represented as $\mathbf{H}(\mathbf{X}) = [\boldsymbol{h}(\mathbf{x}_{t_1}), \ldots, \boldsymbol{h}(\mathbf{x}_{t_n})]^\top$. The distribution of $\mathbf{y}$ according to (3.1) is a multivariate normal (also known as marginal likelihood) having a covariance function diagonally additive with noise elements $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$. Formally, we have

$$\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{l}, \tau^2, \sigma^2 \sim \mathcal{N}\left(\mathbf{H}(\mathbf{X})\boldsymbol{\beta}, \boldsymbol{\Sigma}(\mathbf{X})\right), \tag{3.4}$$

where $\boldsymbol{\Sigma}(\mathbf{X}) = \boldsymbol{k}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n$. The noise elements $\boldsymbol{\epsilon}$ with zero mean and variance σ^2, also called "nugget", account for model uncertainty and numerical stability.

The time-series measurements gathered in dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$ are often tainted with out-

liers, which commonly arise due to human errors, measurement errors, meter sensitivity to changing weather conditions, and systematic variation unconsidered in the simplified model, to name a few. The two types of outliers likely to occur are vertical outliers and bad leverage points [54].

- Vertical outlier is an outlier in the response space whose projection on the input vector space associated with the input variables is an inlier.

- A leverage point is a data point whose projection on the input vector space is an outlier. It can be a good or a bad point depending on whether it is an inlier or an outlier in the response space, respectively.

In the estimation process outlined below, we aim to minimize the impact of outliers on the hyperparameters $\boldsymbol{\eta} = \{\boldsymbol{\beta}, \boldsymbol{l}, \tau^2, \sigma^2\}$, with the goal of obtaining a robust approximation of the simulator function f through the estimated mean $\widehat{\mu}$ and covariance function $\widehat{\boldsymbol{\Sigma}}$ of posterior of f in the GP framework.

3.1 Robust hyperparameter estimation

A widely used way of obtaining hyperparameters estimates $\widehat{\boldsymbol{\eta}}$ is by maximizing the marginal likelihood (evidence), called type-II maximum likelihood estimation introduced by []. Formally:

$$\widehat{\boldsymbol{\eta}} = \underset{(\boldsymbol{\beta},\boldsymbol{l},\tau^2,\sigma^2)\in R^q R^p R^{+*} R^{+*}}{arg\,max} \log \mathcal{L}\left(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{l}, \tau^2, \sigma^2\right). \tag{3.5}$$

The hyperparameter associated with mean function $\boldsymbol{\beta}$ is mostly affected by the bad leverage data points (outliers in $\mathbf{X}$). We propose to bound this effect using weights w, which are

based on the robust version of Mahalanobis distances, referred to as projection statistics:

$$\mathrm{PS}_i = \max_{||\boldsymbol{v}||=1} \frac{\boldsymbol{h}_i^T\boldsymbol{v} - \underset{j}{\mathrm{med}}(\boldsymbol{h}_j^T\boldsymbol{v})}{1.4826 \underset{k}{\mathrm{med}} |\boldsymbol{h}_k^T\boldsymbol{v} - \underset{j}{\mathrm{med}}(\boldsymbol{h}_j^T\boldsymbol{v})|}, \tag{3.6}$$

where $\boldsymbol{h}_i = \boldsymbol{h}(\mathbf{x}_i)$; $\boldsymbol{v}_j = \frac{\boldsymbol{u}_j}{||\boldsymbol{u}_j||}$; $\boldsymbol{u}_j = \boldsymbol{h}_i - \mathbf{M}$; $j = 1, \ldots, n$. Here, $\mathbf{M}$ denotes the coordinatewise median given by

$$\mathbf{M} = \{\underset{j=1,\ldots,n}{\mathrm{med}} \boldsymbol{h}_{j1}, \ldots, \underset{j=1,\ldots,n}{\mathrm{med}} \boldsymbol{h}_{jq}\}.$$

Projection statistics are the largest standardized projection distances, which are calculated by projecting the point cloud along lines that start at the coordinate-wise median and pass through each data point [84].

The weight w_i associated with i^{th} input data point $\mathbf{x}_i$ is calculated on $\boldsymbol{h}(\mathbf{x}_i)$ as:

$$w_i = \begin{cases} 1, & \mathrm{PS}_i^2 \leq c; \\ \frac{c}{\mathrm{PS}_i^2}, & \text{otherwise.} \end{cases} \tag{3.7}$$

The data point $\mathbf{x}_i$ is considered as a leverage point if the associated PS_i^2 is greater than threshold c. The weight w_i are then used to scale the residual $r_i = y_i - \boldsymbol{h}(\mathbf{x}_i)^\top\boldsymbol{\beta}$. associated with i^{th} data point.

A weighted loss function of standardized residuals $r_{Si} = \frac{r_i}{w_i\sigma^2 s}$ is formed based on the Schweppe-type M-estimator [], given by,

$$J(\boldsymbol{\beta}) = \sum_{i=1}^{n} w_i^2 \rho\left(r_{Si}\right). \tag{3.8}$$

The residual scale s is robustly estimated by $\widehat{s} = 1.4826\left(1 + \frac{5}{n-q}\right)\mathrm{med}|\boldsymbol{r}|$ when there is a little to none knowledge about σ^2. We employ the Huber loss function $\rho(\cdot)$ because of its unimodal and its quadratic characteristic at its center (to gain efficiency at the Gaussian

distribution). It is defined as

$$\rho(r_i) = \begin{cases} \frac{r_i^2}{2} & \text{for } r_i < b, \\ b|r_i| - \frac{b^2}{2} & \text{for } r_i \geq b. \end{cases} \tag{3.9}$$

The threshold parameter b is typically chosen to be equal to 1.5, which offers a good compromise between a high statistical efficiency at the Gaussian distribution and good robustness against outliers.

The marginal likelihood is now modified to incorporate $J(\boldsymbol{\beta})$ as:

$$\log\mathcal{L} = -J(\boldsymbol{\beta}) - \log|\boldsymbol{\Sigma}\left(\mathbf{X}|\boldsymbol{l}, \tau^2, \sigma^2\right)| - \frac{n}{2}\log(2\pi) \tag{3.10}$$

We are now ready to propose a hyperparameter estimation algorithm. First, we will iteratively estimate $\widehat{\boldsymbol{\beta}}$, and then estimate τ and $\boldsymbol{l}$ using the robustly obtained $\boldsymbol{\beta}$-profile likelihood.

Setting gradient of the objective function $\log\left(\mathcal{L}(\mathbf{y}|\mathbf{X}, \boldsymbol{\eta})\right)$ with respect to $\boldsymbol{\beta}$ to zero gives

$$\sum_{i=1}^{n} w_i \boldsymbol{h}_i \frac{\partial\rho(r_{Si})}{\partial r_{Si}} = 0. \tag{3.11}$$

Let us define the psi-function as $\psi(r_{Si}) = \frac{\partial\rho(r_{Si})}{\partial r_{Si}}$. (3.11) now becomes

$$\sum_{i=1}^{n} w_i \boldsymbol{h}_i \psi(r_{Si}). \tag{3.12}$$

Dividing (3.12) by the standardized residuals r_{S_i}, we get

$$\sum_{i=1}^{m} q\left(\frac{r_i}{w_i\sigma^2 s}\right)\boldsymbol{h}_i r_i = \mathbf{0}, \tag{3.13}$$

where $q(r_{Si}) = \frac{\psi(r_{Si})}{r_{Si}}$ defines weight function, given by:

$$\boldsymbol{q}(r_{S_i}) = \begin{cases} 1, & r_i \leq c \\ \frac{b\,\text{sign}(r_{S_i})}{r_{Si}}, & \text{otherwise.} \end{cases} \tag{3.14}$$

Substituting the expression of the residuals, r_i, and rewriting (3.13) in matrix form yields

$$\mathbf{H}^T\mathbf{Q}(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}) = \mathbf{0} \tag{3.15}$$

$$\boldsymbol{\beta} = (\mathbf{H}^T\mathbf{Q}\boldsymbol{\Sigma}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Q}^{(k)}\boldsymbol{\Sigma}^{-1}\mathbf{y}, \tag{3.16}$$

where $\mathbf{Q} = \text{diag}(q(r_{S_i}))$. Since $\mathbf{Q}$ is function of residuals, we solve for $\boldsymbol{\beta}$ in an iterative manner by incorporating iterative re-weighted least squares (IRLS) algorithm. Formally, we have:

$$\boldsymbol{\beta}^{(k+1)} = (\mathbf{H}^T\mathbf{Q}^{(k)}\boldsymbol{\Sigma}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Q}^{(k)}\boldsymbol{\Sigma}^{-1}\mathbf{y}. \tag{3.17}$$

Let us denote $\widehat{\boldsymbol{\beta}}$ as the value converged by IRLS. Now, $(\boldsymbol{l}, \tau^2, \sigma^2)$ is estimated by substituting $\widehat{\boldsymbol{\beta}}$ in (3.10) using in maximum likelihood estimation:

$$(\widehat{\boldsymbol{l}}, \widehat{\tau}^2, \widehat{\sigma}_n^2) = \underset{\boldsymbol{l},\tau^2,\sigma_n^2}{arg\,max} \log \mathcal{L}\left(\mathbf{y}|\mathbf{X}, \widehat{\boldsymbol{\beta}}, \boldsymbol{l}, \tau^2, \sigma^2\right), \tag{3.18}$$

which reduces to

$$(\widehat{\boldsymbol{l}}, \widehat{\tau}^2, \widehat{\sigma}^2) = \underset{\boldsymbol{l},\tau^2,\sigma_n^2}{arg\,min}\, \Gamma(\boldsymbol{l}, \tau^2, \sigma_n^2). \tag{3.19}$$

Here

$$\Gamma(\boldsymbol{l}, \tau^2, \sigma_n^2) = \log |\boldsymbol{k}(\mathbf{X}, \mathbf{X}|\boldsymbol{l}, \tau^2) + \sigma^2\mathbf{I}_n|. \tag{3.20}$$

The hyperparameters, $(\widehat{\boldsymbol{l}}, \widehat{\tau}^2, \widehat{\sigma}^2)$, are estimated by utilizing a gradient-based optimizer as

described in [96]. We can then update $\widehat{\boldsymbol{\beta}}$ as $\widehat{\widehat{\boldsymbol{\beta}}} = \widehat{\boldsymbol{\beta}}(\widehat{\boldsymbol{l}}, \widehat{\tau}^2, \widehat{\sigma}^2)$. The algorithm for estimating the hyperparameters is outlined in Algorithm 1.

Algorithm 1 Algorithm for Estimating the Hyperparameters

1: Constitute observational training dataset data-set $\mathcal{D} = (\mathbf{X}, \mathbf{y})$;
2: Construct $\mathbf{H}$ using a suitable basis function;
3: Calculate the projection statistics of the row vectors of $\mathbf{H}$ given by (3.6);
4: Calculate the weights $\boldsymbol{w}$ based on the PS given by (4.4);
5: Initialize $\boldsymbol{\beta}$ using the weighted least squares solution as $\boldsymbol{\beta}_0 = (\mathbf{H}^T\boldsymbol{\Sigma}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\boldsymbol{\Sigma}^{-1}\mathbf{y}$;
6: Update $\boldsymbol{\beta}$ by executing the IRLS algorithm given by (3.17) until convergence while setting the hyperparameters $(\boldsymbol{l}, \tau^2, \sigma_n^2)$ at their initial values to obtain $\widehat{\boldsymbol{\beta}}$;
7: Update $(\boldsymbol{l}, \tau^2, \sigma^2)$ while setting $\boldsymbol{\beta} = \widehat{\boldsymbol{\beta}}$;
8: Iterate Steps 7 and 8 until convergence, e.g. $||\boldsymbol{r}|| \leq 0.001$, to obtain the final hyperparameter estimates, $(\widehat{\widehat{\boldsymbol{\beta}}}, \widehat{\boldsymbol{l}}, \widehat{\tau}^2, \widehat{\sigma}^2)$.

3.2 Model and prediction uncertainty

For the surrogate based uncertainty analysis, let us consider that we draw N samples of the input test variable $\mathbf{x}_{t^*}$ at instances $\boldsymbol{t}^* = [1, \ldots, n^*]$ in the prediction interval.

In time-series analysis, the assumption of stationarity and ergodicity for a specific range of time is often made, very oftenly for steady state estimations. The latter means that the sample average, commonly known as the ensemble average, is equal to the time average. The assumption of ergodicity allows us to model a stochastic time-series process using a single real-time measurement per instance. Let us group the N sampled test predictors for instance t_i^* denoted as $\mathbf{x}_{t_i^*}^{(k)}$ for k in $[1,\ N]$ into $\mathbf{X}_i^* = [\mathbf{x}_{t_i^*}^{(1)}, \ldots, \mathbf{x}_{t_i^*}^{(N)}]^T$. Using a hierarchical formulation, the model output variables $\mathbf{y}^*$ obtained through the power flow simulator $f(\cdot)$ at the test points $\mathbf{X}_i^*$ together with the training output variables follow a joint multivariate

Gaussian distribution given by

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^*|\mathbf{X}_i^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m}(\mathbf{X}) \\ \boldsymbol{m}(\mathbf{X}_i^*) \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}(\mathbf{X}) & \mathbf{C}(\mathbf{X}_i^*) \\ \mathbf{C}^T(\mathbf{X}_i^*) & \mathbf{V}(\mathbf{X}_i^*) \end{bmatrix} \right), \tag{3.21}$$

where $\mathbf{C}(\mathbf{X}_i^*) = \boldsymbol{k}(\mathbf{X}, \mathbf{X}_i^*), \mathbf{C}^T(\mathbf{X}_i^*) = \boldsymbol{k}(\mathbf{X}_i^*, \mathbf{X})$ and $\mathbf{V}(\mathbf{X}_i^*) = \boldsymbol{k}(\mathbf{X}_i^*, \mathbf{X}_i^*)$. The covariance matrix, $\boldsymbol{\Sigma}(\mathbf{X})$, is represented by $\boldsymbol{\Sigma}$ hereafter. Furthermore, we assume an a priori Gaussian probability distribution for the simulator output at the test points, $f(\mathbf{X}_i^*)|\mathbf{X}_i^*$, that is,

$$f(\mathbf{X}_i^*)|\mathbf{X}_i^* \sim \mathrm{GP}\left(\boldsymbol{m}(\mathbf{X}^*), \mathbf{V}(\mathbf{X}_i^*)\right). \tag{3.22}$$

Upon conditioning and using the standard techniques in multivariate distributions, we get

$$f(\mathbf{X}_i^*)|\mathbf{X}_i^*, \mathbf{y}, \mathbf{X}, \boldsymbol{\beta}, \boldsymbol{l}, \tau^2, \sigma^2 \sim \mathrm{GP}\left(\boldsymbol{\mu}^*(\mathbf{X}), \boldsymbol{\Sigma}^*(\mathbf{X})\right), \tag{3.23}$$

where the estimated mean function $\widehat{\boldsymbol{\mu}}^*(\mathbf{X}_i^*)$ is given by

$$\widehat{\boldsymbol{\mu}}^*(\mathbf{X}_i^*) = \widehat{\boldsymbol{m}}(\mathbf{X}_i^*) + \widehat{\mathbf{C}}^T(\mathbf{X}_i^*)\widehat{\boldsymbol{\Sigma}}^{-1}\mathbf{r}, \tag{3.24}$$

and the estimated covariance function $\widehat{\boldsymbol{\Sigma}}^*(\mathbf{X}^*)$ is expressed as

$$\widehat{\boldsymbol{\Sigma}}^*(\mathbf{X}_i^*) = \widehat{\mathbf{V}}(\mathbf{X}_i^*) - \widehat{\mathbf{C}}^T(\mathbf{X}_i^*)\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\mathbf{C}}(\mathbf{X}_i^*), \tag{3.25}$$

for all the instances in the predictive interval, $i = 1, \ldots, n^*$. The estimate of the mean function given by (3.24) acts as a computationally efficient surrogate model that captures the behavior of the power flow while the covariance matrix estimate given by (3.25) quantifies the associated uncertainty.

Let us apply a weak prior for $(\boldsymbol{\beta}, \tau^2)$, $p(\boldsymbol{\beta}, \tau^2) \propto \frac{1}{\tau^2}$, combining with (3.4) and using Bayes' theorem yields a posterior distribution for $(\boldsymbol{\beta}, \tau^2)$, which is normal inverse-gamma distribution given by

$$\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}, \boldsymbol{l}, \tau^2, \sigma^2 \sim \mathcal{N}(\hat{\boldsymbol{\beta}}, \tau^2(\mathbf{H}^T\boldsymbol{\Sigma}^{-1}\mathbf{H})^{-1}), \tag{3.26}$$

where $\hat{\boldsymbol{\beta}}$ is the weighted least squares estimate given by $\hat{\boldsymbol{\beta}} = (\mathbf{H}^T\boldsymbol{\Sigma}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\boldsymbol{\Sigma}^{-1}\mathbf{y}$, and

$$\tau^2|\mathbf{y}, \mathbf{X}, \boldsymbol{l}, \sigma_n^2 \sim \text{InvGamma}\left(\frac{n-q}{2}, \frac{(n-q-2)\hat{\tau}^2}{2}\right), \tag{3.27}$$

where $\hat{\tau}^2 = \frac{\mathbf{y}^T(\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\mathbf{H}(\mathbf{H}^T\boldsymbol{\Sigma}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\boldsymbol{\Sigma}^{-1})\mathbf{y}}{(n-q-2)}$.

3.3 Theoretical robustness of the proposed model

The influence function of the Schweppe type M-estimator is given by

$$\text{IF}(r_{Si}, \boldsymbol{h}_i; \Phi) = \frac{\psi(r_{Si})}{E_\Phi[\psi'(r_{Si})]}(\mathbf{H}^T\mathbf{H})^{-1}\boldsymbol{h}_i w_i. \tag{3.28}$$

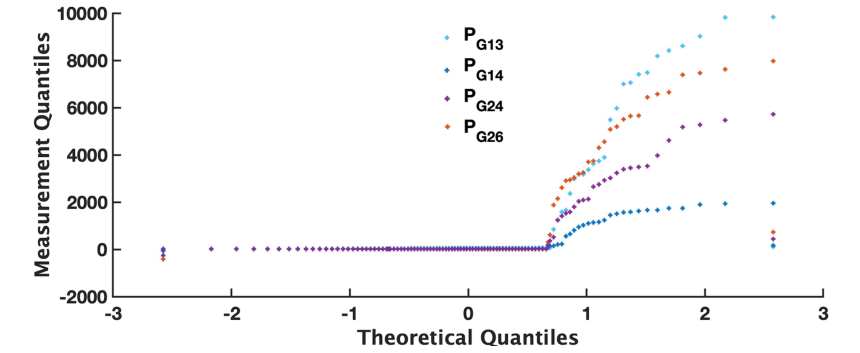
One can notice that the influence of position, $\text{IP}(\boldsymbol{h}_i, \Phi) = (\mathbf{H}^T\mathbf{H})^{-1}\boldsymbol{h}_i w_i$, is bounded thanks to the weights calculated using the projection statistics, whose breakdown point attains the maximum given by $\frac{[(n-q-1)/2]}{n}$ [79]. Note that the estimator reduces to an ℓ_2-norm estimator for small standardized residuals and to the ℓ_1-norm estimator for larger ones. Therefore, it has a high statistical efficiency at the Gaussian distribution while being robust to outliers.

3.4 Simulation results

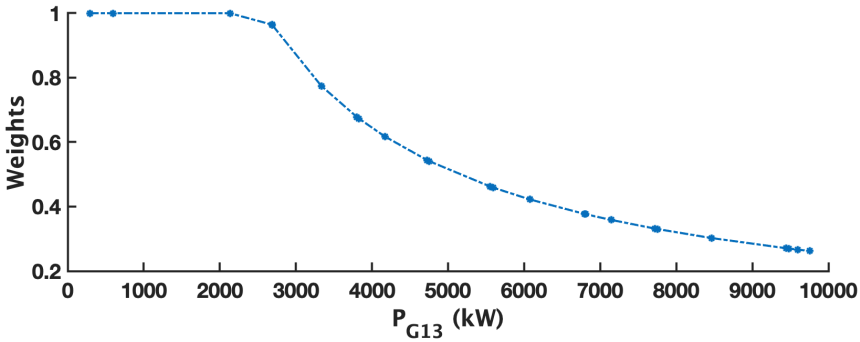
In this section, we compare the performance of the proposed model RPM to that of the Gaussian process model (GPM) when applied to a standard IEEE 33-bus system (Case A) and to a real-world 240-bus distribution system located in the Midwest U.S. with high penetration of RESs and DGs (Case B). We add vertical outliers, i.e., outliers in $\mathbf{y}$, bad leverage points, i.e., outliers in $\mathbf{X}$, and good leverage points, i.e., outliers in both $(\mathbf{X}, \mathbf{y})$ up to 25% in the training data. To demonstrate the good performance of the RPM for non-Gaussian distribution noises, we assume that the noise follows the Student's t distribution with 10 degrees of freedom. This distribution is chosen because it has heavier tails for low degrees of freedom, producing sampling values that may fall far from its median. We compare the performances of the GPM and the RPM using mean absolute error and the root mean square index for each of the cases.

3.4.1 IEEE 33-bus system

The RPM is applied to a standard IEEE 33-bus system, to which are attached four RES, namely, a PV (P_{G24}) to Bus 24, and three WGs ($P_{G13}, P_{G14}, P_{G26}$) to Buses 13, 14, and 26 of capacity 1 kW, 50 kW, 10kW, 10kW, respectively. The time-series data considered for the RES power outputs and loads are the real measurements with a resolution of $1s$. We run the power flow simulator at $n = 150$ input data points $\mathbf{X} = [\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_{150}}]^T$ to obtain the corresponding voltage magnitude and angle values $\mathbf{y} = [y_{t_1}, \ldots, y_{t_{150}}]^T$ that constitutes the training data. Trained on $(\mathbf{y}, \mathbf{X})$, the RPM and the conventional GPM are used to make predictions for the next $n^* = 60$ data points constituted as the validation data set at instances $\boldsymbol{t}^* = [t_{151}, \ldots, t_{210}]$. To perform stochastic analysis, Latin hypercube sampling is employed to generate 7000 samples of the input variables at each instance in the validation data set



(a)



(b)

Figure 3.1: Outliers corrupting the training data set; (a) QQ-plot of the measurements corrupted with 25% of outliers; (b) plot of the weights using the PSs vs. the outlier magnitudes.

following the Weibull distribution for WGs ($P_{t_i^*,G13}, P_{t_i^*,G14}, P_{t_i^*,G24} \sim \text{Weibull}(2.06, 7.1)$) and the Beta distribution for the PV ($P_{t_i^*,G26} \sim \text{Beta}(2.06, 2.5)$); $i = 151, \ldots, 210$. The results obtained from the Monte Carlo (MC) simulations performed at these samples stand as reference values for comparing the results obtained from the RPM and the GPM. The robustness of the RPM is demonstrated by the addition of 25% outliers as shown in Fig. 3.1 (a) in the training data set. To be precise, we impose a worst-case scenario by adding bad leverage points to the input data points $[\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_{37}}]$, i.e., to the measurements $(\mathbf{P}_{G13}, \mathbf{P}_{G14}, \mathbf{P}_{G24}, \mathbf{P}_{G26})$ and to the load consumption of the load buses, $\{\mathbf{P}_{L1}, \mathbf{P}_{L2}, \ldots, \mathbf{P}_{L33}\}$. Similarly, vertical outliers are added to the output data points $[y_{t_1}, \ldots, y_{t_{37}}]$, i.e., to the measurements of voltage phasors.

We observe from the weights displayed in Fig. 3.1 (b) that the SHGM estimator downweights

the bad leverage points and vertical outliers. The prediction results of the voltage magnitude and angle for Bus 19 with the percentage of outliers up to 25 in the training data constitute a benchmark for this study. The mean and standard deviation values (indicated as error bars) of the prediction results for the voltage angle of Bus 19 are displayed in Fig. 3.2 (a), where the error bars represent standard deviation values. Fig. 3.2 (c) depicts the data fit for the training duration $[t_1 - t_{150}]$ obtained for the RPM. Fig. 3.2 (b) compares the probability density of the voltage angle of Bus 19 calculated from the 7000 realizations at the next instance t^*_{151} obtained from the RPM to the MC simulation output. Fig. 3.3 (a) displays the predicted values of the voltage magnitude at Bus 19 obtained from the RPM and from the GPM. We observe that the predicted values from the GPM deviate largely from the true values. This is due to the fact that the estimate of the mean function hyperparameter of the conventional GPM is centered at the basic weighted least squares estimate. Therefore, it fails to represent the simulator in presence of outliers while the RPM succeeds. Also, the prediction accuracy is displayed in Fig. 3.3 (b) using root mean square error (RMSE) values when the training data set is added with an increasing percentage of outliers up to 25%. We notice that the RPM consistently exhibits low RMSE values. The RMSE and mean absolute error (MAE) values for the forecast of the voltage phasors at Bus 19 are listed in Table 3.2 for the cases of training data added with and without the outliers for both the linear and quadratic basis function. We observe that the prediction results are more accurate for the quadratic basis function in the case of added outliers. Therefore, by the principle of parsimony, we choose a quadratic basis to obtain the results for the voltage phasors for all the 33-buses in the network plotted in Fig. 3.4.

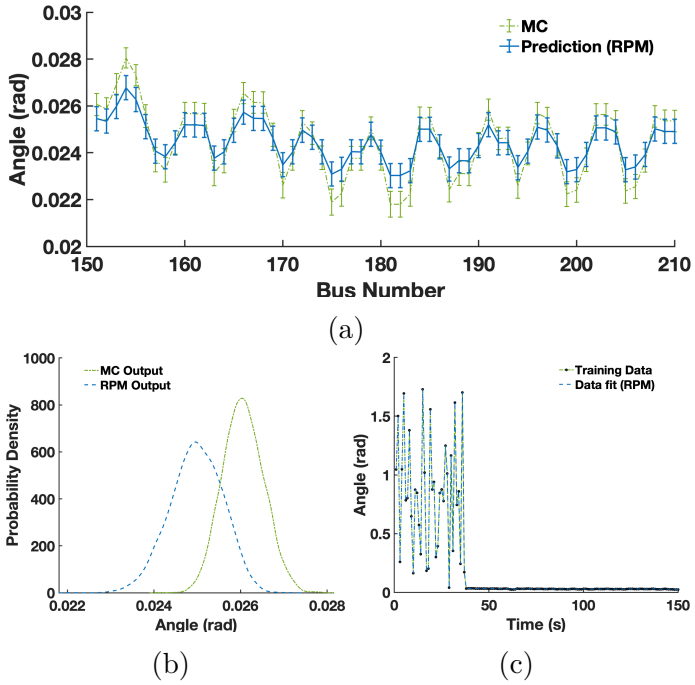


Figure 3.2: RPM results for the voltage phase angle at Bus 19: (a) prediction at the test points; (b) probability density at the test points; and (c) fitted values over the training data points.

Table 3.2: The RMSE and MAE for the Bus 19 of the IEEE 33-bus system

	Quadratic Basis								Linear Basis							
	With 25% outliers				Without outliers				With 25% outliers				Without outliers			
	RPM		GPM		RPM		GPM		RPM		GPM		RPM		GPM	
Measure	V	A	V	A	V	A	V	A	V	A	V	A	V	A	V	A
RMSE	0.0034	$9.7810e^{-4}$	1.0274	0.1527	0.0931	0.0058	0.0657	0.1468	0.0264	0.01264	0.7995	0.0206	0.01005	0.01747	0.00838	0.01645
MAE	0.003	$8.2815e^{-4}$	7.3087	0.1287	0.0672	$2.5516e^{-4}$	0.0334	0.1669	0.0047	$9.7962e^{-4}$	4.2355	0.0029	0.00058	0.00176	0.00042	0.0018

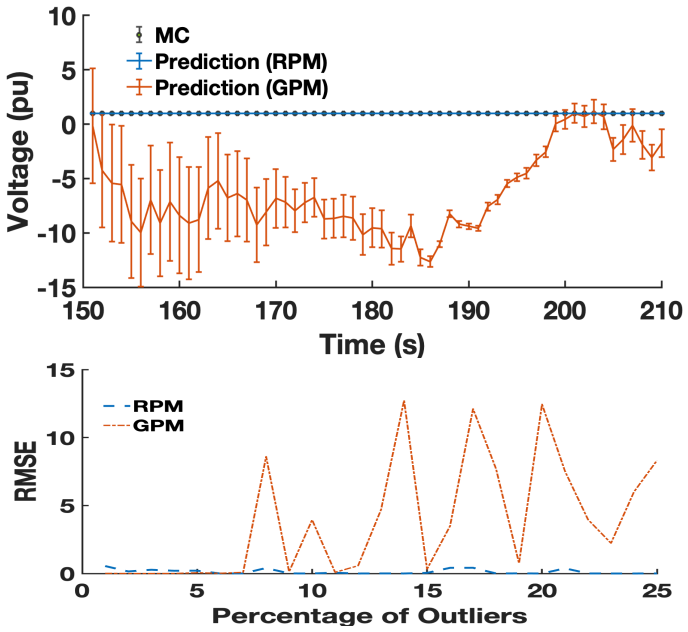
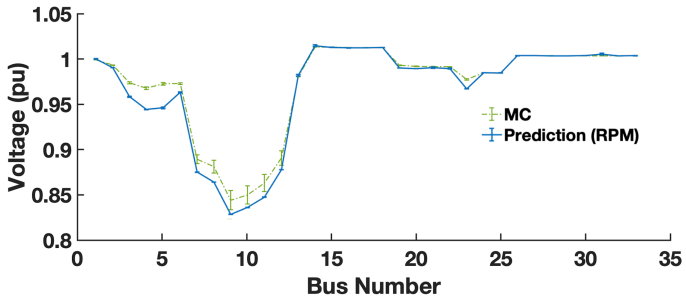


Figure 3.3: Comparison between the performance of the RPM and the GPM: (a) voltage magnitude at Bus 19; (b) RMSE values.

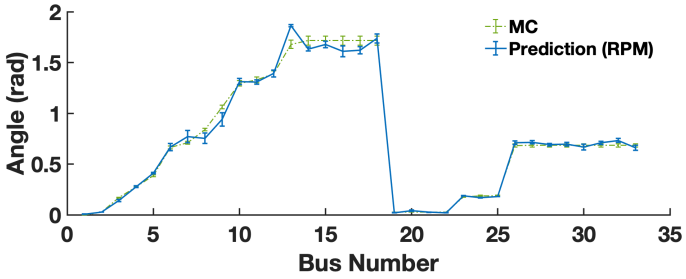
3.4.2 Real-world 240-bus system

We integrate the 240-bus radial distribution system [19] with RES, namely 35 PVs and 35 WGs distributed across the network. Their locations in the network are displayed in Fig. 3.5. Please note that the RES are connected to each phase of the indicated buses in the network.

The training data set $(\mathbf{X}, \mathbf{y})$ is obtained by running the three-phase power flow simulator for the hourly spaced load and active and reactive power injection measurements for 7 days i.e. a total of $n = 168$ data points constitute a training data set for the design of the GPM and the RPM. The GPM and RPM are analyzed using the prediction results obtained for the $n^* = 24$ test data points, which are the validation data points. For the probabilistic analysis, 7000 samples are drawn using Latin hypercube sampling from input variables of load following Gaussian distribution $P_{t_i^*,L} \sim \mathcal{N}(P_{t_i^*,L}, 0.05P_{t_i^*,L})$, $i = 169, \ldots, 192$, the WGs'



(a)



(b)

Figure 3.4: RPM predictions for the IEEE 33-bus system with the training data set added with 25% of outliers: (a) voltage magnitudes; (b) voltage phase angles.

output following the Weibull distribution, and the PVs' output following the Beta distribution with the shape and scale parameters same as mentioned in Section 4.A. Voltage phasors predictions at Bus 2003.2 for a day ahead forecast constitute as a benchmark for this study. We display the mean of the prediction results obtained from the GPM and the RPM of the voltage magnitude at Bus 2003.2 using linear and quadratic basis functions in Figs. 3.6 (b) and (d), respectively, where the error bars represent the standard deviations as showing their probability distributions individually will be inconvenient. Similarly, the prediction results of the voltage angle at Bus 2003.2 are plotted in Figs. 3.7 (b) and (d).

We now add up to 25% of outliers in the power injection measurements and voltage magnitudes and phase angles, i.e., to the first 42 input and output data points in the training dataset. To best demonstrate the robustness of the RPM, we choose the outlier distribution to be Student's t with 10 degrees of freedom because of heavier tails as displayed in Fig.

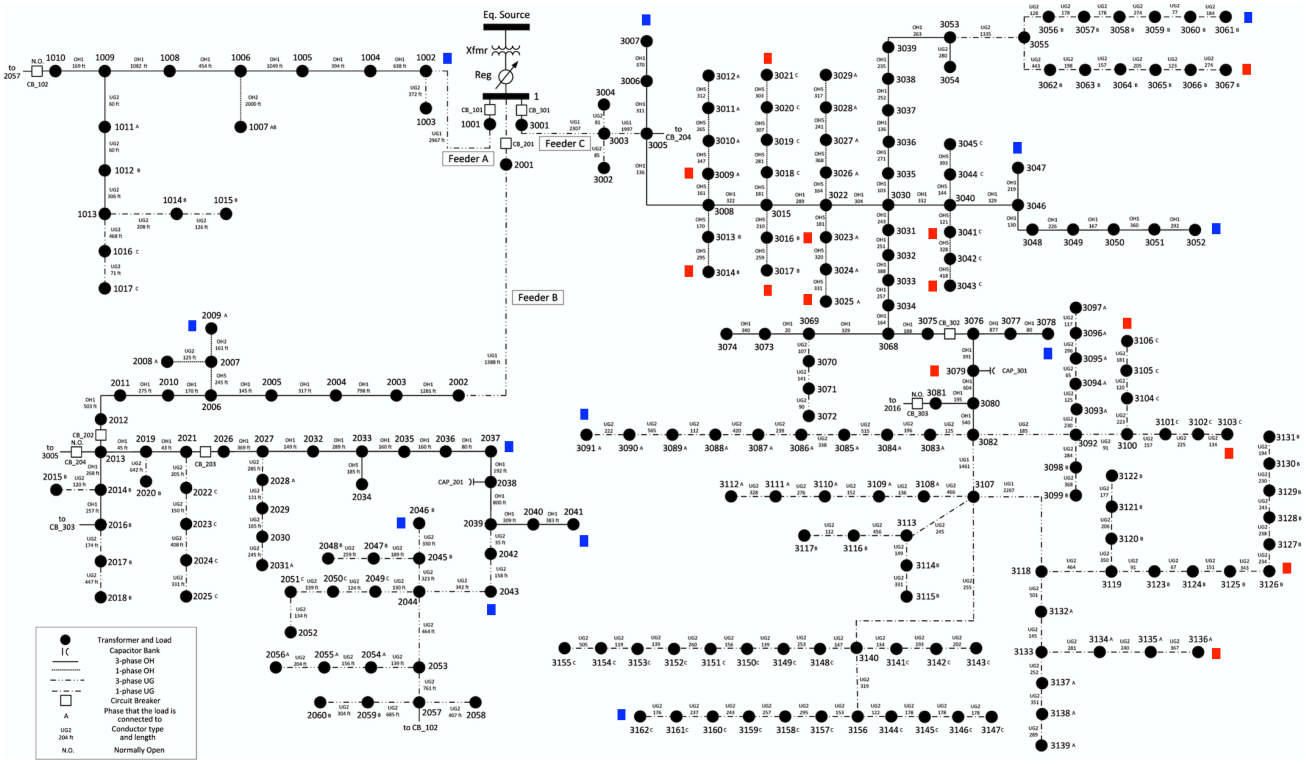
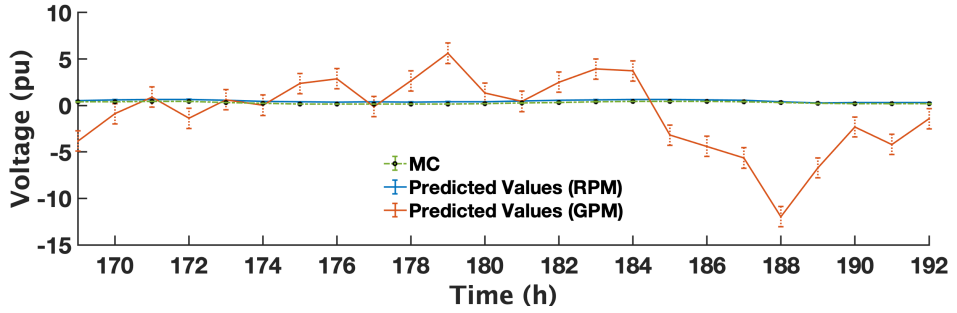
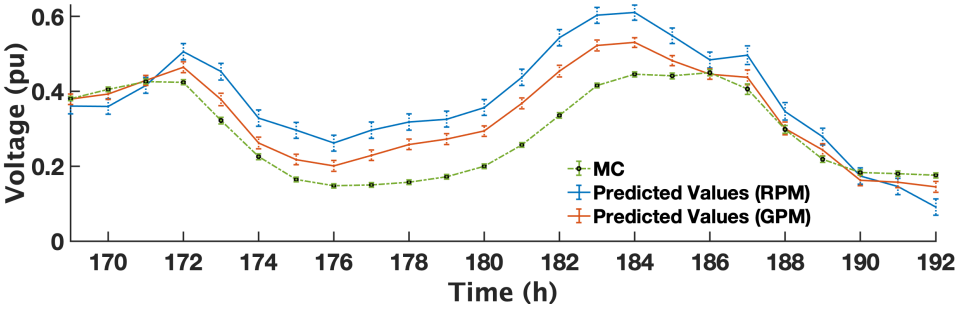


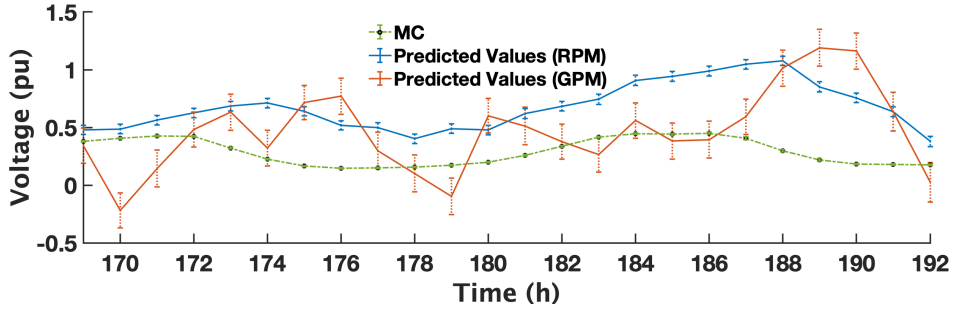
Figure 3.5: The online diagram of the 240 bus system integrated with RES. Blue and red squares indicate the PVs and WGs, respectively



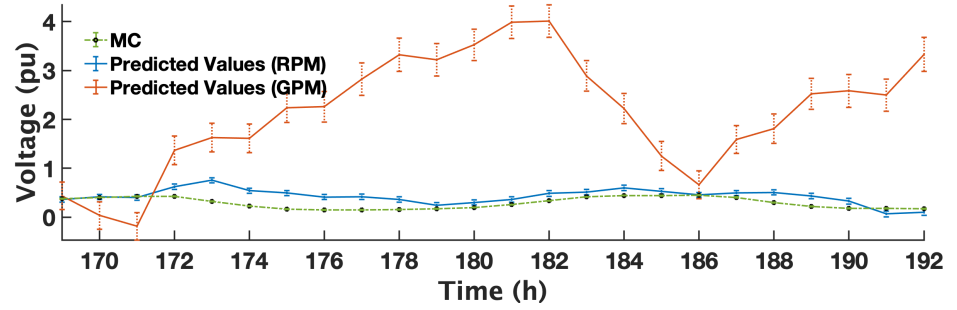
(a)



(b)

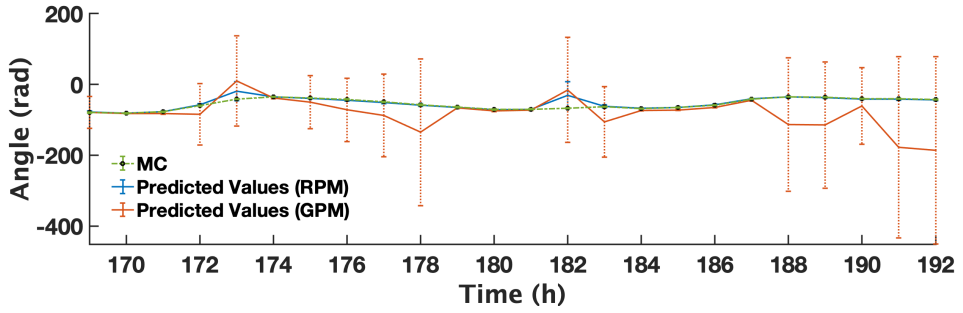


(c)

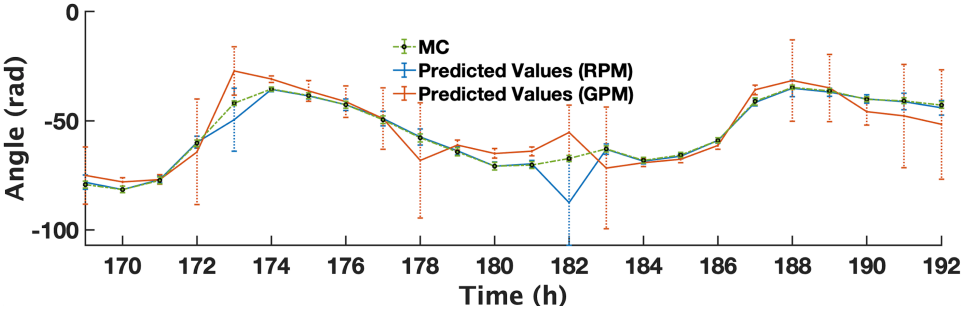


(d)

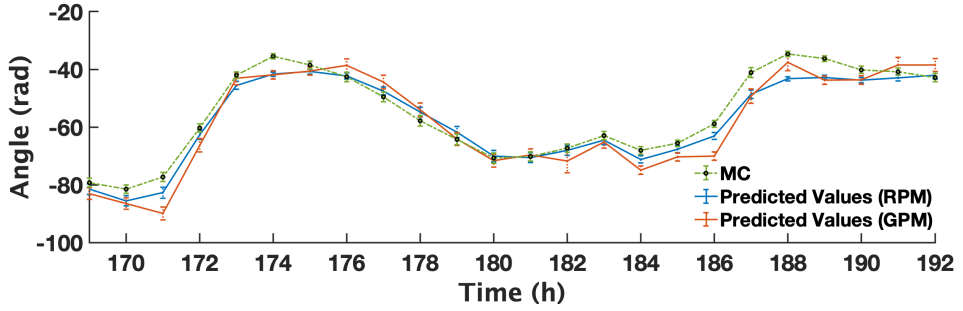
Figure 3.6: Comparison between the GPM and the RPM forecast results for the voltage magnitude of Bus 2003.2 in the 240−bus network when (a) the training data set is added with 25% of outliers ; (b) training data set is not added with outliers for linear basis; (c) the training data set is added with 25% of outliers ; (d) training data set is not added with outliers for quadratic basis.



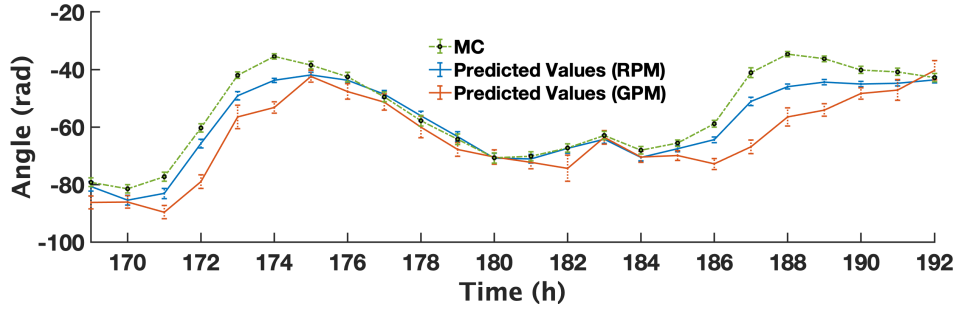
(a)



(b)

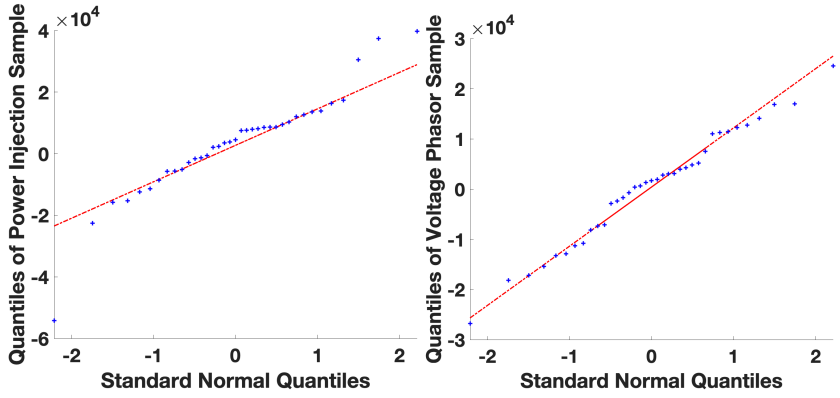


(c)



(d)

Figure 3.7: Comparison between the GPM and the RPM forecast results for the voltage angle of Bus 2003.2 in the 240−bus network when (a) the training data set is added with 25% of outliers ; (b) training data set is not added with outliers for linear basis; (c) the training data set is added with 25% of outliers ; (d) training data set is not added with outliers for quadratic basis.



(a) (b)

Figure 3.8: QQ plot of 25% outliers added in (a) power injection measurements; (b) voltage magnitude measurements.

3.8. Similarly, the same percentage of outliers is included in the measurements of active and reactive load power. We display in Figs. 3.9 (b) and (d) the probability density function of the voltage magnitude at Bus 2003.2 for the test input at instance t^*_{169} obtained using linear and quadratic basis function. Similarly, the probability density results of the voltage angle at Bus 2003.2 are plotted in Figs. 3.10 (a) and (b). We compare the prediction results of the voltage magnitude at Bus 2003.2 obtained from the GPM and the RPM with basis function as linear and quadratic in Figs. 3.6 (a) and (c), respectively. The voltage angle results are displayed in Figs. 3.7 (a) and (c). The RMSE and MAE values of the prediction results for the cases of addition of outliers in the training data set and without the addition of outliers, both with the linear and basis function, are listed in Table 3.3. We observe that the RMSE and the MAE values obtained from the GPM and the RPM for the voltage magnitudes' predictions are lesser with linear basis functions than the ones with the quadratic basis functions for both cases. As for the voltage angles, both models' performance is better with the quadratic basis function for the case with outliers in the training data set. The linear basis function yields better performance for the case without outliers. Therefore, we choose the linear basis to plot the voltage magnitudes and the quadratic basis function for

Table 3.3: The RMSE and MAE for the Bus 2003.2 of the 240-bus system

	Quadratic Basis								Linear Basis							
	With 25% outliers				Without outliers				With 25% outliers				Without outliers			
	RPM		GPM		RPM		GPM		RPM		GPM		RPM		GPM	
	V	A	V	A	V	A	V	A	V	A	V	A	V	A	V	A
RMSE	0.4222	3.8734	0.4242	5.5007	0.1888	4.9835	2.2246	11.1942	0.1921	8.8619	4.0814	53.4306	0.1183	4.4287	0.0637	6.1481
MAE	0.3845	3.1791	0.3164	4.5781	0.1544	3.8252	1.9426	8.5223	0.1859	3.3976	3.0779	34.3391	0.1026	1.5122	0.0513	4.9164

the voltage angles to obtain further predictions at all the buses in the system for the case with outliers in the training data set. The mean values (indicated as dots) and standard deviations (indicated as error bars) of both the models' prediction results for the voltage magnitudes (see, Fig. 3.12) and phase angles (see, Fig. 3.13) at the buses of the 240−bus system are compared with the MC simulation results. We observe that the results obtained from the comparable GPM deviate significantly from the true values in both the mean and standard deviation. The performance of the RPM is comparably accurate on account of the trade-off between the accuracy and robustness of the SHGM estimator. Conventional GPM is strongly biased towards outliers, thus the prediction results deviate further away from the MC results. The bias is particularly significant in phase C results because of the high variance of voltage phasors due to the large power flow in lines. For a large magnitude of outliers, the resulting bias is the worst-case scenario that can be imposed on power system measurements. The proposed RPM keeps this bias finite as long as the added outliers are added without exceeding the breakdown point, whereas the bias is unbounded for the results of conventional GPM.

The RMSE of the predicted values for the voltage magnitude and the angle at Bus 2003.2 with an increasing percentage of outliers added in the training data are plotted in Fig.3.11 (a) and (b), respectively. We observe that the RMSE values obtained from the GPM are higher than the ones obtained from the RPM which provides consistently low RMSE results.

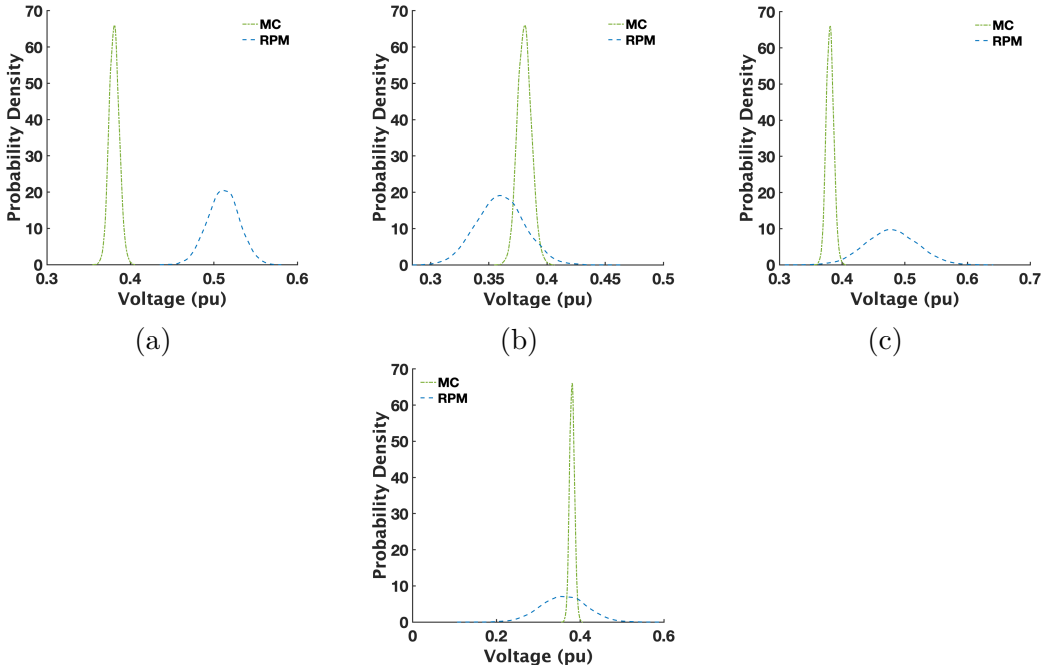


Figure 3.9: Comparison between the GPM and the RPM probability density results for the voltage magnitude of Bus 2003.2 in the 240−bus network when (a) the training data set is added with 25% of outliers ; (b) training data set is not added with outliers for linear basis; (c) the training data set is added with 25% of outliers ; (d) training data set is not added with outliers for quadratic basis.

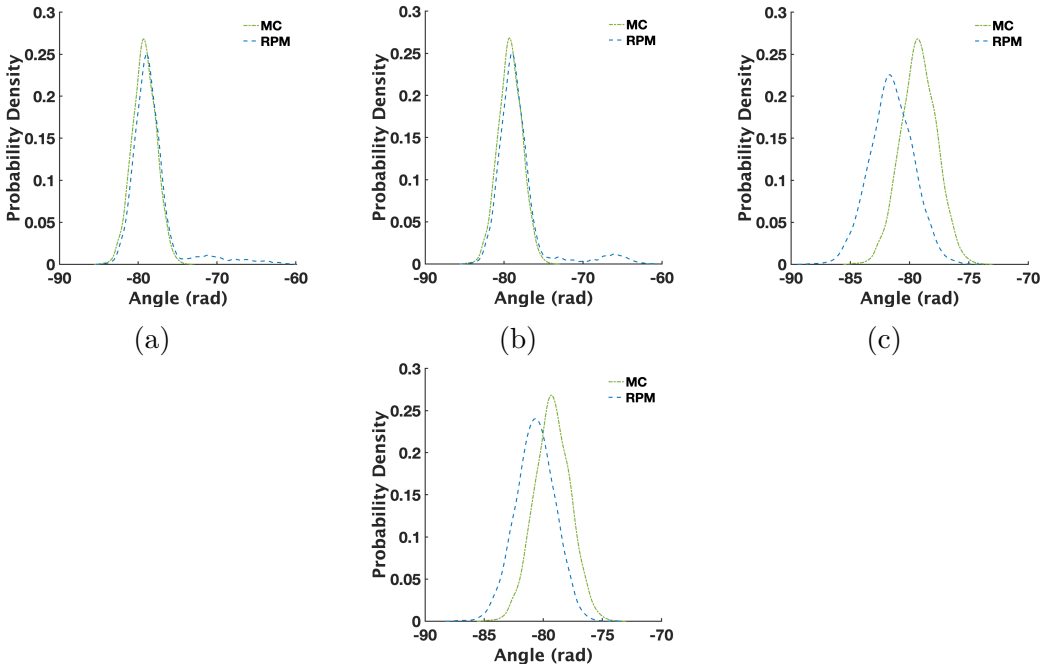


Figure 3.10: Comparison between the GPM and the RPM probability density results for the voltage angle of Bus 2003.2 in the 240−bus network when (a) the training data set is added with 25% of outliers ; (b) training data set is not added with outliers for linear basis; (c) the training data set is added with 25% of outliers ; (d) training data set is not added with outliers for quadratic basis.

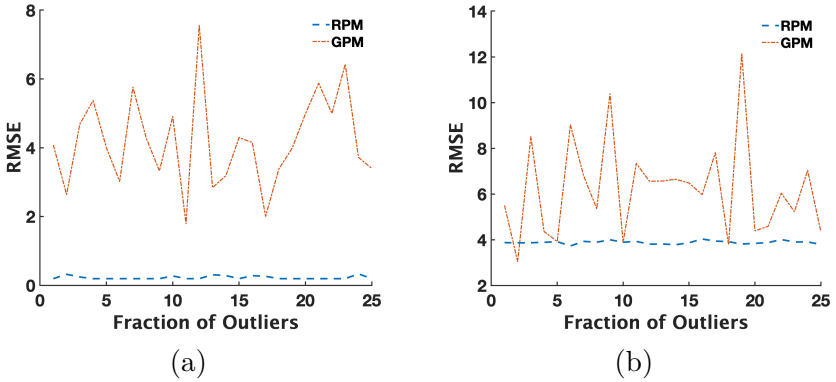


Figure 3.11: RMSE vs the percentage of outliers added in training data for the prediction results at Bus 2003.2 (a) voltage magnitude; (b) voltage angle.

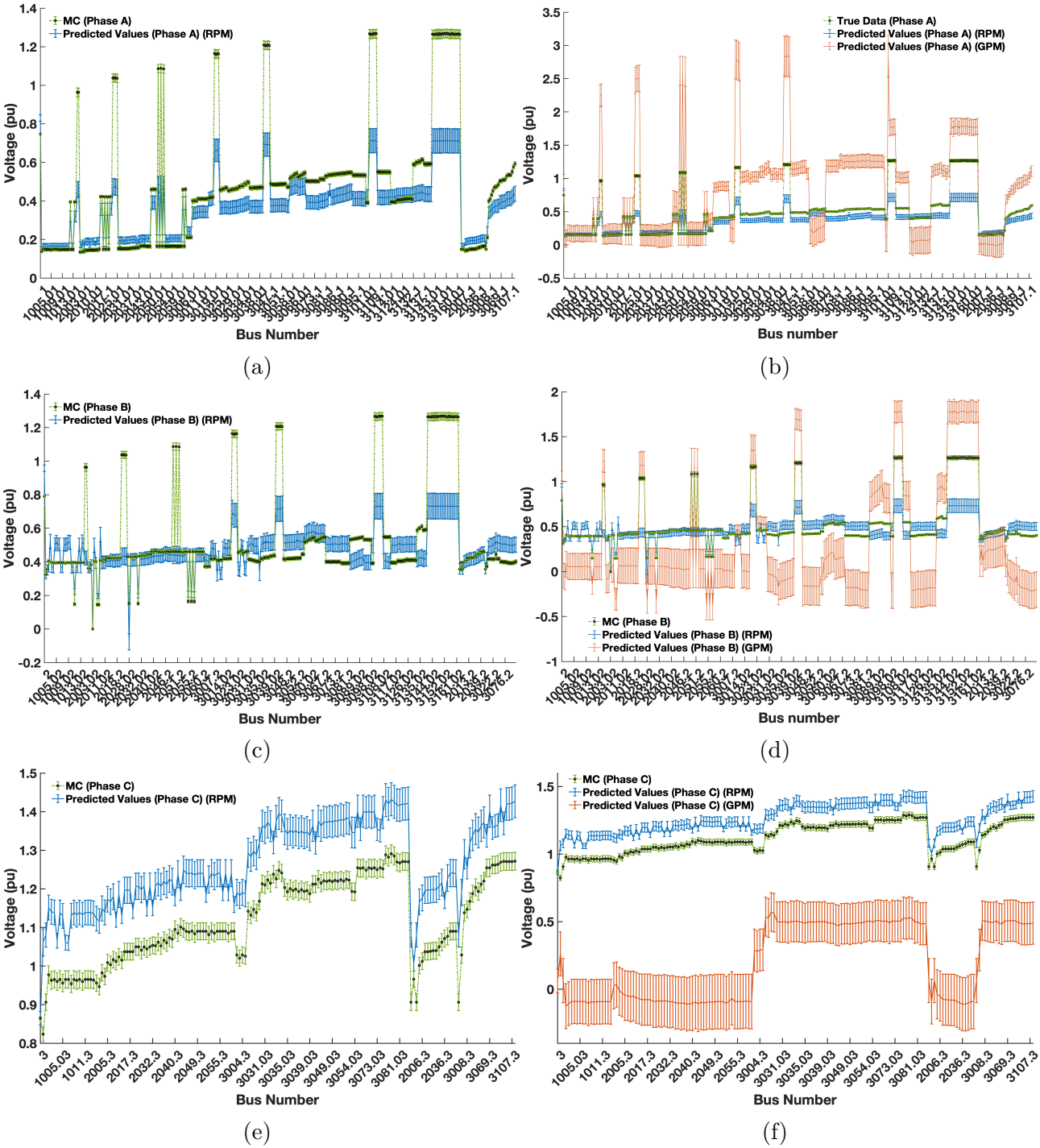


Figure 3.12: The prediction results of voltage magnitude from the RPM compared with those obtained from the GPM of 240−bus system with 25% of outliers added in training data. (a) The results obtained from the RPM for phase a; (b) comparison between the GPM and the RPM for phase a; (c) The results from the RPM for phase b; (d) comparison between the GPM and the RPM for phase b; (e) The results from the RPM for phase c; (f) comparison between the GPM and the RPM for phase c.

Remark 1: Note that, because of the lack of availability of the real measurements of the voltage phasors (output variables $\mathbf{y}$), they are obtained by running the power flow simulator using real measurements of active and reactive power injections (input variables $\mathbf{X}$).

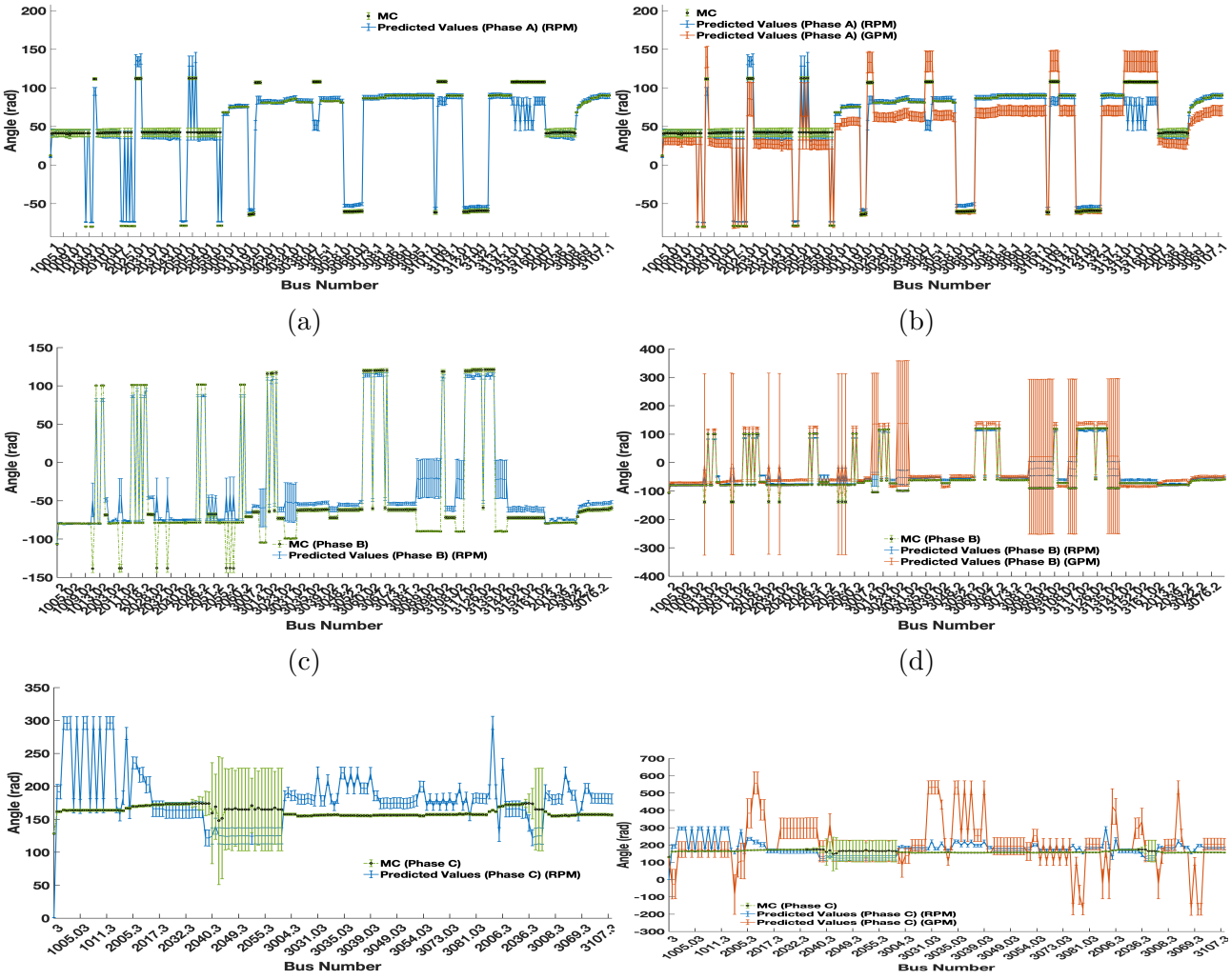


Figure 3.13: The prediction results of voltage angle from the RPM compared with those obtained from the conventional GPM of 240−bus system with 25% of outliers added in training data. (a) The results obtained from the RPM for phase a; (b) comparison between the GPM and the RPM for phase a; (c) The results from the RPM for phase b; (d) comparison between the GPM and the RPM for phase b; (e) The results from the RPM for phase c; (f) comparison between the GPM and the RPM for phase c.

Chapter 4

Robust Gaussian process with Huber likelihood

Let us consider a regression setting $y_i = f(\boldsymbol{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is a homoscedastic i.i.d. random variable with constant variance. In GP models, the systematic dependency between the covariates $\boldsymbol{x}$ and output vector $y \in \mathbb{R}$ is given by a latent function, $f(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$. In a truly non-parametric sense, the latent vector function at n covariates, $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\boldsymbol{x}_n)]^\top$, is assumed to have a priori probability distribution. This distribution is a joint multivariate normal distribution with zero mean vector and covariance matrix, $\mathbf{K}$, that is,

$$\mathbf{f}|\mathbf{X}, \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}). \tag{4.1}$$

The covariance matrix, $\mathbf{K}$, is a positive semi-definite matrix that captures residual spatial association with elements $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i, j = 1, \ldots, n$. The function $k(\cdot, \cdot)$, chosen from a parametric kernel family such as the Gaussian or the Matérn kernel, is characterized by hyperparameters denoted by $\boldsymbol{\theta}$. The likelihood of the data is expressed as $\mathbf{y}|\mathbf{f}, \sigma \sim \mathcal{N}(\mathbf{y}|\mathbf{f}, \boldsymbol{\Sigma})$, and the resulting posterior distribution on $\mathbf{f}$ as where $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$.

Next, we develop three aspects of the proposed GP-Huber model: Huber likelihood, projection pursuit weights, and the resulting unimodal posterior distribution. Following that, we discuss the hyperparametric settings of the GP-Huber.

4.1 Huber likelihood

We propose to use the Huber density function based on the Huber loss proposed by [53] to model the likelihood of the observed data. The Huber loss function $\rho(\cdot)$ is a truncated mixture of two commonly used loss functions: squared loss, $l(r) = r^2$ for residuals below threshold b, and absolute loss, $l(r) = |r|$ for residuals $r_i = y_i - f(\mathbf{x}_i)$ below threshold b, given by

$$\rho(r) = \begin{cases} \frac{1}{2}r^2, & \text{if } |r| \leq b \\ b|r| - \frac{1}{2}b^2. & \text{otherwise} \end{cases} \tag{4.2}$$

[53] considered the contamination model $(1-\varepsilon)G(r) + \varepsilon H(r)$, where $G(r)$ is the Gaussian cumulative density function and $H(r)$ is the unknown cumulative density function. The associated least favorable Huber density function with a fraction of contamination ε is defined as

$$p_H(\mathbf{y}|\mathbf{f}, \phi) = \prod_{i=1}^{n} \frac{1-\varepsilon}{\sqrt{2\pi}\sigma} \exp\left(-\rho(r_i)\right). \tag{4.3}$$

The parameter ε, symbolizing the fraction of the dataset presumed to deviate from the underlying model, can be computed utilizing the minimum covariance determinant estimator [55]. The threshold b is selected to protect estimation of the model parameters and hyperparameters against the fraction of contamination ε.

4.2 Projection pursuit weighting

The idea is to scale the residual r_i associated with the i^{th} data point with projection pursuit weight $w(\boldsymbol{x}_i)$ based on robust variant of Mahalanobis distances, called projection statistics $\text{PS}(\boldsymbol{x}_i) : \mathbb{R}^d \to \mathbb{R}^d$. This scaling highlights the impact of outliers in single or multiple dimensions masking each other in the covariate space. Residual larger than the threshold b

gets robust $L1$ norm treatment, while those smaller than b are treated with an efficient $L2$ norm within the Huber loss $\rho(r)$.

We obtain standardized the residual $r_{S_i} = r_i/(w_i\sigma s)$ by scaling r_i by its corresponding projection pursuit weight w_i and using a scaling factor $s = b_d \text{ med}|\boldsymbol{r}|$, where $b_d = 1 + 5/(n-d)$ is the dimensionality correction factor. When the error distribution is unknown, s accounts for its spread parameter. The projection pursuit weights $\boldsymbol{w}$ limit the influence of outliers simultaneously arising in multiple covariate dimensions at multiple locations on the loss function, are based on projection statistics PS_i, calculated as

$$w_i = \begin{cases} 1, & \text{for } \text{PS}_i^2 \leq c_i, \\ \frac{c_i}{\text{PS}_i^2}, & \text{for } \text{PS}_i^2 > c_i. \end{cases} \tag{4.4}$$

The projection statistics [33, 119] are a robust version of Mahalanobis distances based on the median absolute distance from the median. Formally defined as the maxima of the standardized projection distances obtained by projecting the point cloud in the directions that originate from the co-ordinate wise median and that pass through each of the data points, $\boldsymbol{x}_i$ [83]. They're easy to calculate:

$$\text{PS}_i = \underset{||\boldsymbol{u}_j||=1}{max} \frac{|\boldsymbol{x}_i^T\boldsymbol{u}_j - \underset{k}{\text{median}}(\boldsymbol{x}_k^T\boldsymbol{u}_j)|}{1.4826 \underset{i}{\text{median}} |\boldsymbol{x}_i^T\boldsymbol{u}_j - \underset{k}{\text{median}}(\boldsymbol{x}_k^T\boldsymbol{u}_j)|}, \tag{4.5}$$

where $\boldsymbol{u}_j = \frac{\boldsymbol{x}_j - \mathbf{M}}{||\boldsymbol{x}_j - \mathbf{M}||}$; $j, k = 1, \ldots, n$. The co-ordinate wise median $\mathbf{M}$ is given by $\mathbf{M} = \{\underset{j=1,\ldots,n}{\text{med}} \boldsymbol{x}_{j1}, \ldots, \underset{j=1,\ldots,n}{\text{med}} \boldsymbol{x}_{jd}\}$. The projection statistics attain the maximum breakdown point given by $[(n-d-1)/2]/n$ [78].

[118] and [83] showed that, when $n > 5d$, the squared projection statistics PS_i^2 roughly follow a χ^2 distribution with a degree of freedom equal to the number of non-zero elements ν_i in

the row vector of the associated regressor, $\mathbf{x}_i$, i.e., $\mathrm{PS}_i^2 \sim \chi^2_{\nu_i}$. However, when $n \leq 5d$, it is the PS that roughly follow a χ^2 distribution, that is, $\mathrm{PS}_i \sim \chi^2_{\nu_i}$. Consequently, the threshold c_i is chosen as the 97.5 percentile of the chi-square distribution with ν_i degrees of freedom while defining weights in (4.4).

Throughout the inference process (as detailed in Section 4.4), we use standardized residuals r_{S_i} within the Huber likelihood.

$$p_H(\mathbf{y}|\mathbf{f}, \phi) = \prod_{i=1}^{n} \frac{1-\varepsilon}{\sqrt{2\pi}\sigma} \exp\left(-\rho(r_{S_i})\right). \tag{4.6}$$

4.3 GP-Huber posterior

The posterior distribution resulting from our model, which incorporates a non-conjugate prior, is given as:

$$p(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \sigma) = \frac{p_G(\mathbf{f}|\mathbf{0}, \mathbf{K})}{p(\mathcal{D}|\boldsymbol{\theta}, \sigma)} p_H(\mathbf{y}|\mathbf{f}, \sigma), \tag{4.7}$$

where where $p_G(\mathbf{f}|\mathbf{0}, \mathbf{K})$ is the Gaussian prior $\mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$ and $p_H(\mathbf{y}|\mathbf{f}, \sigma)$ is the likelihood modeled using the Huber density. This formulation leads to a posterior that does not have a closed-form expression due to the non-conjugate nature of the Huber likelihood.

The marginal likelihood (or evidence) of the data, which plays a crucial role in model selection and hyperparameter optimization, is expressed as:

$$p(\mathcal{D}|\sigma, \boldsymbol{\theta}) = \int p_G(\mathbf{f}|\mathbf{0}, \mathbf{K}) p_H(\mathbf{y}|\mathbf{f}, \sigma) d\mathbf{f}. \tag{4.8}$$

Theorem 4.1. *Let $\mathcal{D} = (\boldsymbol{x}_i, y_i)_{i=1}^n$ be a dataset with distinct covariates $\boldsymbol{x}_i \in \mathcal{X}$ and response $y_i \in \mathcal{Y}$, where $n < \infty$. The kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ is positive definite, with elements $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ defined by a continuous kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Assume the*

Huber likelihood function $p_H(\mathbf{y}|\mathbf{f}, \boldsymbol{\sigma})$ *based on strictly convex and continuous Huber loss* $\rho(r_i) : \mathbb{R} \to \mathbb{R}$. *Then the posterior distribution* $p(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \sigma)$ *is unimodal.*

Proof. The GP-Huber posterior distribution is proportional to the expression:

$$p(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \sigma) \propto \exp\left(-\frac{1}{2}\mathbf{f}^\top \mathbf{K}^{-1}\mathbf{f} - \sum_{i=1}^{n} \rho(y_i - f_i)\right), \tag{4.9}$$

where ρ denotes the Huber loss function, which is continuous and strictly convex. The derivative of the log-posterior with respect to $\mathbf{f}$ is:

$$\nabla_{\mathbf{f}} \log p(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \sigma) \propto -\mathbf{K}^{-1}\mathbf{f} - \nabla_{\mathbf{f}} \rho(y_i - f_i), \tag{4.10}$$

and for each component f_i, the derivative becomes:

$$h_i(f_i) = \frac{-(y_i - f_i)}{\sqrt{1 + (y_i - f_i)^2}} - v_i, \tag{4.11}$$

where v_i represents the i^{th} component of $\mathbf{v} = \mathbf{K}^{-1}\mathbf{f}$. The term $\frac{-(y_i - f_i)}{\sqrt{1+(y_i-f_i)^2}}$ is strictly monotonic in f_i, as its derivative is positive for all f_i. Its behavior at the limits is given by:

$$\frac{-(y_i - f_i)}{\sqrt{1 + (y_i - f_i)^2}} = \begin{cases} 0 & \text{if } f_i \to y_i, \\ -1 & \text{if } f_i \to \infty. \end{cases} \tag{4.12}$$

The second term, v_i, arises from the precision matrix $\mathbf{K}^{-1}$, which is symmetric and positive definite. By the spectral theorem, $\mathbf{K}^{-1}$ can be diagonalized as $\mathbf{K}^{-1} = \mathbf{Q}\Lambda\mathbf{Q}^\top$, where Λ is the diagonal matrix of positive eigenvalues, and $\mathbf{Q}$ is an orthogonal matrix. This ensures that v_i is a linear combination of the entries of $\mathbf{f}$ and is therefore continuous and differentiable in f_i. Combining these terms, $h_i(f_i)$ is strictly monotonic because the first term is monotonic

and the second term is linear. By the intermediate value theorem, $h_i(f_i)$ crosses zero exactly once because:

$$\lim_{f_i \to y_i} h_i(f_i) = 0 \quad \text{and} \quad \lim_{f_i \to \infty} h_i(f_i) = -1. \tag{4.13}$$

Thus, $h_i(f_i)$ has a unique root, ensuring that the log-posterior $\log p(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \sigma)$ is strictly concave. The strict concavity of the log-posterior implies that the posterior distribution $p(\mathbf{f}|\mathcal{D}, \boldsymbol{\theta}, \sigma)$ has a unique mode, proving that it is unimodal. □ □

This theorem indicates that despite the non-Gaussian and potentially complex nature of the Huber likelihood, the posterior retains a single peak, simplifying both inference and hyperparameter optimization.

We can set the threshold $b = 1.5$ to achieve high efficiency at the Gaussian distribution (see appendix B.1). This would make our model robust to 10% outliers (since fraction of contamination is $\varepsilon = 0.1$). Note that, in the context of our work, "efficiency" refers to the estimator's ability to achieve low variance when the noise follows a Gaussian distribution. Specifically, a highly efficient estimator can make the best use of data that is predominantly Gaussian, leading to more accurate parameter estimation. The contamination fraction ε defines the model's tolerance to deviations from the Gaussian assumption, allowing it to handle a proportion of outlier points without being overly influenced by them. The parameter b controls the threshold for identifying outliers and thus influences the transition between $L2$ and $L1$ norm treatment. By setting $b = 0.45$, we get $\varepsilon = 0.45$ for heavy-tailed and Gaussian error distributions, we aim to accommodate up to 45% outliers while maintaining reasonable efficiency. The only hyperparameter of the likelihood function requiring estimation is $\phi = \sigma^2$. Thus, the incorporation of projection pursuit weighting and the Huber likelihood does not introduce any extra hyperparameters.

4.4 Approximate Bayesian inference

By retaining the optimization-friendly properties of convex problems ensured by to unimodality (see Theorem 4.1), our method enables the use of the Laplace approximation [131] for the posterior. To facilitate predictions f^*, we develop Gibbs sampling and Laplace's method. The key requirement for the latter is the continuity of the Huber density function, ensuring that its derivatives exist for all r_S in the interval $(-\infty, \infty)$. In Gibbs sampling, the joint posterior distribution $p(\mathbf{f}, \boldsymbol{\theta}, \sigma^2)$ can be simplified using the scale mixture model of the Laplace distribution for data points with residuals $r \geq b$: this representation expresses the likelihood of these points as a normal distribution—making the sampling process more efficient.

4.4.1 Gibbs sampling

The Huber density function is a mixture of a truncated normal and a Laplace density function for an absolute standardized residual respectively lying within and outside the threshold b. This yields

$$p_H(y|f, \boldsymbol{\sigma}) = \begin{cases} \frac{C_1}{\sqrt{2\pi} w_i \sigma_g s} \exp\left(-\frac{r_i^2}{2 w_i^2 \sigma_g^2 s^2}\right) & |r_{S_i}| \leq b, \\ \frac{C_2}{2 w_i a s} \exp\left(-\frac{b|r_i|}{w_i a s}\right) & |r_{S_i}| > b, \end{cases} \tag{4.14}$$

where C_1 and C_2 are the constants respectively, defined as $C_1 = 1 - \varepsilon$ and $C_2 = \sqrt{\frac{\pi}{2}} \exp(b^2/2)$. The Laplace distribution $p_L(y_i|f(\mathbf{x}_i), a)$ with location parameter a can be represented as a scale mixture of normal distributions $\mathcal{N}(y_i|f(\mathbf{x}_i), \sigma_i^2)$ where σ_i^2 follows an exponential distribution $p_E(\sigma_i^2|\beta)$ [5] and $i = 1, \ldots, n_l$ are the indices of the points associated with the standardized residuals larger than the threshold b hereafter referred to as outlying points.

Formally, we have

$$p_L(y_i|f(\mathbf{x}_i), a) = \int p_G(y_i|f(\mathbf{x}_i), \sigma_i^2) p_E(\sigma_i^2|\beta) d\sigma_i^2. \tag{4.15}$$

Using this property, we represent the individual standard deviations corresponding to n_l outlying training points as $\{\sigma_{l_1}, \ldots, \sigma_{l_{n_l}}\}$, which are elements of the vector $\boldsymbol{\sigma_l}$. The variance associated with n_g inlying points is denoted as σ_g^2. Conclusively, the Huber probability density function takes the form

$$\mathbf{y}|\mathbf{f}, \sigma_g^2, \boldsymbol{\sigma}_l^2, \beta \sim \begin{cases} \prod_{i=1}^{n_g} C_1 \mathcal{N}(y_i|f(\mathbf{x}_i), \sigma_g^2) & |r_{S_i}| \leq b, \\ \prod_{i=1}^{n_l} C_2 \mathcal{N}(y_i|f(\mathbf{x}_i), \sigma_{l_i}^2) \text{Exponential}(\sigma_{l_i}^2, \beta) & |r_{S_i}| > b, \end{cases} \tag{4.16}$$

where $n_g + n_l = n$ is the total number of points in the training dataset. An alternative representation of the likelihood function is given by

$$\mathbf{y}_g, \mathbf{y}_l|\mathbf{f}_g, \mathbf{f}_l, \sigma_g^2, \boldsymbol{\sigma}_l^2 \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{y}_g|\mathbf{f}_g \\ \mathbf{y}_l|\mathbf{f}_l \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{gg} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{ll} \end{bmatrix} \right), \tag{4.17}$$

where $\boldsymbol{\Sigma}_{gg}$ and $\boldsymbol{\Sigma}_{ll}$ both are diagonal matrices, the former with constant diagonal elements equal to σ_g^2 and the latter with diagonal entries $\{\sigma_{l_1}^2, \ldots, \sigma_{l_{n_l}}^2\}$. Let the hyperparameter vector $\boldsymbol{\sigma^2}$ consist of the diagonal entries of the matrix $\boldsymbol{\Sigma}_{gg}$, which are $\boldsymbol{\sigma}_g^2$ and $\boldsymbol{\sigma}_l^2$. The joint posterior probability density function of $\mathbf{f}$, $\boldsymbol{\sigma}^2$, and $\boldsymbol{\theta}$ is given by

$$p(\mathbf{f}, \boldsymbol{\sigma}^2, \boldsymbol{\theta}) \propto p(\mathbf{y}|\mathbf{f}, \boldsymbol{\sigma}^2) p_G(\mathbf{f}|\mathbf{0}, \mathbf{K}) p(\boldsymbol{\sigma}^2|\boldsymbol{\beta}) p(\boldsymbol{\beta}|\boldsymbol{\zeta}) p(\boldsymbol{\theta}|\boldsymbol{\zeta}). \tag{4.18}$$

We assume that the hyper-hyperparameter vector $\boldsymbol{\beta}$ and the hyperparameter vector $\boldsymbol{\theta}$ follow the log-uniform distribution with parameters contained in $\boldsymbol{\zeta}$. Since the distribution of the variance parameter σ_g^2 of n_g inlying training points is degenerate, the hyper-hyperparameter

vector $\boldsymbol{\beta} = [\beta_g, \beta_l]^T$ corresponding to the n_g points follows a degenerate distribution as well. Therefore, $p(\sigma_g^2|\beta_g)$ is a Dirac impulse while $\boldsymbol{\sigma}_l^2|\beta_l \sim \text{Exponential}(\boldsymbol{\sigma}_l^2|\beta_l)$. The samples generated from this distribution are highly correlated. Therefore, in order to better mix the Monte Carlo chains, we follow the trick used by [65] as follows:

$$p(\boldsymbol{\sigma}^2, \boldsymbol{\beta}, \boldsymbol{\theta}) \propto \left[\int p_G(\mathbf{y}|\mathbf{f}, \boldsymbol{\Sigma}) p_G(\mathbf{f}|\mathbf{0}, \mathbf{K}) d\mathbf{f}\right] p(\boldsymbol{\sigma}^2|\boldsymbol{\beta}) p(\boldsymbol{\beta}|\boldsymbol{\zeta}) p(\boldsymbol{\theta}|\boldsymbol{\zeta}), \tag{4.19}$$

where the covariance matrix of the n_g inlying samples and the n_l outlying samples is given by $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{gg} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{ll} \end{bmatrix}$. The samples can be used to obtain the approximated probability density functions of the latent vector function, $p(\mathbf{f}^*|\mathcal{D}, \mathbf{X}^*)$, at the new test covariates contained in $\mathbf{X}^*$ by averaging over all unknowns. Formally, we have

$$p(\mathbf{f}^*|\mathcal{D}, \mathbf{X}^*) = \int p(\mathbf{f}^*|\mathbf{f}, \boldsymbol{\sigma}^2, \boldsymbol{\theta}, \mathbf{X}^*, \mathcal{D}) p(\mathbf{f}, \boldsymbol{\sigma}^2, \boldsymbol{\theta}|\mathcal{D}) d\mathbf{f} d\boldsymbol{\sigma}^2 d\boldsymbol{\theta}. \tag{4.20}$$

For T samples, it can be evaluated as

$$p(\mathbf{f}^*|\mathcal{D}, \mathbf{X}^*, \boldsymbol{\zeta}) = \frac{1}{T} \sum_{t=1}^{T} \int p(\mathbf{f}^*|\mathbf{f}, \mathbf{X}, \mathbf{X}^*, \boldsymbol{\theta}_t) p(\mathbf{f}|\mathcal{D}, \boldsymbol{\sigma}_t^2, \boldsymbol{\theta}_t) d\mathbf{f}. \tag{4.21}$$

4.4.2 Laplace approximation

To ensure the continuity of the derivative of the Huber density function with respect to the latent vector function $\mathbf{f}$, we utilize the pseudo-Huber loss function [23], which is defined as

$$\rho(r_S) = b^2 \left(\sqrt{(1 + \left(\frac{r_S}{b}\right)^2} - 1 \right). \tag{4.22}$$

Laplace approximation of the posterior requires the likelihood to be log-concave in order for it to be represented by a unimodal multivariate normal distribution. It is executed by approximating the posterior distribution of $\mathbf{f}$ with a normal distribution [102], that is,

$$\mathbf{f}|\mathcal{D}, \sigma, \boldsymbol{\theta} \sim \mathcal{N}(\hat{\mathbf{f}}|\mathbf{f}, \mathbf{A}). \tag{4.23}$$

A Taylor series expansion about the largest mode of the un-normalized posterior density function of $\mathbf{f}$ yields $q(\mathbf{f}|\mathcal{D}, \sigma, \boldsymbol{\theta}) \approx p_H(\mathbf{y}|\mathbf{f}, \sigma)p_G(\mathbf{f}|\mathbf{0}, \mathbf{K})$. The latter is used to define the MAP estimate $\hat{\mathbf{f}}$, given by

$$\hat{\mathbf{f}} = \underset{f}{arg\,max}\ \ln q(\mathbf{f}|\mathcal{D}, \sigma, \boldsymbol{\theta}), \tag{4.24}$$

which may converge to a local mode in case of multimodal likelihood. As for the posterior covariance matrix, $\mathbf{A}$, it is given by

$$\mathbf{A} = (\mathbf{K}^{-1} + \mathbf{W})^{-1}, \tag{4.25}$$

where $\mathbf{W} = -\nabla\nabla_{\mathbf{f}} \ln \left(p_H(\mathbf{y}|\hat{\mathbf{f}}, \sigma) \right)$. The hyperparameter vector $(\sigma, \boldsymbol{\theta})$ is estimated by maximizing the log of the approximate evidence given by (4.8) using the gradient descent or the conjugate gradient method since the gradient can be analytically derived. Formally, we have

$$(\hat{\sigma}, \hat{\boldsymbol{\theta}}) = \underset{(\sigma,\boldsymbol{\theta})}{arg\,max}\ \ln q(\mathcal{D}|\sigma, \boldsymbol{\theta}), \tag{4.26}$$

where $q(\mathcal{D}|\sigma, \boldsymbol{\theta}) \approx p(\mathcal{D}|\sigma, \boldsymbol{\theta})$ is the approximate log evidence given by

$$\ln q(\mathcal{D}|\sigma, \boldsymbol{\theta}) = \ln p_H(\hat{\mathbf{f}}|\mathbf{f}) - \frac{1}{2}\ln|\mathbf{K}| - \frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f} + \frac{1}{2}\ln|\mathbf{A}|. \tag{4.27}$$

4.5 Experiments

Through our experiments, we aim to address the following questions:

($\mathcal{Q}1$) When is HuberLA (GP-Huber with Laplace's method) preferable, and under which outlier scenarios is HuberMCMC (GP-Huber with Gibbs sampling) more suitable?

($\mathcal{Q}2$) Does GP-Huber show a significant performance improvement over standard GP regression and the RCGP method proposed by [2]?

($\mathcal{Q}3$) Does the use of projection pursuit weighting offer a tangible advantage?

($\mathcal{Q}4$) Does GP-Huber provide more accurate estimates of the planet-to-star radius ratio compared to the standard GP method used by [43] in the transmission spectroscopy experiment?

We performed extensive experiments on benchmark datasets, considering cases of extreme outliers based on their locations, magnitudes, and various error distributions. The threshold b was set to 1.5 for Gaussian error distributions and 0.45 for Student's-t, Laplace, and Cauchy distributions. For all experiments, including the transmission spectroscopy, we used a anisotropic squared exponential kernel function. The mean function is assumed to be zero except for the spectroscopy experiment. Performance was evaluated using root mean square error (RMSE) and mean absolute error (MAE) metrics.

4.5.1 Neal dataset

We evaluate the proposed GP-Huber on the Neal dataset [87] for the following cases of extreme outliers:

Case 1: Extreme outliers $y_i^{(l)}, \boldsymbol{x}_j^{(l)}$ in added in output and covariate dimensions, respectively.

Case 2: Only output dimensions $y_i^{(l)}$ were contaminated with extreme data points.

	SCtMCMC	tLA	HuberMCMC	HuberLA	RCGP	GP	LaplaceMCMC
				$\varepsilon \sim \mathcal{N}(0.01, 0.08)$			
RMSE	0.74 (0.52)	0.75 (1.31)	0.37 (0.42)	**0.25** (**0.25**)	1.84 (0.82)	1.44 (0.90)	0.43 (0.46)
MAE	0.47 (0.25)	0.48 (0.61)	0.31 (0.25)	**0.14** (**0.14**)	1.28 (0.54)	1.24 (0.68)	0.33 (0.26)
				$\varepsilon \sim$ Student-$t(10)$			
RMSE	4.86 (11.56)	1.22 (1.31)	**0.50** (0.81)	1.17 (**0.37**)	1.89 (0.88)	1.52 (0.98)	0.59 (0.93)
MAE	1.67 (1.25)	0.77 (0.65)	**0.41** (0.39)	0.79 (**0.18**)	1.71 (0.85)	1.34 (0.22)	0.43 (0.35)
				$\varepsilon \sim$ Laplace$(0, 0.1)$			
RMSE	4.76 (0.48)	1.23 (1.31)	**0.58** (0.42)	1.17 (**0.35**)	1.95 (0.86)	1.51 (0.89)	1.06 (0.82)
MAE	1.64 (0.23)	0.76 (0.61)	**0.41** (0.24)	0.68 (**0.18**)	1.27 (0.46)	1.23 (0.41)	0.75 (0.34)
				$\varepsilon \sim$ Student-$t(1)$ (Cauchy)			
RMSE	4.75 (0.57)	1.25 (1.32)	**0.61** (0.49)	1.20 (**0.17**)	1.97 (0.62)	1.50 (0.89)	0.42 (0.75)
MAE	1.65 (0.27)	0.78 (0.67)	**0.47** (0.27)	0.81 (**0.11**)	1.78 (0.42)	1.32 (0.66)	0.66 (0.38)

Table 4.1: RMSE and MAE values on the Neal dataset for the Case 1. Values in parentheses represent the performance for Case 3. Bold values highlight the best performance with the lowest RMSE and MAE.

Case 3: Bad data points $y_i^{(c)}$, $\boldsymbol{x}_j^{(l)}$in added to both output and covariate dimensions, respectively, with the former being relatively close to the main data cluster compared to Case 1.

Case 4: Only output dimensions were contaminated with data points $y_i^{(c)}$ relatively close to the data cloud compared to Case 1.

In all the cases above, the locations i and j of the output and covariate outliers may differ or coincide (refer to Appendix B.2.1 for the location and magnitude details on outliers). For each case, we considered four different error distributions: $\mathcal{N}(0.01, 0.08)$, Student-t(10), Laplace$(0, 0.1)$, Student's-t(1).

The baseline models considered for comparison on the Neal dataset, along with RCGP, include: GP with a Student's t error model solved using MCMC integration (SCtMCMC), GP with a Student's t error model using Laplace approximation (tLA), and GP with a Laplace likelihood solved via MCMC integration (LaplaceMCMC). Table 4.1 presents the RMSE and MAE values comparing GP-Huber against these baselines for Cases 1 and 3.

Refer to Appendix B.2.1 for the Tables B.1, B.2 for the Cases 2, 4. Now, we are in position to answer $\mathcal{Q}1$.

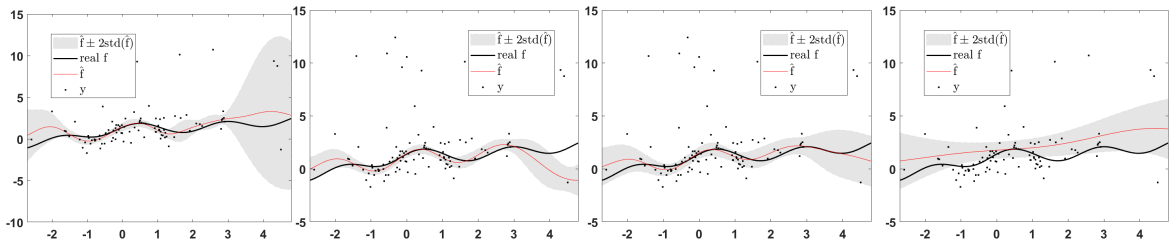
When is HuberMCMC better?

In scenarios with $y^{(l)}, \boldsymbol{x}^{(l)}$ (Case 1), HuberMCMC performed better than HuberLA (see, Tables 4.1 and B.1). HuberMCMC also outperformed tLA in predictive accuracy, demonstrating a more robust fit that is less influenced by $\boldsymbol{x}^{(l)}$ (Figure 4.1). HuberLA generally provided better uncertainty quantification compared to HuberMCMC (see Figures 4.1 and B.2), while maintaining competitive predictive performance. In outlier scenarios with $y^{(l)}$ (Case 2), HuberMCMC exhibited superior performance across Student's-t, Laplace, and Cauchy error distributions (see, Table B.1). This suggests that HuberMCMC is a robust choice for datasets containing extreme output outliers i.e. outlier scenarios similar to Cases 1 and 2.

When is HuberLA better?

HuberLA exhibited superior performance in handling closer output outliers $y^{(c)}$ compared to HuberMCMC (values in parenthesis in the Table 4.1 and Table B.2). Figure B.3 highlights HuberLA's robustness to , in contrast to tLA which is influenced by such points. While HuberLA generally provided more accurate predictions and reliable uncertainty quantification than both HuberMCMC and tLA, HuberMCMC performed competitively for the Cases 3 and 4.

When we did not add $\boldsymbol{x}^{(l)}$ (Cases 2 and 4), HuberLA and HuberMCMC exhibited performance comparable to other baselines, indicating their robustness to exclusively $y^{(l)}$ and $y^{(c)}$.In this case, we did not need to apply projection pursuit weighting on $\boldsymbol{x}$. However, the RMSE and MAE values in Table 4.1 (with projection pursuit weighting) are clearly lower than those in Tables B.1 and B.2 (without weighting). This demonstrates that the weighting mechanism enhances GP-Huber's accuracy, addressing $\mathcal{Q}3$. Compared to the RCGP, both



(a) (b) (c) (d)

Figure 4.1: Predicted values for the Case 1 of the Student's t-error distribution for the Neal dataset obtained from the eight considered GP regression models: (a) SCtMCMC; (b) tLA ; (c) HuberLA; (d) GP.

HuberLA and HuberMCMC consistently produced better predictive performance across all outlier cases and error distributions applied to the Neal dataset. (Please refer to Appendix B.2.1 for Figures B.2, B.3.)

4.5.2 UCI datasets

In this set of experiments, we compared the performance of GP-Huber on the UCI datasets, Energy and Yacht, against RCGP and other baselines: t-GP, m-GP, and standard GP, using the outlier settings from [2]. We specifically focused on the "focused outlier" and "asymmetrical outlier" scenarios, as they closely resemble our extreme and close outlier cases.

	GP	RCGP	t-GP	m-GP	HuberMCMC	HuberLA
			Focused Outliers			
Energy	0.03	**0.02**	0.03	0.24	0.12	0.04
Yacht	0.26	**0.10**	0.20	0.24	0.37	0.28
			Asymmetric Outliers			
Energy	0.54	0.44	0.42	0.41	**0.06**	0.07
Yacht	0.54	0.35	0.41	0.40	**0.29**	0.42

Table 4.2: MAE values for energy and yacht. Bold values indicate the best performance for each row.

MAE values of the comparison are presented in Table 4.2. As expected, HuberLA demonstrates to be more robust than HuberLA since the asymmetrical and focused outliers cases considered in the study of [2] broadly fall under the Cases 3 and 4 in our study. On the Energy dataset, HuberLA outperformed both tLA and RCGP. On the twitter flash crash dataset, HuberLA outperforms RCGP in both RMSE and MAE (see Table 4.3).

	GP	RCGP	HuberMCMC	HuberLA
RMSE	0.354	0.331	0.0118	**0.0021**
MAE	0.154	0.124	0.0089	**0.0014**

Table 4.3: RMSE and MAE for Twitter flash crash.

	RCGP	HuberMCMC	HuberLA
Flashcrash	8.71	26.6	**3.41**
Neal	4.47	6.28	**2.73**

Table 4.4: Processing times (in seconds).

HuberMCMC (Gibbs sampling) and HuberLA (Laplace approximation) have similar computational times to RCGP. HuberLA consistently converged within 2 to 4 seconds, while HuberMCMC showed more variability, with times ranging from 5 to 30 seconds. Table 4.4 shows the processing times for HuberLA and HuberMCMC compared to RCGP on the Twitter flash crash and Neal datasets for Case 1. In our experiments, both HuberMCMC and HuberLA outperformed RCGP and standard GP, with HuberLA showing the best computational efficiency, thus answering $\mathcal{Q}2$.

4.5.3 Transmission spectroscopy

Transmission spectroscopy records the relative change in the stellar flux, which is the incident photons per unit area, as a planet travels in front of the star. The sources of error, such as photon noise and instrumental and astrophysical systematics, raise many potential challenges for precise planet's atmosphere characterization. The goal is to infer the planet to star radius ratio ρ_{radius} from the observed flux as the planet passes in front of the star. The optical state

parameters are metered via auxiliary measurements of the spectral trace such as position, width, angle, or other parameters, indicating the state of the detector and optics, which are thought to be the cause of instrumental systematics. Instead of modeling the latter as a linear function of the optical state parameters, [43] proposed a non-parametric model by leveraging GPs.

The observation set obtained from HST-NICMOS includes the light curves for 18 wavelength channels extracted from $n = 638$ spectra of the planetary system HD-189733. The flux measurements contained in the vector, $\boldsymbol{f} = [f_1, f_2, \ldots, f_n]^T$, are recorded at n time instants, $\{t_1, t_2, \ldots, t_n\}$ and the optical state parameters $\mathbf{x}_{t_i}$ collected in the matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ constitute the training dataset. We extend the work of [43] by using the GP-Huber model to estimate the planet-to-star radius ratio ρ_{radius}. As demonstrated earlier, the robustness to outliers of GP-Huber allows us to utilize 517 measurements associated with four out-of-transit orbits, namely orbit numbers, $\{2, 3, 4, 5\}$, and 137 measurements associated with one in-transit orbit, namely orbit number 1. The latter was excluded from the analysis performed by [43] as it constitutes much larger systematics effects attributed to the spacecraft settling. The observed transit flux modeled in the GP framework follows a normal distribution, that is,

$$\boldsymbol{f}(\boldsymbol{t}, \mathbf{X}) \sim \mathcal{N}(\boldsymbol{T}(\boldsymbol{t}, \boldsymbol{\phi}), \mathbf{K}), \tag{4.28}$$

where the parameter vector, $\boldsymbol{\phi}$, include the parameter of interest, ρ_{radius}, and other parameters. We consider the analytical quadratic limb darkening transit function proposed by [76]. Analogous with (4.16), we assume that the observed transit flux vector, $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{t}, \mathbf{X})$, in the GP-Huber framework follows a normal distribution, that is,

$$\boldsymbol{f} | \boldsymbol{T}(\boldsymbol{t}, \boldsymbol{\phi}), \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2 \sim \mathcal{N}\left(\boldsymbol{T}(\boldsymbol{t}, \mathbf{X}), \boldsymbol{\Sigma} + \mathbf{K}\right). \tag{4.29}$$

The joint un-normalized log-posterior function of $\boldsymbol{\phi}$, $\boldsymbol{\beta}$, and $\boldsymbol{\theta}$ with the gamma aprior probability density function, $p(\boldsymbol{\theta}) = \frac{1}{\boldsymbol{l}}\exp\left(\frac{-\boldsymbol{\theta}}{\boldsymbol{l}}\right)$, over the covariance function hyperparameters is given by

$$\log P(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2, \boldsymbol{\beta}|\boldsymbol{f}, \mathbf{X}, \boldsymbol{\zeta}) = \log\ (\mathcal{L}(\mathbf{r}_S|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2)) - \frac{\tau}{l_\tau} - \sum_{i=1}^{d}\left(\frac{1}{s_i l_i}\right) + \log(\boldsymbol{\beta}) - \boldsymbol{\beta}^T\boldsymbol{\sigma}^2 + \log(p(\boldsymbol{\beta}|\boldsymbol{\zeta})) + \mathrm{C}. \tag{4.30}$$

The challenging task now is to infer the parameter ρ_{radius} from the joint posterior distribution of $(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2, \boldsymbol{\beta})$. The log-likelihood $\mathcal{L}$ term is expressed as

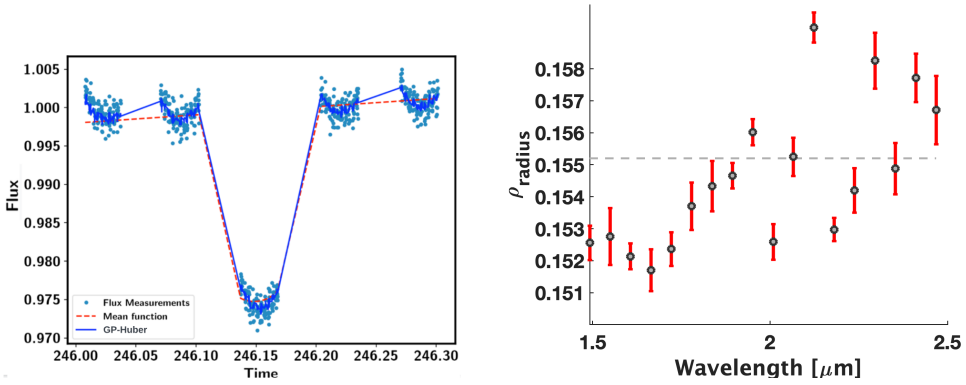
$$\log \mathcal{L}(\boldsymbol{r}_S|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2) = \frac{-1}{2}\mathbf{r}_S^T(\boldsymbol{\Sigma} + \mathbf{K})^{-1}\mathbf{r} - \frac{1}{2}\log|\boldsymbol{\Sigma} + \mathbf{K}| - \frac{n}{2}\log(2\pi) + \log(1-\varepsilon), \tag{4.31}$$

where $\mathbf{r} = \boldsymbol{f} - \boldsymbol{T}(\boldsymbol{t}, \mathbf{X})$. One of the approaches is to use the Bayesian method that seeks the posterior distribution of ρ_{radius} by marginalizing over the other parameters of the mean function parameters $\boldsymbol{\phi}$ and the covariance function hyperparameters, $\boldsymbol{\theta}$ using MCMC methods. The other method proposed as the type-II maximum likelihood method by [43], where the hyperparameters, $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}^2$. Formally, we have

$$(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\sigma}}^2, \hat{\boldsymbol{\beta}}) = \underset{\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2, \boldsymbol{\beta}}{\arg\max} \log P(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2, \boldsymbol{\beta}|\boldsymbol{f}, \mathbf{X}, \boldsymbol{\zeta}). \tag{4.32}$$

And the posterior distribution of the parameter of interest ρ_{radius} is obtained by marginalizing the joint posterior distribution $p(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2, \boldsymbol{\beta})$ over the hyperparameters and the rest of the mean function parameters. In the standard type II maximum likelihood method, the hyperparameters are fixed to their maximum likelihood estimates i.e. by maximizing the evidence $p(\mathcal{D}|\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2)$.

Figure 4.2(a) shows the transit fit obtained for one wavelength channel. Figure 4.2(b) shows the estimated ρ_{radius} obtained using MCMC integration over the rest of the mean function



(a) (b)

Figure 4.2: Transit curve fit and estimated ρ_{radius}. (a) Transit curve mean function $T(t, \boldsymbol{\theta})$ and GP-Huber model fit; (b) results of planet-to-star radius ratios (ρ_{radius}) obtained from GP-Huber with error-bars.

parameters $\boldsymbol{\phi}$ and hyperparameters $\boldsymbol{\theta}$ along with the values estimated from the white light curve represented as the white dashed line. Note that the estimated ρ_{radius} values are very close to the white light curve value of 0.155. Most of our results agree with the results obtained from the Gibson model except for wavelength channels 1.665μm and 2.124μm (see, Appendix B.2.2), which effectively answers $\mathcal{Q}4$.

Our code[1] was implemented in Matlab R2023a with the help of package gpstuff on Intel i7.

[1]https://github.com/apooja1/GpHuber

Chapter 5

Measurement uncertainty quantification in DMD-based Koopman operator approximations

Numerous algorithms have been proposed to approximate the Koopman operator from data [20, 99, 134], with DMD [108] and extended DMD [145] being prominent examples. This chapter is focused on the algorithmic steps of DMD, a method that analyzes system dynamics by extracting dominant modes and their associated temporal dynamics from snapshot data.

Suppose we have $m+1$ data snapshots $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{m+1}$ sampled from a continuous-time system at instances $t_1, \ldots, t_{m+1}$. Let us organize $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m]$ and $\boldsymbol{Y} = [\boldsymbol{x}_2, \ldots, \boldsymbol{x}_{m+1}]$. An estimate of the DMD operator, assuming a full-state observable, is given by

$$\boldsymbol{A} \approx \boldsymbol{X}^{\dagger}\boldsymbol{Y}, \tag{5.1}$$

where $\boldsymbol{X}^{\dagger}$ is the Moore-Penrose pseudoinverse of $\boldsymbol{X}$. The elements of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are assumed to follow a normal distribution $\mathcal{N}(\mu_x, \sigma_x^2)$ and $\mathcal{N}(\mu_y, \sigma_y^2)$, respectively.

This chapter devises a numerical method to quantify the impact of measurement uncertainty on the random variables a_{ij}, elements of $\boldsymbol{A}$. That is, it provides expectations and confidence bounds on a_{ij}. For that, we propagate the first and second moments of $\boldsymbol{X}^{\dagger}$ and $\boldsymbol{Y}$. Although

the moments of $\boldsymbol{Y}$ are straightforward, $\mathbb{E}[y_{kt}] = \mu_{y_{kt}}$ and $\mathbb{E}[y_{kt}^2] = \sigma_{y_{kt}}^2$, we need to derive the first and second moments of $x_{tk}^{\downarrow}$, elements of $\boldsymbol{X}^{\dagger}$. This is done in the next subsection, followed by the moments of a_{ij}, the elements of $\boldsymbol{A}$.

5.1 Element-wise moments of the DMD operator

The Moore-Penrose pseudoinverse of $\boldsymbol{X}$ is given by

$$\boldsymbol{X}^{\dagger} = (\boldsymbol{X}^{\top}\boldsymbol{X})^{-1}\boldsymbol{X}^{\top}, \tag{5.2}$$

and the tk element of $\boldsymbol{X}^{\dagger}$, x_{tk}, can be expressed as

$$x_{tk}^{\downarrow} = \mathbf{e}_k^{\top}(\boldsymbol{X}\boldsymbol{X}^{\top})^{-1}\boldsymbol{x}_t. \tag{5.3}$$

Now, let

$$\boldsymbol{X}\boldsymbol{X}^{\top} = \boldsymbol{V} + \boldsymbol{x}_t\boldsymbol{x}_t^{\top}, \tag{5.4}$$

where $\boldsymbol{V} = \sum_{l=1,l\neq t}^{m} \boldsymbol{x}_l\boldsymbol{x}_l^{\top}$. Using the Sherman–Morrison formula [114], we get

$$(\boldsymbol{X}\boldsymbol{X}^{\top})^{-1} = \boldsymbol{V}^{-1} - \frac{\boldsymbol{V}^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^{\top}\boldsymbol{V}^{-1}}{1 + \boldsymbol{x}_t^{\top}\boldsymbol{V}^{-1}\boldsymbol{x}_t}. \tag{5.5}$$

where $\boldsymbol{V}^{-1}$ is symmetric. Substituting (5.5) into (5.3),

$$x_{tk}^{\downarrow} = \mathbf{e}_k^{\top}\left(\boldsymbol{V}^{-1} - \frac{\boldsymbol{V}^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^{\top}\boldsymbol{V}^{-1}}{1 + \boldsymbol{x}_t^{\top}\boldsymbol{V}^{-1}\boldsymbol{x}_t}\right)\boldsymbol{x}_t, \tag{5.6}$$

where $\kappa = (\boldsymbol{x}_t^{\top}\boldsymbol{V}^{-1}\boldsymbol{x}_t)$ is a scalar, and

$$
\begin{aligned}
x_{tk}^{\downarrow} &= \mathbf{e}_k^\top \left(\frac{\boldsymbol{V}^{-1} + \boldsymbol{V}^{-1}\kappa - \boldsymbol{V}^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^\top\boldsymbol{V}^{-1}}{1+\kappa} \right) \boldsymbol{x}_t \\
&= \frac{\mathbf{e}_k^\top \boldsymbol{V}^{-1}\boldsymbol{x}_t + \mathbf{e}_k^\top \boldsymbol{V}^{-1}\kappa\boldsymbol{x}_t - \mathbf{e}_k^\top \boldsymbol{V}^{-1}\boldsymbol{x}_t\kappa}{1+\kappa} \\
&= \frac{\mathbf{e}_k^\top \boldsymbol{V}^{-1}\boldsymbol{x}_t}{1+\boldsymbol{x}_t^\top \boldsymbol{V}^{-1}\boldsymbol{x}_t}.
\end{aligned} \tag{5.7}
$$

Now, for the sake of simplicity of notation, let $\boldsymbol{R} = \boldsymbol{V}^{-1}$. Let us also define $s_1 = \mathbf{e}_k^\top \boldsymbol{R}\boldsymbol{x}_t$ and $s_2 = 1 + \boldsymbol{x}_t^\top \boldsymbol{R}\boldsymbol{x}_t$. In what follows, we derive the first and second moments of $x_{tk}^{\downarrow}$ using moment generating functions (MGFs). The MGF of a real-valued random variable is an alternative specification of its probability distribution; it encodes all the moments of a random variable into a single function from which they can be extracted again later. An MGF $h : \mathbb{R} \to [0, \infty)$ of a random variable s and parameter p is defined as

$$
h(p) \coloneqq \mathbb{E}[\exp(p \cdot s)]. \tag{5.8}
$$

The Taylor series expansion of the MGF around $p = 0$ is given by

$$
h(p) = \mathbb{E}[\exp(p \cdot s)] = 1 + p\mathbb{E}[s] + \frac{p^2\mathbb{E}[s^2]}{2!} + \frac{p^3\mathbb{E}[s^3]}{3!} + \cdots
$$

The η-th moment about 0 is the η-th derivative of the MGF evaluating at $p = 0$ is expressed as

$$
\mathbb{E}[s^\eta] = \frac{d^\eta}{dp^\eta} h(p) \bigg|_{p=0}.
$$

Note that s_1 and s_2 in (5.7) are both functions of random variables, thus requiring the joint MGF. Following Hoque [51], the joint MGF for a rational function with quadratic forms in

the nominator and denominator is given by

$$h(p_1, p_2) = \mathbb{E}\left[\exp(p_1 \cdot s_1 + p_2 \cdot s_2)\right], \tag{5.9}$$

where p_1 and p_2 are parameters. However, unlike [51] where the matrix that multiplies the vector of random variables is deterministic, here $\boldsymbol{R}$ is stochastic. Therefore, we derive the joint MGF conditioned on $\boldsymbol{R}$ for the expression in equation (5.7). Hence, we redefine the joint MGF in (5.9) is as follows:

$$h(p_1, p_2|\boldsymbol{R}) = \mathbb{E}\left[\exp(p_1 \cdot s_1 + p_2 \cdot s_2)|\boldsymbol{R}\right]. \tag{5.10}$$

Now, drawing parallels to the approach in [51], we apply the integrals by Sawa [106] to derive exact moments of the joint MGF. Note that $h(p_1, -p_2|\boldsymbol{R})$ incorporates the negative sign on p_2 to achieve the quadratic form of $s_2 = 1 + \boldsymbol{x}_t^\top \boldsymbol{V}^{-1}\boldsymbol{x}_t$ in the denominator of the expression for the moments in (5.7). Then, we have:

$$\mathbb{E}\left[(x_{tk}^{\downarrow})^{\eta}\right] \frac{1}{(\eta-1)!} \int_0^\infty p_2^{\eta-1} \left(\frac{\partial^\eta}{\partial p_1^\eta} h(p_1, -p_2|\boldsymbol{R})\right)\Big|_{p_1=0} dp_2, \tag{5.11}$$

to extract the η^{th} moment of $x_{tk}^{\downarrow}$ from (5.10), where η is a positive integer.

Now we are in a position to state the main result of this chapter.

Theorem 5.1. *Let the t-column vector $\boldsymbol{x}_t$ of the data matrix $\boldsymbol{X}$ follow a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma})$. The exact first moment of $x_{tk}^{\downarrow}$, k-element of the t-column vector $\boldsymbol{x}_t^{\downarrow}$*

$$h(p_1, -p_2|\boldsymbol{R}) = \int_{-\infty}^{+\infty} f(\boldsymbol{x}) \exp(p_1 \cdot s_1 - p_2 \cdot s_2) d\boldsymbol{x}$$

$$= \int_{-\infty}^{+\infty} \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_x})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_x})\right) \exp\left(p_1 \mathbf{e}^\top \boldsymbol{R}\boldsymbol{x} - p_2(1+\boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{x})\right) d\boldsymbol{x}$$

$$= \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{\mu}_x^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_x\right) \tag{5.12}$$

$$\int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_x} + p_1 \mathbf{e}^\top \boldsymbol{R}\boldsymbol{x} - p_2 - p_2 \boldsymbol{x}^\top \boldsymbol{R}\boldsymbol{x}\right) d\boldsymbol{x}$$

$$= c_1 \int_{-\infty}^{+\infty} \exp\left(-\boldsymbol{x}^\top \left(\frac{1}{2}\boldsymbol{\Sigma}^{-1} + p_2 \boldsymbol{R}\right)\boldsymbol{x} + (\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_x} + p_1 \boldsymbol{r})^\top \boldsymbol{x}\right) d\boldsymbol{x} \tag{5.13}$$

$$= c_1 \int_{-\infty}^{+\infty} \exp\left(-\boldsymbol{x}^\top \mathbf{S}\boldsymbol{x} + (\mathbf{b} + p_1 \boldsymbol{r})^\top \boldsymbol{x}\right) d\boldsymbol{x} \tag{5.14}$$

---

of $\boldsymbol{X}^\dagger$*, the Moore-Penrose pseudoinverse of* $\boldsymbol{X}$*, is given by*

$$\mathbb{E}[x_{tk}^{\perp}] = \mathbb{E}\left[\exp(p_1 s_1 - p_2 s_2)|\boldsymbol{R}\right]$$
$$= c \int_0^{\infty} |\mathbf{S}|^{-1/2} \exp(-p_2) \exp\left(\frac{\mathbf{b}^\top \mathbf{S}^{-1}\mathbf{b}}{4}\right) \cdot \left(\frac{\boldsymbol{r}^\top \mathbf{S}^{-1}\mathbf{b}}{2}\right) dp_2, \tag{5.15}$$

where $\mathbf{S} = \frac{1}{2}\boldsymbol{\Sigma}^{-1} + p_2 \boldsymbol{R}$*,* $\mathbf{b} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_x}$*,* $\boldsymbol{R}$ *is symmetric with* k*-row (*k*-column) vector* $\boldsymbol{r}^\top$ *(*$\boldsymbol{r}$*), and*

$$c = \frac{1}{2^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{\mu_x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_x}\right).$$

Proof. To simplify the notation, we omit the subscripts in $\boldsymbol{x}$ and $\mathbf{e}$. We develop MGF $h(p_1, -p_2|\boldsymbol{R})$ in (5.14), where $c_1 = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp(-\frac{1}{2}\boldsymbol{\mu}_x^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_x} - p_2)$. Using the Gaussian integral [69] in the form

$$\int \exp\left(-\boldsymbol{u}^\top \boldsymbol{L}\boldsymbol{u} + \boldsymbol{v}^\top \boldsymbol{u}\right) d\boldsymbol{u} = \frac{\pi^{n/2}}{|\boldsymbol{L}|^{1/2}} \exp\left(\frac{\boldsymbol{v}^\top \boldsymbol{L}^{-1}\boldsymbol{v}}{4}\right),$$

$$\frac{\partial h(p_1,-p_2|\boldsymbol{R})}{\partial p_1} = C_2 \exp\left(\frac{\mathbf{b}^\top \mathbf{S}^{-1}\mathbf{b}}{4}\right) \exp\left(\frac{p_1 \boldsymbol{r}^\top \mathbf{S}^{-1}\mathbf{b}}{2} + \frac{p_1^2 \boldsymbol{r}^\top \mathbf{S}^{-1}\boldsymbol{r}}{4}\right) \cdot \left(\frac{\boldsymbol{r}^\top \mathbf{S}^{-1}\mathbf{b}}{2} + \frac{p_1 \boldsymbol{r}^\top \mathbf{S}^{-1}\boldsymbol{r}}{2}\right) \tag{5.16}$$

$$\frac{\partial^2 h(p_1,-p_2|\boldsymbol{R})}{\partial p_1^2} \tag{5.17}$$

$$= c_3 \exp\left(\frac{p_1 \boldsymbol{r}^\top \mathbf{S}^{-1}\mathbf{b}}{2} + \frac{p_1^2 \boldsymbol{r}^\top \mathbf{S}^{-1}\boldsymbol{r}}{4}\right) \left[\frac{(\boldsymbol{r}^\top \boldsymbol{S}^{-1}\boldsymbol{b})}{4}\left(\boldsymbol{r}^\top \boldsymbol{S}^{-1}\boldsymbol{b} + 2p_1 \boldsymbol{r}^\top \boldsymbol{S}^{-1}\boldsymbol{r}\right) + \frac{(\boldsymbol{r}^\top \boldsymbol{S}^{-1}\boldsymbol{r})}{4}\left(p_1^2 \boldsymbol{r}^\top \boldsymbol{S}^{-1}\boldsymbol{r} + 2\right)\right] \tag{5.18}$$

we arrive at the solution of the integral in MGF (5.14):

$$h(p_1,-p_2|\boldsymbol{R}) = c_2 \exp\left(\frac{(\mathbf{b}+p_1\boldsymbol{r})^\top \mathbf{S}^{-1}(\mathbf{b}+p_1\boldsymbol{r})}{4}\right), \tag{5.19}$$

where $c_2 = \frac{1}{2^{n/2}|\mathbf{S}|^{1/2}|\boldsymbol{\Sigma}|^{1/2}} \exp(-\frac{1}{2}\boldsymbol{\mu}_{\boldsymbol{x}}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{\boldsymbol{x}} - p_2)$.

The first derivative of (5.19) with respect to p_1, required to calculate the first moment of $x_{tk}^{\downarrow}$, is given by (5.16). Evaluating (5.16) at $p_1 = 0$ yields

$$\frac{\partial}{\partial p_1} h(p_1,-p_2|\boldsymbol{R})\Big|_{p_1=0} = c_2 \exp\left(\frac{\mathbf{b}^\top \mathbf{S}^{-1}\mathbf{b}}{4}\right) \cdot \left(\frac{\boldsymbol{r}^\top \mathbf{S}^{-1}\mathbf{b}}{2}\right) \tag{5.20}$$

Substituting (5.20) into (5.11),

$$\mathbb{E}[x_{tk}^{\downarrow}] = c \int_0^\infty |\mathbf{S}|^{-1/2} \exp(-p_2) \exp\left(\frac{\mathbf{b}^\top \mathbf{S}^{-1}\mathbf{b}}{4}\right) \cdot \left(\frac{\boldsymbol{r}^\top \mathbf{S}^{-1}\mathbf{b}}{2}\right) dp_2.$$

□

Theorem 5.2. *Let the t-column vector $\boldsymbol{x}_t$ of the data matrix $\boldsymbol{X}$ follow a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma})$. The exact second moment of $x_{tk}^{\downarrow}$, k-element of the t-column vector*

$\boldsymbol{x}_t^{\downarrow}$ of $\boldsymbol{X}^{\dagger}$, the Moore-Penrose pseudoinverse of $\boldsymbol{X}$, is given by

$$\mathbb{E}\left[(x_{tk}^{\downarrow})^2\right] = c\int_0^{\infty} p_2|\mathbf{S}|^{-1/2}\exp(-p_2)\exp\left(\frac{\mathbf{b}^{\top}\mathbf{S}^{-1}\mathbf{b}}{4}\right)\cdot\left(\frac{\boldsymbol{r}^{\top}\mathbf{S}^{-1}\boldsymbol{r}}{2}+\frac{(\boldsymbol{r}^{\top}\mathbf{S}^{-1}\mathbf{b})^2}{4}\right)dp_2. \tag{5.21}$$

Proof. Starting from (5.16), we obtain (5.18), where $c_3 = c_2\cdot\exp\left(\frac{\mathbf{b}^{\top}\mathbf{S}^{-1}\mathbf{b}}{4}\right)$. Evaluating (5.18) at $p_1 = 0$,

$$\frac{\partial^2}{\partial p_1^2}h(p_1,-p_2|\boldsymbol{R})\bigg|_{p_1=0} = c_3\left(\frac{(\boldsymbol{r}^{\top}\mathbf{S}^{-1}\mathbf{b})^2}{4}+\frac{\boldsymbol{r}^{\top}\mathbf{S}^{-1}\boldsymbol{r}}{2}\right) \tag{5.22}$$

and substituting (5.22) into (5.11) yields (5.21). □

Although we assumed independence among the variables in (5.14) by using a diagonal covariance matrix $\boldsymbol{\Sigma}$, the correlations between the variables in $\boldsymbol{x}_t$ can be introduced by including off-diagonal elements in the covariance matrix.

We proceed with an element-wise derivation of the first and second moments of $\boldsymbol{A}$ in (2.7). Recall that our objective is to quantify the impact of measurement uncertainty on the random variables a_{ij}, elements of $\boldsymbol{A}$. Let

$$a_{ij} = x_{i1}^{\downarrow}y_{1j} + \ldots + x_{in}^{\downarrow}y_{nj} = \sum_{k=1}^{n} x_{ik}^{\downarrow}y_{kj}, \tag{5.23}$$

$i = 1,\ldots,m$ and $j = 1,\ldots,m$. The first moment of a_{ij} is

$$\begin{aligned}\mathbb{E}\left[a_{ij}\right] &= \sum_{k=1}^{n}\mathbb{E}\left[x_{ik}^{\downarrow}y_{kj}\right] = \sum_{k=1}^{n}\mathbb{E}\left[x_{ik}^{\downarrow}\right]\mathbb{E}\left[y_{kj}\right]\\ &= \sum_{k=1}^{n}\mathbb{E}\left[x_{ik}^{\downarrow}\right]\mu_{y_{kj}},\end{aligned} \tag{5.24}$$

where $\mu_{y_{kj}}$ is straightforward to estimate using the sample mean. On the other hand, we use the result from Theorem 5.1 to estimate $\mathbb{E}\left[x_{ik}^{\downarrow}\right]$. Let us gather the estimates of $\mathbb{E}\left[x_{ik}^{\downarrow}\right]$ in a matrix $\boldsymbol{M}_{\boldsymbol{X}^{\dagger}}^{(1)}$, and denote its i-row by $\boldsymbol{\mu}_{\boldsymbol{x}_i^{\downarrow}}^{\top}$. Similarly, we gather the estimated values of $\mu_{y_{kj}}$ in a matrix $\boldsymbol{M}_{\boldsymbol{Y}}^{(1)}$ and denote its j-column by $\boldsymbol{\mu}_{\boldsymbol{y}_j}$. Then, an estimate of the first moment of a_{ij} is given by:

$$\widehat{\mu}_{a_{ij}} = \boldsymbol{\mu}_{\boldsymbol{x}_i^{\downarrow}}^{\top} \boldsymbol{\mu}_{\boldsymbol{y}_j}. \tag{5.25}$$

We now focus on obtaining an expression for the element-wise second moments of $\boldsymbol{A}$ in (2.7). To this aim, starting from (5.23),

$$\begin{aligned}
\mathbb{E}[a_{ij}^2] &= \mathbb{E}\left[(x_{i1}^{\downarrow} y_{1j} + \ldots + x_{in}^{\downarrow} y_{nj})^2\right] \\
&= \mathbb{E}\left[(x_{i1}^{\downarrow} y_{1j})^2\right] - \mathbb{E}\left[x_{i1}^{\downarrow}\right]^2 \mathbb{E}\left[y_{1j}\right]^2 + \ldots + \mathbb{E}\left[(x_{in}^{\downarrow} y_{nj})^2\right] - \mathbb{E}\left[x_{in}^{\downarrow}\right]^2 \mathbb{E}\left[y_{nj}\right]^2 \\
&= \mathbb{E}\left[(x_{i1}^{\downarrow})^2\right] \mathbb{E}\left[y_{1j}^2\right] - \mathbb{E}\left[x_{i1}^{\downarrow}\right]^2 \mathbb{E}\left[y_{1j}\right]^2 + \ldots + \mathbb{E}\left[(x_{in}^{\downarrow})^2\right] \mathbb{E}\left[y_{nj}^2\right] - \mathbb{E}\left[x_{in}^{\downarrow}\right]^2 \mathbb{E}\left[y_{nj}\right]^2 \\
&= \sum_{k=1}^{n} \left(\mathbb{E}\left[(x_{ik}^{\downarrow})^2\right] \mathbb{E}\left[y_{kj}^2\right] - \mathbb{E}\left[x_{ik}^{\downarrow}\right]^2 \mathbb{E}\left[y_{kj}\right]^2 \right).
\end{aligned} \tag{5.26}$$

Note that the uncertainty around the k-state x_k is given by its associated variance, σ_k^2. In what follows, we assume the random variables to be *homoscedastic*, that is, all random variables have the same finite variance. This is also known as homogeneity of variance. Thus, the random element y_{kj} follows a normal distribution $\mathcal{N}(\mu_{y_{kj}}, \sigma_k^2)$.

Substituting $\mu_{y_{kj}}$ for the mean and σ_k^2 for the variance of y_{kj} in (5.26), the second moment of a_{ij} is given by:

$$\mathbb{E}\left[(a_{ij})^2\right] = \sum_{k=1}^{n} \mathbb{E}\left[(x_{ik}^{\downarrow})^2\right] \sigma_k^2 - \mathbb{E}\left[x_{ik}^{\downarrow}\right]^2 \mu_{y_{kj}}^2. \tag{5.27}$$

We now make use of the result in Theorem 5.2. Let us gather the estimates of $\mathbb{E}\left[(x_{ik}^{\downarrow})^2\right]$ in $\boldsymbol{M}_{\boldsymbol{X}^{\dagger}}^{(2)}$, and denote its i-row by $(\boldsymbol{\sigma}_{\boldsymbol{x}_i^{\downarrow}}^2)^{\top}$. The confidence levels of the random elements a_{ij},

Algorithm 2 Estimate the first and second moments of $\boldsymbol{A}$

- Obtain the data $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$
- Obtain or calculate $\boldsymbol{\sigma}_{\boldsymbol{y}}^2 = [\sigma_1^2, \ldots, \sigma_n^2]^\top$
- Apply (5.15) to estimate the first moments $\boldsymbol{M}_{\boldsymbol{X}^\dagger}^{(1)}$
- Apply (5.21) to estimate the second moments $\boldsymbol{M}_{\boldsymbol{X}^\dagger}^{(2)}$
- Initialize $i = 1$ and $j = 1$

for $i : 1 : m$ **do**
  for $j : 1 : m$ **do**
    - $m_1(i,j) = \boldsymbol{\mu}_{\boldsymbol{x}_i^\downarrow}^\top \boldsymbol{\mu}_{\boldsymbol{y}_j}$
    - $m_2(i,j) = (\boldsymbol{\sigma}_{\boldsymbol{x}_i^\downarrow}^2)^\top (\boldsymbol{\sigma}_{\boldsymbol{y}}^2) - (\boldsymbol{\mu}_{\boldsymbol{x}_i^\downarrow}^2)^\top (\boldsymbol{\mu}_{\boldsymbol{y}_j}^2)$.

  end for
  - $\widehat{\mu}_{a_{ij}} = m_1(i,j)$
  - $\widehat{\sigma}_{a_{ij}}^2 = m_2(i,j)$

end for

which represent the measurement uncertainties propagated from system states, can finally be estimated as

$$\widehat{\sigma}_{a_{ij}}^2 = (\boldsymbol{\sigma}_{\boldsymbol{x}_i^\downarrow}^2)^\top (\boldsymbol{\sigma}_{\boldsymbol{y}}^2) - (\boldsymbol{\mu}_{\boldsymbol{x}_i^\downarrow}^2)^\top (\boldsymbol{\mu}_{\boldsymbol{y}_j}^2), \tag{5.28}$$

where $\boldsymbol{\sigma}_{\boldsymbol{y}}^2 = [\sigma_1^2, \ldots, \sigma_n^2]^\top$. Algorithm 2 summarizes the process for estimating the confidence levels in terms of the first and second moments of the random elements of $\boldsymbol{A}$.

Remark 5.3. Note that our proposal to quantify the impact of measurement uncertainty on the elements of $\boldsymbol{A}$ is agnostic of specific dynamic mode decomposition methods. This feature is desirable, given the number of variants (see, e.g., [56]) of the original dynamic mode decomposition method [107].

5.2 Experiments

We perform simulations on a spring-mass system and a multi-machine power system. The covariance matrix $\boldsymbol{\Sigma}$ captures the measurement uncertainty of the recorded states; the two

ways to accomplish this are (i) to use the standard deviations of the measurement devices provided by the manufacturers and (ii) to calculate the variance of the ambient measurements using sample variance. We make the blanket assumption of a Gaussian distribution around the recorded data in each instance: $x_{tk} \sim \mathcal{N}(\mathrm{x}_{tk}, \sigma_k^2)$ and $y_{tk} \sim \mathcal{N}(\mathrm{y}_{tk}, \sigma_k^2)$. We employ (5.15) and (5.21) to estimate the element-wise mean and variance of $\boldsymbol{X}^\dagger$, necessary for the DMD method. Following Algorithm 2, we further estimate the first and second moments of the elements of $\boldsymbol{A}$. The estimated moments are compared with those obtained from Monte Carlo simulations, which are the true values. To perform Monte Carlo simulations, N random trajectories of each state are drawn as illustrated in Figure 5.1. Then, N samples of $\boldsymbol{A}$ are obtained using DMD in (2.7) as

$$\boldsymbol{A}^{(l)} = \boldsymbol{X}^{\dagger(l)} \boldsymbol{Y}^{(l)}, \tag{5.29}$$

$l = 1, \ldots, N$. Finally, the first and second moments of a_{ij} are estimated using sample mean $\mu_{a_{ij}} = \frac{1}{N} \sum_{l=1}^{N} a_{ij}^{(l)}$ and sample variance $\sigma_{a_{ij}}^2 = \frac{1}{N-1} \sum_{l=1}^{N} (a_{ij}^{(l)} - \mathrm{a}_{ij})^2$, respectively.

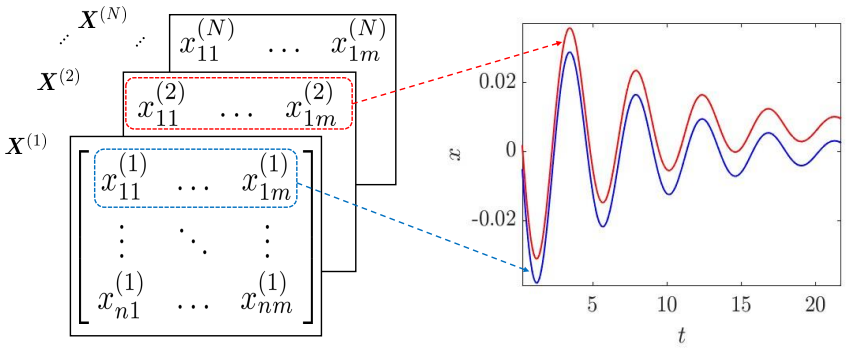


Figure 5.1: Obtaining instances of the random matrix $\boldsymbol{X}$ from the normal distribution of elements $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$.

5.2.1 Spring-mass system

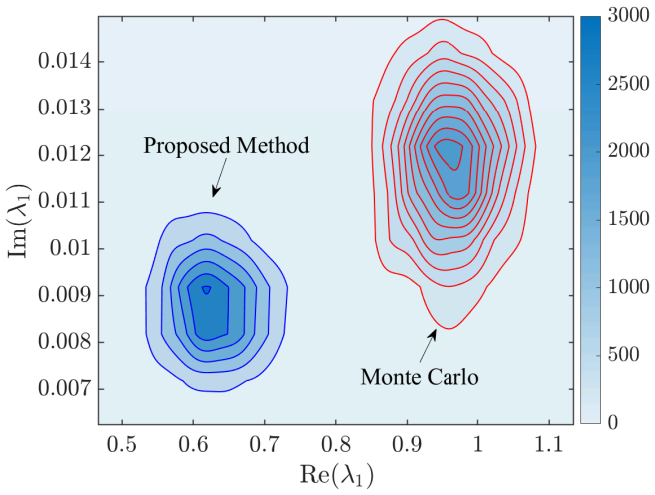


Figure 5.2: Comparison of kernel densities for the largest eigenvalue λ_1 of $\boldsymbol{A}$: Density distributions estimated from samples obtained through our proposed method and Monte Carlo simulation. The colorbar indicates density values.

Consider a spring-mass system with $n = 2$ states: displacement x_1 and velocity x_2, as follows:

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = \frac{-k}{m} x_1 - g,$$

where m is the mass of the object, k is the spring constant, g is the acceleration due to gravity. We choose $m = 5$ kg and $k = 20$ N/m. The simulation with initial conditions set as $\boldsymbol{x}_0 = [0.03; 0.01]$ recorded for 40 seconds constitutes our observed data $\mathcal{D}_{obs} = \{\mathbf{X}, \mathbf{Y}\}$. We estimate σ_1^2 and σ_2^2 using the sample variance of the measurements taken between 30–40 seconds, where they remain steady. The diagonal matrix $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \sigma_2^2)$ now constitutes the uncertainty in the states.

The estimated first (second) moments of a_{ij} employing Algorithm 2 are gathered in $\boldsymbol{M}_{\boldsymbol{A}}^{(1)}$ ($\boldsymbol{M}_{\boldsymbol{A}}^{(2)}$). The matrices $\boldsymbol{M}_{\boldsymbol{A}}^{(1)}$ and $\boldsymbol{M}_{\boldsymbol{A}}^{(2)}$ are compared with those obtained from Monte Carlo

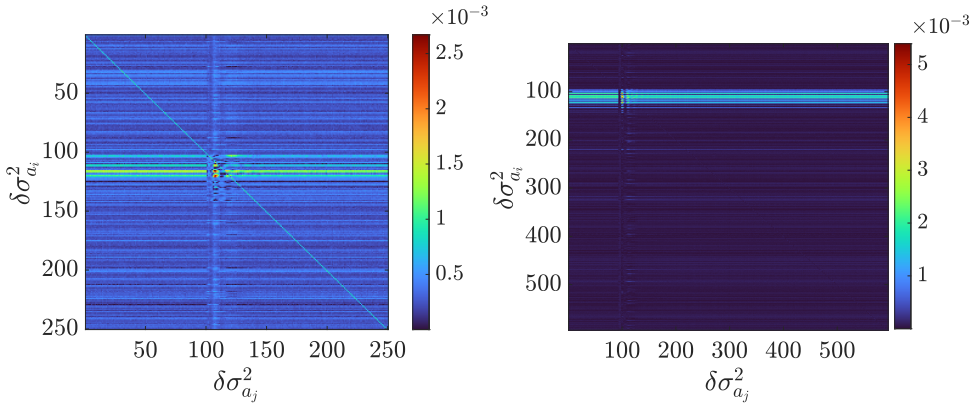
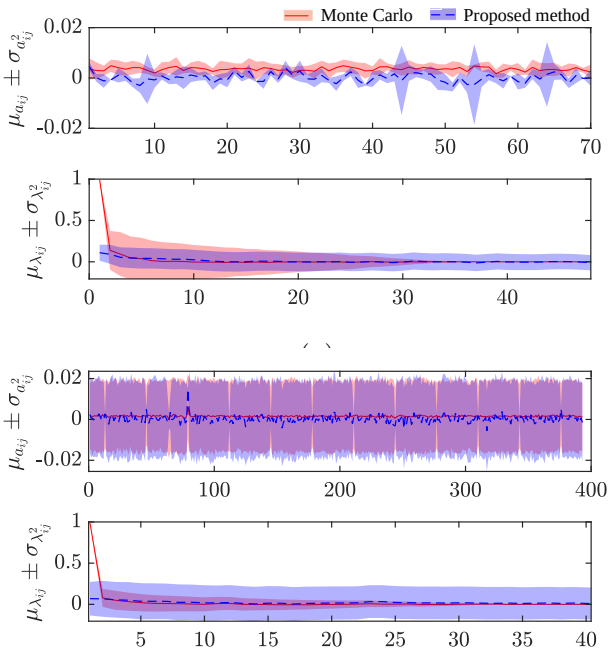


Figure 5.3: Absolute error differences between the estimated $\widehat{\sigma}^2_{a_{ij}}$ and true second moments $\sigma^2_{a_{ij}}$ of DMD operator for (a) event A and (b) event B for the multi-machine power system, where $\delta\sigma^2{}_{a_{ij}} = |\sigma^2_{a_{ij}} - \widehat{\sigma}^2_{a_{ij}}|$.

simulations. A similar comparison is made for the matrices $\boldsymbol{M}^{(1)}_{\boldsymbol{X}^\dagger}$ and $\boldsymbol{M}^{(2)}_{\boldsymbol{X}^\dagger}$ that constitute the first and the second moments of $x^{\downarrow}_{tk}$, respectively. Table 5.1 lists the root mean square error (RMSE), mean absolute error (MAE), Frobenius norm (Fr-norm), and cosine similarity (COS) of the performed comparisons.

Let us obtain $N = 1000$ instances $\boldsymbol{A}^{(l)};\ l = 1, \ldots, N$ of $\boldsymbol{A}$ assuming a normal distribution on its elements a_{ij} with means as their first moments and variances as their second moments estimated using the proposed method as $a^{(l)}_{ij} \sim \mathcal{N}(\widehat{\mu}_{a_{ij}}, \widehat{\sigma}^2_{a_{ij}})$. For each instance $\boldsymbol{A}^{(l)}$, we calculate the eigenvalues $\{\lambda^{(l)}_1 \geq \lambda^{(l)}_2 \geq \ldots \lambda^{(l)}_m\}$ to obtain their N samples. Similarly, let us obtain N instances of eigenvalues of $\boldsymbol{A}^{(l)}$ generated using Monte Carlo simulation. Since the two largest eigenvalues $\{\lambda_1, \lambda_2\}$ form a conjugate pair, we focus on examining the densities of the real and imaginary parts of the largest eigenvalue λ_1. Figure 5.2 compares the kernel density of the largest eigenvalue λ_1 estimated from these samples obtained by both the proposed and Monte Carlo methods, with a squared exponential kernel fit.



(b)

Figure 5.4: Comparison of kernel densities for DMD operator a_{ij} and its eigenvalues λ_i in the case of (a) even A and (b) event B estimated using samples obtained from the proposed method and Monte Carlo simulations for the multi-machine system.

Table 5.1: The RMSE, MAE, Fr-norm, and COS values for the spring-mass system and multi-machine power system

	Spring Mass System				Multi-machine Power System							
					Event A				Event B			
Measure	RMSE	MAE	Fr-norm	COS	RMSE	MAE	Fr-norm	COS	RMSE	MAE	Fr-norm	COS
$\boldsymbol{M}^{(1)}_{\boldsymbol{X}^\dagger}$	3.79e-2	3.32e-2	1.19	1.24	4.94e-2	3.14e-2	4.55	1.86	1.97e-2	1.19e-2	2.80	1.57
$\boldsymbol{M}^{(2)}_{\boldsymbol{X}^\dagger}$	2.45e-3	2.36e-3	7.75e-2	9.75e-1	2.82e^-4	9.15e-5	2.60e-2	5.22e-1	3.39e-5	1.25e-5	4.83e-3	7.10e-1
$\boldsymbol{M}^{(1)}_{\boldsymbol{A}}$	1.04e-3	9.15e-4	5.23e-1	1.41	2.14e-2	1.58e-2	5.35	4.66	8.14e-3	6.16e-3	4.83	4.88
$\boldsymbol{M}^{(2)}_{\boldsymbol{A}}$	4.84e-6	4.48e-6	2.41e-3	9.17e-1	3.38e-4	3.04e-4	8.47e-2	7.28e-1	2.78e-4	1.11e-4	1.6e-1	5.92e-1

5.2.2 Multi-machine power system

The data obtained from time domain simulation of the multi-machine power system [64] is gathered in $\mathcal{D}_{obs}$. For this study, we considered detailed dynamical models associated with synchronous generators that led us to $n = 34$ states in total simulated for $m = 120$ s. The dynamical events are further divided into two sub-events A and B, with measurement uncertainty characterized by the steady-state period from 0 s to 53.99 s and from 63.18 s to 87.20 s, respectively.

Table 5.1 compares $\boldsymbol{M}^{(1)}_{\boldsymbol{X}^\dagger}$, $\boldsymbol{M}^{(2)}_{\boldsymbol{X}^\dagger}$, first and second moments of $\boldsymbol{X}^\dagger$, and $\boldsymbol{M}^{(1)}_{\boldsymbol{A}}$, $\boldsymbol{M}^{(2)}_{\boldsymbol{A}}$, the first and second moments of $\boldsymbol{A}$ estimated using the proposed method with their true values obtained from Monte Carlo simulations. The comparison is conducted using RMSE, MAE, Fr-norm, and COS. Figure 5.3 illustrates the absolute error differences $\delta\sigma^2{}_{a_{ij}} = |\sigma^2_{a_{ij}} - \widehat{\sigma}^2_{a_{ij}}|$ between the true second moment $\sigma^2_{a_{ij}}$ obtained from Monte Carlo simulations and its estimated counterpart $\widehat{\sigma}^2_{a_{ij}}$ using the proposed method.

In Figure 5.4, a detailed comparison of $\mu_{a_{ij}}$ and $\sigma^2_{a_{ij}}$, the first and the second moments of $\boldsymbol{A}$, estimated using the proposed method, are made with those obtained from Monte Carlo simulations. Given the considerable number of elements of $\boldsymbol{A}$ (specifically $m \times m = 6.25e4$ ($3.52e5$) in the case of event A (event B)), visualizing all of them does not help us to compare their estimated moments with the true ones. To address this, we plot the moments of a_{ij} in the intervals of 900 data points for better visualization. Additionally, to ensure

consistency, the second moments are normalized using min-max scaling. Figure 5.4 also compares $\mu_{\lambda_i}, \sigma^2_{\lambda_i}$, the first and second moments of the eigenvalues, estimated by employing sample mean and sample variance on N samples obtained using the proposed method and Monte Carlo simulations as explained above, $i, j = 1, \ldots, m$. The shaded region represents the range within ± 2 second moments from the first moments.

Remarkably, the absolute errors associated with the second moments of most of the DMD operator elements in Figure 5.3 are on the order of 10^{-3}. The estimated first and second moments of $\boldsymbol{A}$ and its eigenvalues in Figure 5.4 exhibit a high degree of comparability with the values obtained from Monte Carlo simulations. Similar is the case for the kernel densities of the eigenvalues compared in Figure 5.2. Further strengthening the reliability of our approach, Table 5.1 illustrates the accurate estimation of moments for $\boldsymbol{X}^\dagger$. Accurate estimations of the first and second moments for $\boldsymbol{X}^\dagger$, $\boldsymbol{A}$, and $\boldsymbol{\lambda}$ in our proposed measurement uncertainty analysis instill trustworthiness in the DMD method.

Chapter 6

Random matrix theory-based quantification of measurement uncertainty in (E)DMD approximations of Koopman operator

The DMD algorithm, as we have seen earlier, involves computation of pseudo inverse of $\boldsymbol{X}$ followed by multiplication with $\boldsymbol{Y}$. The significant effort is dedicated to compute the first and second moments of random elements of $\boldsymbol{X}^{\dagger}$ because the first and second moments of $\boldsymbol{Y}$ are straightforward.

To compute the moments of $x_{ij}^{\downarrow}$, the (i,j)-th element of $\boldsymbol{X}^{\dagger}$, we present Theorem 6.1. It provides formulas for the moments using zero-mean data $\tilde{\boldsymbol{X}}$, which works since we focus on the second moments of a_{ij}.

Theorem 6.1. *Let the columns of* $\tilde{\boldsymbol{X}} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m]' \in \mathbb{R}^{n\times m}$ *be a sample,* $\boldsymbol{x}_t$*, from* $\mathcal{N}_n(0, \boldsymbol{\Sigma})$*,* $n-$ *variate normal probability distribution with a positive definite variance matrix,* $\boldsymbol{\Sigma}$*. The sum of squares matrices, given by* $\boldsymbol{S} = \tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{\top}$*, follows a* n*-variate Wishart distribution with* m *degrees of freedom, denoted by* $W_n(\boldsymbol{\Sigma}, m)$*. If* $m > n+3$*, then the first and second moments*

of $\tilde{\boldsymbol{X}}^{\dagger}$ *are given by:*

$$\mathbb{E}[x_{ij}^{\downarrow}] = 0, \tag{6.1}$$

$$\mathbb{E}[x_{ij}^{\downarrow^2}] = \frac{1}{\sigma_{ij}^2 m(m-n-1)}. \tag{6.2}$$

Proof. By invoking Theorem 2.1 from the seminal work by Cook et al. [26], we establish the first moment of the generalized inverse, $W^{\dagger}$, of a matrix, W, following the distribution $W_n(\mathbf{I}_n, m)$:

$$\mathbb{E}[W^{\dagger}] = \frac{n}{m(m-n-1)}\mathbf{I}_m \tag{6.3}$$

Considering that $\mathbb{E}[W^{\dagger}] = (\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{\top})^{\dagger}$, we derive:

$$\mathbb{E}[\tilde{\boldsymbol{X}}^{\top\dagger}\tilde{\boldsymbol{X}}^{\dagger}] = \frac{n}{m(m-n-1)}\mathbf{I}_m \tag{6.4}$$

Examining the diagonal elements of the mean of this matrix, we observe a direct relationship with the variance of each element:

$$\mathbb{E}[\tilde{\boldsymbol{X}}^{\top\dagger}\tilde{\boldsymbol{X}}^{\dagger}] = n\mathbb{E}[(x_{ij}^{\downarrow})^2]\mathbf{I}_m. \tag{6.5}$$

This relationship leads to the following insightful result:

$$\mathbb{E}[x_{ij}^{\downarrow^2}] = \frac{1}{\sigma_{ij}^2 m(m-n-1)}, \tag{6.6}$$

for a case where $W \sim W_n(\boldsymbol{\Sigma}, m)$ with elements of $\boldsymbol{\Sigma}$ as σ_{ij}^2. □

Theorem 6.2. *As* $n \to \infty, m \to \infty$*, the probability distribution of* $\tilde{\boldsymbol{X}}^{\dagger}$ *approaches the Gaussian distribution, that is,* $x_{ij}^{\downarrow} \sim \mathcal{N}(0, \frac{1}{\sigma_{ij}^2(m-n-1)})$.

Here, measurement uncertainty—quantified in the context of the (E)DMD operator—is a

broader term with a probabilistic basis, attributed to the spread of the estimated value (from numerical procedures such as the Arnoldi approach or singular value decomposition), $\hat{\mathbf{A}}$, from the true value, $\mathbf{A}_0$. Therefore, when we refer to the quantification of measurement uncertainty, we are essentially addressing the quantification of the algorithmic uncertainty.

Recalling the DMD method's algorithmic steps, $\hat{\mathbf{A}} = \boldsymbol{X}^{\dagger}\boldsymbol{Y}$, the expression of the random variable a_{ij} is:

$$a_{ij} = x^{\downarrow}_{i1}y_{1j} + \ldots + x^{\downarrow}_{in}y_{nj} = \sum_{k=1}^{n} x^{\downarrow}_{ik}y_{kj}, \tag{6.7}$$

From (5.24), we note that the first moment of a_{ij} is:

$$\begin{aligned}\mathbb{E}\left[a_{ij}\right] &= \sum_{k=1}^{n} \mathbb{E}\left[x^{\downarrow}_{ik}y_{kj}\right] = \sum_{k=1}^{n} \mathbb{E}\left[x^{\downarrow}_{ik}\right]\mathbb{E}\left[y_{kj}\right] \\ &= \sum_{k=1}^{n} \mathbb{E}\left[x^{\downarrow}_{ik}\right]\mu_{y_{kj}},\end{aligned} \tag{6.8}$$

The data is centered around zero mean reflects $\mu_{y_{kj}} = 0$ and $\mathbb{E}\left[x^{\downarrow}_{ik}\right] = 0$ from Theorem 6.1. Therefore,

$$\mathbb{E}\left[a_{ij}\right] = 0. \tag{6.9}$$

Recalling expression for the second moments of a_{ij} from (5.26):

$$\begin{aligned}\mathbb{E}[a^2_{ij}] &= \mathbb{E}\left[(x^{\downarrow}_{i1}y_{1j} + \ldots + x^{\downarrow}_{in}y_{nj})^2\right] + \mathbb{E}\left[(x^{\downarrow}_{in}y_{nj})^2\right] - \mathbb{E}\left[x^{\downarrow}_{in}\right]^2\mathbb{E}\left[y_{nj}\right]^2 \\ &= \mathbb{E}\left[(x^{\downarrow}_{i1})^2\right]\mathbb{E}\left[y^2_{1j}\right] - \mathbb{E}\left[x^{\downarrow}_{i1}\right]^2\mathbb{E}\left[y_{1j}\right]^2 + \ldots + \mathbb{E}\left[(x^{\downarrow}_{in})^2\right]\mathbb{E}\left[y^2_{nj}\right] - \mathbb{E}\left[x^{\downarrow}_{in}\right]^2\mathbb{E}\left[y_{nj}\right]^2 \\ &= \sum_{k=1}^{n}\left(\mathbb{E}\left[(x^{\downarrow}_{ik})^2\right]\mathbb{E}\left[y^2_{kj}\right] - \mathbb{E}\left[x^{\downarrow}_{ik}\right]^2\mu^2_{y_{kj}}\right).\end{aligned} \tag{6.10}$$

Substituting $\mu_{y_{kj}} = 0$ and $\mathbb{E}\left[(x^{\downarrow}_{ik})^2\right] = \frac{1}{\sigma^2_{ik}(m-n-1)}$ from Theorem 6.1, and $\mathbb{E}\left[(y_{kj})^2\right] = \sigma^2_{kj}$,

Algorithm 3 MUQ in Koopman Operator via DMD

- Obtain the data $\mathcal{D}_{obs} = \{\mathbf{X}_{obs}, \mathbf{Y}_{obs}\}$ and
- Compute $\mathbf{A}_{obs} = \mathbf{X}_{obs}^{\dagger}\mathbf{Y}_{obs}$ using the DMD algorithm
- Obtain the variance matrix $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$
- Generate N random samples around X_{obs}, Y_{obs} following the distribution $X_{ij} \sim \mathcal{N}(X_{obs_{ij}}, \sigma_j^2)$ and $Y_{ij} \sim \mathcal{N}(Y_{obs_{ij}}, \sigma_j^2)$
- Initialize $k = 1, i = 1, j = 1$

for $i : 1 : m$ **do**
  for $j : 1 : m$ **do**
    $-\ S(i,j) = \frac{\sigma_j^2}{\sigma_i^2(m-n-1)}$;
    if $i \neq j$ **then**
      $-S(i,j) = \frac{1}{(m-n-1)}$
    end if
    $-\ j = j + 1$
  end for
  $-\ i = i + 1$
end for

- $\mathbb{E}[a_{ij}] = A_{obs_{ij}}$
- $\mathbb{E}[a_{ij}^2] = S_{ij}$

we get

$$\mathbb{E}[a_{ij}^2] = \frac{\sigma_{kj}^2}{\sigma_{ik}^2 m(m-n-1)}. \tag{6.11}$$

Before deriving the moments for the EDMD operator, let us discuss the significant analytical insights (6.11) sheds: dependencies of second moment of a_{ij} on the variances of the random elements in data matrices, $\boldsymbol{X}$ and $\boldsymbol{Y}$. For the diagonal elements of $\boldsymbol{A}$ (i.e., a_{ij}, where $i = j$), the variance is $\mathbb{E}[a_{ij}^2] = \frac{1}{m(m-n-1)}$, which depends solely on the dimensions of data matrices- $\boldsymbol{X}$ and $\boldsymbol{Y}$. For the off-diagonal elements (a_{ij}, where $i \neq j$)—the variance is inversely related to the variances of the data column $\boldsymbol{x}_i$ at the i^{th} instance and directly related to the variances of the data column $\boldsymbol{y}_j$ at the j^{th} instance.

6.1 Moments of the kEDMD operator

In the following, we derive the first and second moment of a_{ij} in the case of kernel variant of EDMD operator, kEDMD. Since, the EDMD algorithm involves lifting the state space to the higher-dimensions of observable space, a distinctive step from DMD, let us first focus on the probability distributions of lifting functions $\psi_k(\boldsymbol{x}_i)$. Even in the simplest case of dictionary functions - quadratic polynomials $\psi_k(\boldsymbol{x}_i) = [x_{1i}, \ldots, x_{ni}, x_{1i}^2, \ldots, x_{ni}^2]$—introduce state interactions that quickly become overwhelmingly non-linear. Not only does this make it difficult to derive their probability distributions, but the functional dependence also spirals further during operations - inversion of matrix $\mathbf{G}$ and multiplication with matrix $\mathbf{A}$. Therefore, we proceed to use radial basis functions (RBF) $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2\right)$ as the dictionary functions, which map the state space to infinite dimensional Hilbert space, a desirable property in the EDMD algorithm. Therefore, we refer to kernel EDMD [60] to obtain an approximation of Koopman operator generator:

$$\boldsymbol{A} = \boldsymbol{G}^{-1}\boldsymbol{L}, \tag{6.12}$$

where $g_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $l_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{y}_j)$. The expression for a_{ij}, i, j^{th} random element of $\boldsymbol{A}$ is:

$$a_{ij} = g_{i1}^{\downarrow} l_{1j} + \ldots + g_{im}^{\downarrow} l_{mj} = \sum_{k=1}^{m} g_{ik}^{\downarrow} l_{kj}, \tag{6.13}$$

where $g_{ij}^{\downarrow}$ represents i, j^{th}- element of $\boldsymbol{G}^{-1}$, and l_{ij} denotes i, j^{th}- element of $\boldsymbol{L}$. The first and second moments of a_{ij}, $\mathbb{E}[a_{ij}]$ and $\mathbb{E}[a_{ij}^2]$, respectively, analogous to the steps in the case of DMD in (6.8) and (6.10), are expressed as:

$$\mathbb{E}\left[a_{ij}\right] = \sum_{k=1}^{m} \mathbb{E}\left[g_{ik}^{\downarrow}\right] \mathbb{E}\left[l_{kj}\right], \tag{6.14}$$

and

$$\mathbb{E}[a_{ij}^2] = \sum_{k=1}^{m} \left(\mathbb{E}\left[(g_{ik}^{\downarrow})^2\right] \mathbb{E}\left[l_{kj}^2\right] - \mathbb{E}\left[g_{ik}^{\downarrow}\right]^2 \mathbb{E}\left[l_{kj}^2\right] \right). \tag{6.15}$$

Before deriving the first and second moments of the elements of $\boldsymbol{G}^{-1}$ and $\boldsymbol{L}$, we clarify that our derivation applies to both homoscedastic (where noise is identical at every instance, $\boldsymbol{\Sigma}_1 = \ldots = \boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}$) and heteroscedastic (where noise varies across instances, $\boldsymbol{\Sigma}_1 \neq \ldots \neq \boldsymbol{\Sigma}_m$) cases. These are common real-world scenarios, and our approach ensures the proposed measurement uncertainty quantification remains universal. The derivation accommodates both cases, providing the first and second moments of the elements of $\boldsymbol{G}^{-1}$ and $\boldsymbol{A}$, which are necessary for computing the moments of $\boldsymbol{L}$, l_{ij}.

Theorem 6.3. *Let $\boldsymbol{x}_i \sim N(\boldsymbol{\mu}_{x_i}, \boldsymbol{\Sigma}_i)$ and $\boldsymbol{x}_j \sim N(\boldsymbol{\mu}_{x_j}, \boldsymbol{\Sigma}_j)$, where $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_j$ are covariance matrices. Define the RBF kernel function as:*

$$k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{1}{2\gamma}\boldsymbol{z}^T\boldsymbol{z}\right), \tag{6.16}$$

where $\boldsymbol{z} = \boldsymbol{x}_i - \boldsymbol{x}_j$ and $\gamma > 0$. Then the first moment of k_{ij}, $\mathbb{E}[k_{ij}]$, is given by:

$$\mathbb{E}[k_{ij}] = \frac{1}{\sqrt{\det\left(\frac{1}{\gamma}\boldsymbol{I} + \boldsymbol{\Sigma}_z^{-1}\right)}} \exp\left(-\frac{1}{2}\boldsymbol{\mu}_z^T \left(\boldsymbol{\Sigma}_z + \gamma\boldsymbol{I}\right)^{-1} \boldsymbol{\mu}_z\right), \tag{6.17}$$

where $\boldsymbol{\Sigma}_z = \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j$ and p is the dimension of the vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

Proof. Let $\boldsymbol{z} = \boldsymbol{x}_i - \boldsymbol{x}_j$. Since $\boldsymbol{x}_i \sim N(\boldsymbol{\mu}_{x_i}, \boldsymbol{\Sigma}_i)$ and $\boldsymbol{x}_j \sim N(\boldsymbol{\mu}_{x_j}, \boldsymbol{\Sigma}_j)$, the difference $\boldsymbol{z} \sim N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$, where $\boldsymbol{\mu}_z = \boldsymbol{\mu}_{x_i} - \boldsymbol{\mu}_{x_j}$ and $\boldsymbol{\Sigma}_z = \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j$. The expectation of k_{ij} is given by:

$$\mathbb{E}[k_{ij}] = \int_{\mathbb{R}^p} \exp\left(-\frac{1}{2\gamma}\boldsymbol{z}^T\boldsymbol{z}\right) f_z(\boldsymbol{z})\, d\boldsymbol{z}, \tag{6.18}$$

where $f_z(\boldsymbol{z})$ is the PDF of $\boldsymbol{z} \sim N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$:

$$f_z(\boldsymbol{z}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_z|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_z)^T\boldsymbol{\Sigma}_z^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_z)\right). \tag{6.19}$$

Thus,

$$\mathbb{E}[k_{ij}] = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_z|^{1/2}} \int_{\mathbb{R}^p} \exp\left(-\frac{1}{2\gamma}\boldsymbol{z}^T\boldsymbol{z}\right) \times \exp\left(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_z)^T\boldsymbol{\Sigma}_z^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_z)\right) d\boldsymbol{z}. \tag{6.20}$$

Expanding $(\boldsymbol{z}-\boldsymbol{\mu}_z)^T\boldsymbol{\Sigma}_z^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_z)$, the integral becomes:

$$\mathbb{E}[k_{ij}] = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_z|^{1/2}} \exp\left(\frac{-1}{2}\boldsymbol{\mu}_z^T\boldsymbol{\Sigma}_z^{-1}\boldsymbol{\mu}_z\right) \int_{\mathbb{R}^p} \exp\left(-\frac{1}{2}\left(A_z\boldsymbol{z}^T\boldsymbol{z} - 2\boldsymbol{\mu}_z^T\boldsymbol{\Sigma}_z^{-1}\boldsymbol{z}\right)\right) d\boldsymbol{z}, \tag{6.21}$$

where $A_z = \frac{1}{\gamma} + \boldsymbol{\Sigma}_z^{-1}$. Using the Gaussian integral,

$$\int_{\mathbb{R}^p} \exp\left(-\frac{1}{2}A_z\left(\boldsymbol{z}-\boldsymbol{z}_0\right)^T\left(\boldsymbol{z}-\boldsymbol{z}_0\right)\right) d\boldsymbol{z} = \frac{(2\pi)^{p/2}}{\sqrt{\det(A_z)}}, \tag{6.22}$$

we have:

$$\mathbb{E}[k_{ij}] = \frac{\exp\left(-\frac{1}{2}\boldsymbol{\mu}_z^T\left(\boldsymbol{\Sigma}_z+\gamma\boldsymbol{I}\right)^{-1}\boldsymbol{\mu}_z\right)}{\sqrt{\det(A_z)}}. \tag{6.23}$$

Factoring out $\frac{1}{\gamma}$ from the determinant gives:

$$\mathbb{E}[k_{ij}] = \frac{1}{\sqrt{\det(A_z)}} \exp\left(-\frac{1}{2}\boldsymbol{\mu}_z^T\left(\boldsymbol{\Sigma}_z+\gamma\boldsymbol{I}\right)^{-1}\boldsymbol{\mu}_z\right). \tag{6.24}$$

□

Theorem 6.4. *Under the same assumptions as Theorem 6.3, the second moment of* k_{ij},

$\mathbb{E}[k_{ij}^2]$, *is given by:*

$$\mathbb{E}[k_{ij}^2] = \frac{1}{\det\left(\frac{2}{\gamma}\boldsymbol{I} + \boldsymbol{\Sigma}_z^{-1}\right)} \exp\left(-\frac{1}{\gamma}\boldsymbol{\mu}_z^T \left(\boldsymbol{\Sigma}_z + \gamma\boldsymbol{I}\right)^{-1} \boldsymbol{\mu}_z\right) \tag{6.25}$$

Proof. The second moment of k_{ij} follows similarly to Theorem 6.3. Here, we compute:

$$\mathbb{E}[k_{ij}^2] = \int_{\mathbb{R}^p} \exp\left(-\frac{1}{\gamma}\boldsymbol{z}^T\boldsymbol{z}\right) f_z(\boldsymbol{z})\, d\boldsymbol{z}. \tag{6.26}$$

Following the same steps as Theorem 6.3, with the factor $\frac{1}{\gamma}$ instead of $\frac{1}{2\gamma}$, the combined exponent becomes:

$$\left(\frac{2}{\gamma} + \boldsymbol{\Sigma}_z^{-1}\right)\boldsymbol{z}^T\boldsymbol{z} - 2\boldsymbol{\mu}_z^T\boldsymbol{\Sigma}_z^{-1}\boldsymbol{z}. \tag{6.27}$$

We complete the square as in Theorem 6.3 and apply the Gaussian integral:

$$\int_{\mathbb{R}^p} \exp\left(-\frac{1}{2}\boldsymbol{z}^T A_z \boldsymbol{z}\right) d\boldsymbol{z} = \frac{(2\pi)^{p/2}}{\sqrt{\det(A_z)}}, \tag{6.28}$$

where $A_z = \frac{2}{\gamma}\boldsymbol{I} + \boldsymbol{\Sigma}_z^{-1}$. The final result is:

$$\mathbb{E}[k_{ij}^2] = \frac{\exp\left(-\frac{1}{\gamma}\Delta\boldsymbol{\mu}_x^\top(\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j + \gamma\boldsymbol{I})^{-1}\Delta\boldsymbol{\mu}_x\right)}{\det(\boldsymbol{I} + 2\gamma\boldsymbol{\Sigma}_z^{-1})}, \tag{6.29}$$

$\Delta\boldsymbol{\mu}_x = \boldsymbol{\mu}_{x_i} - \boldsymbol{\mu}_{x_j}$. Factoring out γ^p completes the proof:

$$\mathbb{E}[k_{ij}^2] = \frac{1}{\det\left(\frac{2}{\gamma}\boldsymbol{I} + \boldsymbol{\Sigma}_z^{-1}\right)} \exp\left(-\frac{1}{\gamma}\boldsymbol{\mu}_z^T \left(\boldsymbol{\Sigma}_z + \gamma\boldsymbol{I}\right)^{-1} \boldsymbol{\mu}_z\right) \tag{6.30}$$

□

As we have discussed earlier, EDMD fashions the Koopman operator estimation in the least-

squares framework as follows:

$$\boldsymbol{A} = \boldsymbol{G}^{-1}\boldsymbol{L}, \tag{6.31}$$

$$= \begin{bmatrix} g^{\downarrow}_{11} & \dots & g^{\downarrow}_{1n} \\ \vdots & \ddots & \vdots \\ g^{\downarrow}_{m1} & \dots & g^{\downarrow}_{mn} \end{bmatrix} \begin{bmatrix} l_{11} & \dots & l_{1m} \\ \vdots & \ddots & \vdots \\ l_{n1} & \dots & l_{nm} \end{bmatrix}. \tag{6.32}$$

We can generally compute the moments of a_{kj} and g_{kj} by substituting the first and second moments of $g^{\downarrow}_{ij}$ and l_{ij} in (6.14) and (6.15) computed using Theorems 6.3 and 6.4. To obtain the moments of $g^{\downarrow}_{kj}$, we need to do some approximations. When the fluctuations of $\boldsymbol{G}$ are small around its expectation $\mathbb{E}[\boldsymbol{G}]$, we can approximate its inverse:

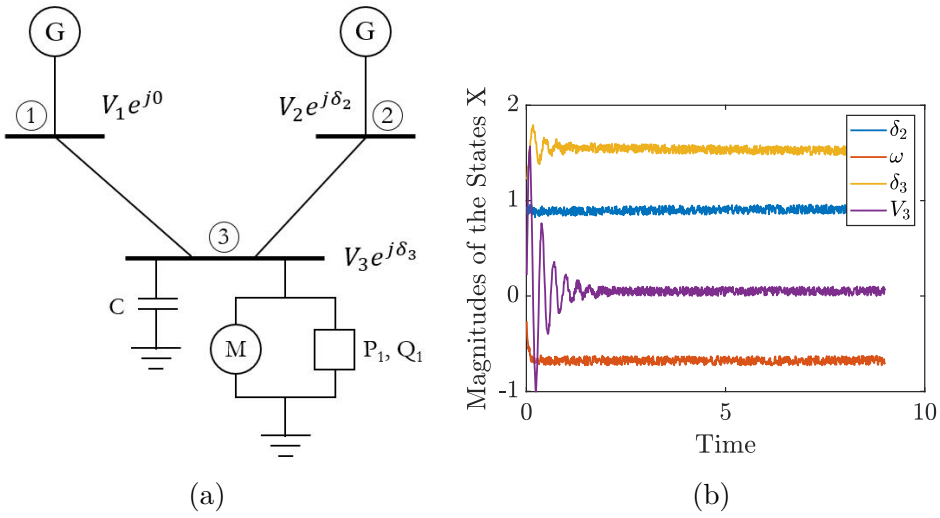
$$\mathbb{E}[\boldsymbol{G}^{-1}] \approx \mathbb{E}[\boldsymbol{G}]^{-1}. \tag{6.33}$$

To satisfy this condition, $\boldsymbol{A}$ must be positive definite and well-conditioned, ensuring the existence of its inverse. The perturbation approximation relies on two criteria: a condition number less than 1 and $\text{var}(a_{ij}) \leq E[a_{ij}]^2$ being sufficiently small. In this study, since the data is centered around zero, these conditions are met.

6.2 Probability distributions of the eigenvalues of the DMD operator

The eigenvalues of a symmetric random matrix follow the Marchenko-Pastur distribution [77], expressed as:

$$f(x;\lambda,\sigma^2) = \frac{\sqrt{(\lambda_+ - x)(x - \lambda_-)}}{2\pi\sigma^2 x}, \tag{6.34}$$



(a) (b)

Figure 6.1: (a) 3-bus system with 2 generators and a load and (b) data gathered in $\mathbf{D}$ from the simulation of the 3-bus system.

with $\lambda_{\pm} = \sigma^2(1 \pm \sqrt{\lambda})^2$. Here, x is the variable, λ_+ and λ_- are the upper and lower bounds of the spectrum, and σ^2 is the variance of the entries of the random matrix. First, we will use this knowledge to construct the eigenvalues of the symmetric matrix, constructed as $\boldsymbol{A}_{sym} = \frac{\boldsymbol{A}\boldsymbol{A}^\top}{m}$. The probability distribution of the eigenvalues of the random matrix, $\boldsymbol{A}_{sym}$, can be described by the probability density function defined in (6.34).

To derive the distribution of the eigenvalues of the DMD operator, $\lambda_1, \ldots, \lambda_m$ of $\boldsymbol{A}$, which is the quantity of our interest, we take a positive square root of the eigenvalues $s_1, \ldots, s_m$ of $\boldsymbol{A}_{sym}$.

6.3 Numerical analysis

In this section, we discuss the implementation of the proposed uncertainty quantification in the data-driven identification of nonlinear dynamics using measurements from both a simulated example power system and real-world data.

6.3.1 Three-Bus Example

Consider a power network with two generators and a load, as shown in Figure 6.1 (a). The load is modeled as an induction motor in parallel with a constant PQ load. The system is modeled as a four-dimensional dynamical system with the states being generator angle (δ_2), generator angular velocity (ω), load angle (δ_3), and load voltage magnitude (V_3). For a detailed analysis of the dynamics and parameters of the system, we refer the reader to [1, 32].

Let us obtain the measurements of the $n = 4$ dynamic states $\{\delta_2, \omega, \delta_3, V_3\}$ by solving the continuous differential equations of the given system for the duration T. We stack these so-called observed quantities in the data matrix, $\mathcal{D}_{obs} \in \mathbb{R}^{T\times n} = [\boldsymbol{\delta}_2, \boldsymbol{\omega}, \boldsymbol{\delta}_3, \boldsymbol{V}_3]'$, shown in Figure 6.1 (b). Let us form the observed input matrix using the $T-1$ rows of the matrix $\mathbf{D}$ as $\mathbf{X}_{obs} \in \mathbb{R}^{T-1\times n} = [\mathbf{d}_1, \ldots, \mathbf{d}_{T-1}]$ and the observed output matrix $\mathbf{Y}_{obs} \in \mathbb{R}^{T-1\times n} = [\mathbf{d}_2, \ldots, \mathbf{d}_T]$. The Koopman operator is estimated using the observations $\mathcal{D}_{obs} = \{\mathbf{X}_{obs}, \mathbf{Y}_{obs}\}$ from the DMD algorithm as $\mathbf{A}_{obs} = \mathbf{X}_{obs}^{\dagger}\mathbf{Y}_{obs}$.

Let us now characterize the measurement uncertainty using the variance matrix, $\mathbf{\Sigma}$, calculated from the steady-state measurements from the time, approximately 5 s to 10 s, which is given as $\mathbf{\Sigma} = \text{diag}(0.39745, 0.0026435, 0.5936, 0.23668)$. Now, with the variance, $\mathbf{\Sigma}$, and mean, $\mathbf{X}_{obs}$, i.e., $x_{ij} \sim \mathcal{N}(x_{obs_{ij}}, \Sigma_{jj})$ and $y_{ij} \sim \mathcal{N}(y_{obs_{ij}}, \Sigma_{jj})$, we can draw $N = 1000$ random samples, $\mathcal{D}^{(k)} = \{\boldsymbol{X}^{(k)}, \boldsymbol{Y}^{(k)}\}_{k=1}^{N}$, to conduct a Monte Carlo simulation. The realizations of the DMD operator, $\boldsymbol{A}^{(k)}$; $k = 1, \ldots, N$, are obtained by employing the DMD algorithm.

The variance of the elements of the Koopman operator, $\hat{R}_{ij} = \text{Var}(a_{ij}) = \sum_{k=1}^{N} \frac{(a_{ij}^{(k)} - a_{obs_{ij}})^2}{N-1}$, is estimated using the sample variance, which represents the true variance of the DMD operator for comparison. We employ Algorithm 3 to estimate variances of the elements of the DMD operator, $\mathbf{S}$. The DMD operator is sampled from the distribution: $a_{ij} \sim \mathcal{N}(a_{obs_{ij}}, \hat{S}_{ij})$. The

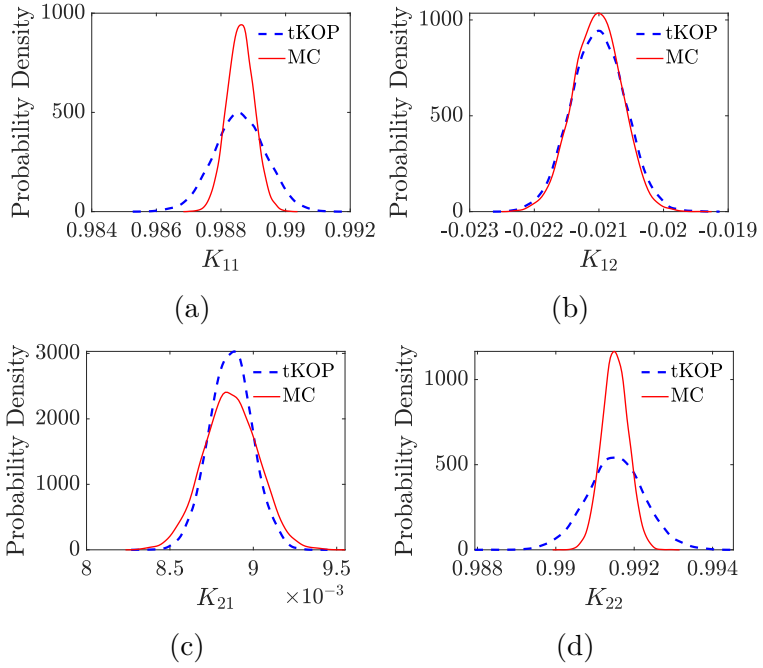


Figure 6.2: Comparison between the true variance, R, and the ones obtained from the proposed MUQ algorithm, S, for the elements (a) a_{11}, (b) a_{12}, (c) a_{21}, (d) a_{22} of the DMD operator, $\mathbf{A}$. The results generated using the proposed method are referred to as tKOP.

results are compared with the Monte Carlo method (considered as true values) where the elements of the DMD operator are sampled from $a_{ij} \sim \mathcal{N}(a_{obs_{ij}}, \hat{R}_{ij})$. The comparison of the results is shown in Figure 6.2. Similarly, we also analyze the given system in the function space using polynomial dictionary functions.

Figure 6.2 show that the uncertainty quantification obtained by employing Algorithm 3 is quite close to the variance values obtained from Monte Carlo simulations for the single-machine system.

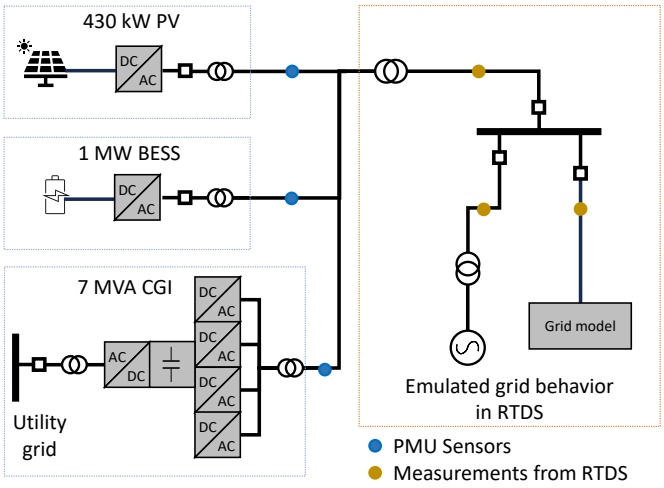
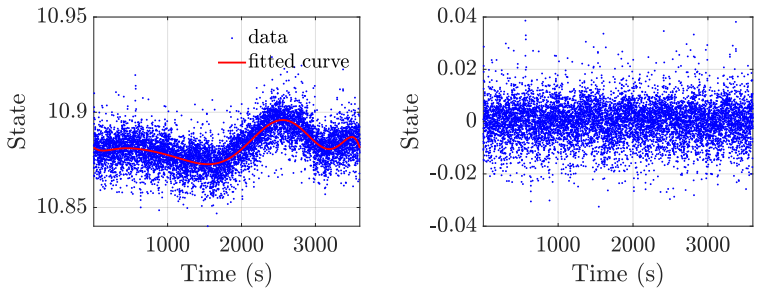


Figure 6.3: Power-Hardware-In-Loop (PHIL) setup for grid emulation



(a) (b)

Figure 6.4: (a) The data plot for State 1 and (b) the data with subtracted regressed values.

6.3.2 Analysis on real sensor measurements

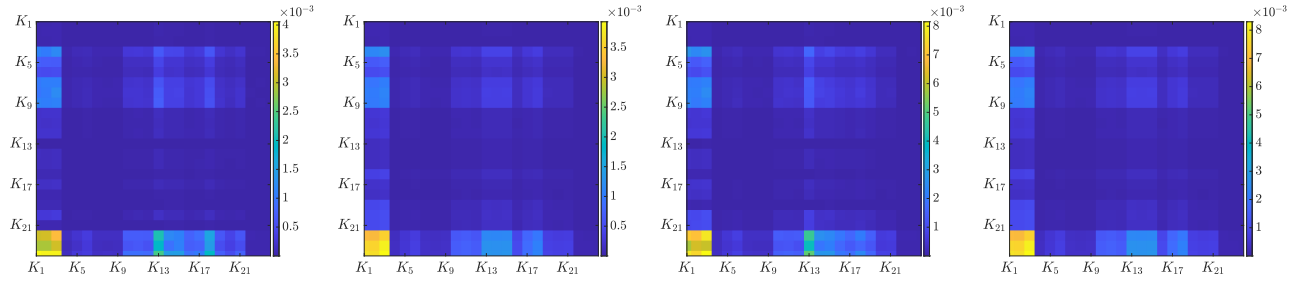
System setup

The real-world data comprises phasor measurement unit measurements that capture the voltage magnitude and phase angles of a hardware-in-loop system (NREL ARIES) as shown in Figure 6.3. Recorded at a sampling frequency of 100 samples per second, the data span a duration of 10 min. Figure 6.4 (a) illustrates one of the steady state states observed.

To quantify measurement uncertainty for each state, we fit the measured data using a 9th-

order polynomial regression (see Figure 6.4(b)). The variance of the residuals serves as the uncertainty estimate for the state.

Results on the element-wise distributions of the DMD operator



(a) (b) (c) (d)

Figure 6.5: Comparison between (a) the true variance, R, obtained from Monte Carlo and (b) the ones obtained from the proposed algorithm, S, via DMD; (c) variances obtained from Monte Carlo and (d) proposed algorithm for EDMD.

Figure 6.5 presents a comparison between the true variance values obtained from the Monte Carlo simulation and those derived from our proposed MUQ method that employs the DMD and EDMD estimation algorithms.

Chapter 7

Conclusions and future research

7.1 Conclusions

This dissertation explores robust and data-driven uncertainty quantification (UQ) methods to address the challenges of modern, high-dimensional systems under dynamic and uncertain conditions. A key focus was on developing statistical frameworks that combine robust modeling techniques with data-driven approaches to enhance the reliability of predictions in real-time decision-making contexts. The robust process model (RPM) was demonstrated as an effective tool for handling stochastic power flow calculations, accommodating uncertainties introduced by renewable energy sources (RES) and distributed generation (DG). The GP-Huber model further expanded the scope by addressing heavy-tailed distributions and extreme outliers in both covariate and output dimensions, offering computationally efficient solutions for various real-world applications. Additionally, the dissertation proposed a novel measurement uncertainty quantification (MUQ) methodology for the Koopman operator, leveraging random matrix theory to enable robust system control and stability assessments.

The proposed frameworks were validated through rigorous experiments across various domains, including power systems, renewable energy integration, and nonlinear dynamical systems. Collectively, this work underscores the importance of integrating robust UQ methods into data-driven models to achieve greater accuracy, adaptability, and resilience in the face of uncertainty.

7.2 Directions of Future Research

In future work, several promising directions emerge to further enhance the methodologies developed in this dissertation. Building on the RPM's robustness, future research will focus on extending its application to optimal power flow calculations, particularly in systems with more than 25% outliers. High breakdown estimators will be explored to enhance performance under extreme data corruption scenarios.

To handle the increasing size of datasets in real-world applications, future efforts will aim at implementing sparse inference techniques for the GP-Huber model. These advancements will ensure scalability while preserving robustness against skewed error distributions and highly corrupted data. Beyond the first and second moments explored in this work, deriving higher-order moments such as skewness and kurtosis will provide deeper insights into uncertainty distributions. This direction will be complemented by extending the MUQ methodology to non-normal cases, improving its applicability to diverse real-world systems.

Bridging robust UQ methods with real-time decision-making systems remains a critical goal. Applications such as energy management, autonomous systems, and robotics will benefit from UQ-driven feedback mechanisms to improve reliability and performance. Future studies will derive analytical expressions for variances in Koopman operator tuples, establishing stronger theoretical foundations. Additionally, exploring the interplay between uncertainty quantification and modal analysis will expand its utility in areas like stability evaluation and control optimization.

By pursuing these research directions, this work sets the stage for advancing the capabilities of data-driven models and robust uncertainty quantification methods, ensuring their effectiveness across an ever-growing array of dynamic and complex systems.

Bibliography

[1] V. Ajjarapu and B. Lee. Bifurcation theory and its application to nonlinear dynamical phenomena in an electrical power system. *IEEE Trans. Power Syst.*, 7(1):424–431, 1992.

[2] Matias Altamirano, Francois-Xavier Briol, and Jeremias Knoblauch. Robust and conjugate gaussian process regression. In *Forty-first International Conference on Machine Learning*, 2024.

[3] Naomi Altman and Martin Krzywinski. The curse (s) of dimensionality. *Nat Methods*, 15(6):399–400, 2018.

[4] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in neural information processing systems*, 33:14927–14937, 2020.

[5] David F Andrews and Colin L Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(1):99–102, 1974.

[6] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[7] A. M. Avila and I. Mezić. Data-driven analysis and forecasting of highway traffic dynamics. *Nat. Commun.*, 11(1), 2020.

[8] Emilio Barocio, Bikash C. Pal, Nina F. Thornhill, and Arturo Roman Messina. A dynamic mode decomposition framework for global power system oscillation analysis. *IEEE Trans. Power Syst.*, 30(6):2902–2912, 2015.

[9] Leonardo S Bastos and Anthony O'hagan. Diagnostics for gaussian process emulators. *Technometrics*, 51(4):425–438, 2009.

[10] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

[11] Thomas Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, (53):370–418, 1763.

[12] T. Bedford and R. M. Cooke. Vines: A new graphical model for dependent random variables. *Ann. Stat.*, 30(4):1031–1068, 2002. URL http://www.jstor.org/stable/1558694.

[13] Lori Bird, Michael Milligan, and Debra Lew. Integrating variable renewable energy: Challenges and solutions. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2013.

[14] Christopher M Bishop. Mixture density networks. 1994.

[15] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[16] Daniel Bruder, Xun Fu, R. Brent Gillespie, C. David Remy, and Ram Vasudevan. Data-driven control of soft robots using Koopman operator theory. *IEEE Trans. Robot.*, 37 (3):948–961, 2021. doi: 10.1109/TRO.2020.3038693.

[17] Steven L. Brunton. Notes on Koopman operator theory. 2019. https://api.semanticscholar.org/CorpusID:229348032.

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

[19] Fankun Bu, Yuxuan Yuan, Zhaoyu Wang, Kaveh Dehghanpour, and Anne Kimber. A time-series distribution test system based on real utility data. In *2019 North American Power Symposium (NAPS)*, pages 1–6. IEEE, 2019.

[20] Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4), 2012.

[21] Russel E Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7: 1–49, 1998.

[22] Bradley P Carlin and Thomas A Louis. Empirical bayes: Past, present and future. *Journal of the American Statistical Association*, 95(452):1286–1289, 2000.

[23] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on image processing*, 6(2):298–311, 1997.

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

[25] Matthew J Colbrook and Alex Townsend. Rigorous data-driven computation of spectral properties of Koopman operators for dynamical systems. *Commun. Pure Appl. Math.*, 77(1):221–283, 2024.

[26] R. Dennis Cook and Liliana Forzani. On the mean and variance of the generalized inverse of a singular Wishart matrix. *Electron. J. Stat.*, 5:146–158, 2011.

[27] Karel Crombecq, Eric Laermans, and Tom Dhaene. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. *European Journal of Operational Research*, 214(3):683–696, 2011.

[28] Andreas Damianou and Neil D Lawrence. Deep gaussian processes. In *Artificial intelligence and statistics*, pages 207–215. PMLR, 2013.

[29] Scott T M Dawson, Maziar S Hemati, Matthew O Williams, and Clarence W Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Exp. Fluids*, 57, 2014.

[30] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):408–423, 2013.

[31] Abdourahmane Diaw, Michael McKerns, Irina Sagert, LG Stanton, and Michael S Murillo. Efficient learning of accurate surrogates for simulations of complex systems. *Nature Machine Intelligence*, pages 1–10, 2024.

[32] I. Dobson, H.-D. Chiang, J.S. Thorp, and L. Fekih-Ahmed. A model of voltage collapse in electric power systems. In *IEEE Conf. Decis. Control*, volume 3, pages 2104–2109, 1988.

[33] David L Donoho. Breakdown properties of multivariate location estimators. Technical report, Technical report, Harvard University, Boston. URL http://www-stat.stanford …, 1982.

[34] Daniel Duke, Julio Soria, and Damon Honnery. An error analysis of the dynamic mode decomposition. *Exp. Fluids*, 52:529–542, 2012.

[35] Oliver G Ernst, Antje Mugler, Hans-Jörg Starkloff, and Elisabeth Ullmann. On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(2):317–339, 2012.

[36] Aurélien Falco, Tiziano Zingales, William Pluriel, and Jérémy Leconte. Toward a multidimensional analysis of transmission spectroscopy-i. computation of transmission spectra using a 1d, 2d, or 3d atmosphere structure. *Astronomy & Astrophysics*, 658: A41, 2022.

[37] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 3266–3273. IEEE, 2018.

[38] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

[39] David A Freedman. On the so-called "huber sandwich estimator" and "robust standard errors". *The American Statistician*, 60(4):299–302, 2006.

[40] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

[41] Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

[42] Alan E Gelfand, Susan E Hills, Amy Racine-Poon, and Adrian FM Smith. Illustration

of bayesian inference in normal data models using gibbs sampling. *Journal of the American Statistical Association*, 85(412):972–985, 1990.

[43] NP Gibson, Suzanne Aigrain, S Roberts, TM Evans, Michael Osborne, and F Pont. A gaussian process framework for modelling instrumental systematics: application to transmission spectroscopy. *Monthly notices of the royal astronomical society*, 419(3): 2683–2694, 2012.

[44] Joseph A Goguen. La zadeh. fuzzy sets. information and control, vol. 8 (1965), pp. 338–353.-la zadeh. similarity relations and fuzzy orderings. information sciences, vol. 3 (1971), pp. 177–200. *The Journal of Symbolic Logic*, 38(4):656–657, 1973.

[45] David E Golberg. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989(102):36, 1989.

[46] A. Hanea, D. Kurowicka, R. M. Cooke, and D. Ababei. Mining and visualising ordinal data with non-parametric continuous bbns. *Comput. Stat. Data Anal.*, 54(3):668–687, 2010.

[47] Arman Hasanzadeh, Ehsan Hajiramezanali, Shahin Boluki, Mingyuan Zhou, Nick Duffield, Krishna Narayanan, and Xiaoning Qian. Bayesian graph neural networks with adaptive connection sampling. In *International conference on machine learning*, pages 4094–4104. PMLR, 2020.

[48] Simon Haykin. *Neural networks and learning machines, 3/E*. Pearson Education India, 2009.

[49] Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. Bayesian deep ensembles via the neural tangent kernel. *Advances in neural information processing systems*, 33: 1010–1022, 2020.

[50] Paul Hewson. Bayesian data analysis 3rd edn a. gelman, j. b. carlin, h. s. stern, d. b. dunson, a. vehtari and d. b. rubin, 2013 boca raton, chapman and hall–crc 676 pp., isbn 1-4398-4095-4. *Journal of The Royal Statistical Society: Series A (Statistics in Society)*, 178(1):301–301, 2015. doi: 10.1111/j.1467-985X.2014.12096_1.x.

[51] Asraul Hoque. The exact moments of forecast error in the general dynamic model. *Sankhya: Indian J. Stat.*, pages 128–143, 1985.

[52] Zhen Hu and Sankaran Mahadevan. Bayesian network learning for data-driven design. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 4(4):041002, 2018.

[53] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

[54] Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

[55] Mia Hubert and Michiel Debruyne. Minimum covariance determinant. *Wiley interdisciplinary reviews: Computational statistics*, 2(1):36–43, 2010.

[56] Sara M. Ichinaga, Francesco Andreuzzi, Nicola Demo, Marco Tezzele, Karl Lapo, Gianluigi Rozza, Steven L. Brunton, and J. Nathan Kutz. PyDMD: A Python package for robust dynamic mode decomposition. *Preprint, arXiv:2402.07463*, 2024.

[57] Myong Chol Jung, He Zhao, Joanna Dipnall, and Lan Du. Beyond unimodal: Generalising neural processes for multimodal uncertainty estimation. *Advances in Neural Information Processing Systems*, 36, 2024.

[58] M. H. Kalos and P. A. Whitlock. *Monte Carlo Methods*. John Wiley & Sons, Hoboken, NJ, 2009.

[59] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6): 422–440, 2021.

[60] I Kevrekidis, Clarence W Rowley, and M Williams. A kernel-based method for data-driven koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2016.

[61] Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

[62] Igor Kononenko. Bayesian neural networks. *Biological Cybernetics*, 61(5):361–370, 1989.

[63] Laura Kreidberg. Exoplanet atmosphere measurements from transmission spectroscopy and other planet-star combined light observations. *arXiv preprint arXiv:1709.05941*, 2017.

[64] Prabha S. Kundur and Om P. Malik. *Power System Stability and Control*, chapter 17. McGraw-Hill Education, New York, 2nd edition, 2022.

[65] Malte Kuss. *Gaussian process models for robust regression, classification, and reinforcement learning*. PhD thesis, echnische Universität Darmstadt Darmstadt, Germany, 2006.

[66] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

[67] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Penning-

ton, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[68] Gengda Li, Zhenqing Duan, Ling Liang, Honglu Zhu, Aoyu Hu, Qingru Cui, Baowei Chen, and Wensen Hu. Outlier data mining method considering the output distribution characteristics for photovoltaic arrays and its application. *Energy Reports*, 6:2345–2357, 11 2020. ISSN 2352-4847. doi: 10.1016/J.EGYR.2020.08.034.

[69] Robert G. Littlejohn. Lecture notes on Physics 221A, Appendix C: Gaussian Integrals, 2021. https://bohr.physics.berkeley.edu/classes/221/notes/gaussint.pdf.

[70] Zhizhao Liu, Ming Yang, and Wei Li. A sequential latin hypercube sampling method for metamodeling. In *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems: 16th Asia Simulation Conference and SCS Autumn Simulation Multi-Conference, AsiaSim/SCS AutumnSim 2016, Beijing, China, October 8-11, 2016, Proceedings, Part I 16*, pages 176–185. Springer, 2016.

[71] Dimitrios Loukrezis and Herbert De Gersem. Adaptive sparse polynomial chaos expansions via leja interpolation. *arXiv preprint arXiv:1911.08312*, 2019.

[72] Zhanyu Ma and Arne Leijon. Bayesian estimation of beta mixture models with variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 (11):2160–2173, 2011.

[73] David JC MacKay. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 354(1):73–80, 1995.

[74] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gor-

don Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32, 2019.

[75] Giorgos Mamakoukas, Maria Castano, Xiaobo Tan, and Todd Murphey. Local koopman operators for data-driven control of robotic systems. In *Robotics: science and systems*, 2019.

[76] Kaisey Mandel and Eric Agol. Analytic light curves for planetary transit searches. *The Astrophysical Journal*, 580(2):L171, 2002.

[77] V. A. Marčenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967.

[78] Ricardo A Maronna and Victor J Yohai. The behavior of the stahel-donoho robust multivariate estimator. *Journal of the American Statistical Association*, 90(429):330–341, 1995.

[79] Ricardo A Maronna and Victor J Yohai. The behavior of the stahel-donoho robust multivariate estimator. *Journal of the American Statistical Association*, 90(429):330–341, 1995.

[80] Alexandre Mauroy, Igor Mezić, and Yoshihiko Susuki (Editors). *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*. Springer Nature Switzerland AG, Cham, Switzerland, 2020.

[81] Michael D McKay. Latin hypercube sampling as a tool in uncertainty analysis of computer models. In *Proceedings of the 24th conference on Winter simulation*, pages 557–564, 1992.

[82] Hendrik Alexander Mehrtens, Camila González, and Anirban Mukhopadhyay. Improv-

ing robustness and calibration in ensembles with diversity regularization. In *DAGM German Conference on Pattern Recognition*, pages 36–50. Springer, 2022.

[83] Lamine Mili, MG Cheniae, NS Vichare, and Peter J Rousseeuw. Robust state estimation based on projection statistics [of power systems]. *IEEE Transactions on Power Systems*, 11(2):1118–1127, 1996.

[84] L. Mill, M. G. Cheniae, and N. S. Vichare. Robust state estimation based on projection statistics. *IEEE Transactions on Power Systems*, 11(2):1118–1127, 1996. doi: 10.1109/59.496203.

[85] Adam Moss. Accelerated bayesian inference using deep learning. *Monthly Notices of the Royal Astronomical Society*, 496(1):328–338, 2020.

[86] Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002. URL https://ibug.doc.ic.ac.uk/media/uploads/documents/courses/DBN-PhDthesis-LongTutorial-Murphy.pdf. Ph.D. dissertation.

[87] Radford M Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*, 1997.

[88] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[89] Marcos Netto, Yoshihiko Susuki, and Lamine Mili. Data-driven participation factors for nonlinear systems based on Koopman mode decomposition. *IEEE Control Syst. Lett.*, 3(1):198–203, 2019.

[90] Feliks Nüske, Sebastian Peitz, Friedrich Philipp, Manuel Schaller, and Karl Worth-

mann. Finite-data error bounds for Koopman-based prediction and control. *J. Nonlinear Sci.*, 33(1):14, 2023.

[91] Judea Pearl. From bayesian networks to causal networks. In *Mathematical models for handling partial knowledge in artificial intelligence*, pages 157–182. Springer, 1995.

[92] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11): 559–572, 1901.

[93] Jon Perez-Cerrolaza, Jaume Abella, Markus Borg, Carlo Donzella, Jesús Cerquides, Francisco J Cazorla, Cristofer Englund, Markus Tauber, George Nikolakopoulos, and Jose Luis Flores. Artificial intelligence for safety-critical systems in industrial and transportation domains: A survey. *ACM Computing Surveys*, 56(7):1–40, 2024.

[94] Sascha Ranftl, Wolfgang von der Linden, and MaxEnt 2021 Scientific Committee. Bayesian surrogate analysis and uncertainty propagation. In *Physical Sciences Forum*, volume 3, page 6. MDPI, 2021.

[95] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International conference on machine learning*, pages 324–333. PMLR, 2016.

[96] CE. Rasmussen and CKI. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, January 2006.

[97] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages

1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

[98] Clarence W. Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S. Henningson. Spectral analysis of nonlinear flows. *J. Fluid Mech.*, 641:115–127, 2009.

[99] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.

[100] Christopher J Roy and William L Oberkampf. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Computer methods in applied mechanics and engineering*, 200(25-28):2131–2144, 2011.

[101] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5): 206–215, 2019.

[102] Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 71(2):319–392, 2009.

[103] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, November 1989. doi: 10.1214/ss/1177012413.

[104] Yusuf Sale, Viktor Bengs, Michele Caprio, and Eyke Hüllermeier. Second-order uncertainty quantification: A distance-based approach. In *Forty-first International Conference on Machine Learning*, 2023.

[105] Shankar Sankararaman and Sankaran Mahadevan. Integration of model verification, validation, and calibration for uncertainty quantification in engineering systems. *Reliability Engineering & System Safety*, 138:194–209, 2015.

[106] Takamitsu Sawa. Finite-sample properties of the k-class estimators. *Econometrica: Journal of the Econometric Society*, pages 653–680, 1972.

[107] Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.*, 656:5–28, 2010.

[108] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[109] Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

[110] Ransalu Senanayake. The role of predictive uncertainty and diversity in embodied ai and robot learning. *arXiv preprint arXiv:2405.03164*, 2024.

[111] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[112] Pranav Sharma, Venkataramana Ajjarapu, and Umesh Vaidya. Data-driven identification of nonlinear power system dynamics using output-only measurements. *IEEE Trans. Power Syst.*, 37(5):3458–3468, 2021.

[113] P. P. Shenoy and J. C. West. Inference in hybrid bayesian networks using mixtures of polynomials. *Int. J. Approximate Reasoning*, 52(5):641–657, 2011.

[114] Jack Sherman and Winifred J Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Stat.*, 21(1):124–127, 1950.

[115] David C Slaughter, DK Giles, and Daniel Downey. Autonomous robotic weed control systems: A review. *Computers and electronics in agriculture*, 61(1):63–78, 2008.

[116] AFM Smith, AM Skene, JEH Shaw, JC Naylor, and M Dransfield. The implementation of the bayesian paradigm. *Communications in Statistics-Theory and Methods*, 14(5): 1079–1102, 1985.

[117] AFM Smith, AM Skene, JEH Shaw, and JC Naylor. Progress with numerical and graphical methods for practical bayesian statistics. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 36(2-3):75–82, 1987.

[118] Werner Stahel, Sanford Weisberg, Peter J Rousseeuw, and Bert C van Zomeren. Robust distances: simulations and cutoff values. In *Directions in Robust Statistics and Diagnostics: Part II*, pages 195–203. Springer, 1991.

[119] Werner A Stahel. *Robuste schätzungen: infinitesimale optimalität und schätzungen von kovarianzmatrizen*. PhD thesis, ETH Zurich, 1981.

[120] Ralf C Staudemeyer and Eric Rothstein Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.

[121] Hongsheng Su, Xiaoyang Dong, and Xiaoying Yu. Probabilistic load flow analysis based on sparse polynomial chaotic expansion. *Journal of Electrical Engineering & Technology*, 15:527–538, 2020.

[122] Bruno Sudret, Stefano Marelli, and Joe Wiart. Surrogate models for uncertainty quantification: An overview. In *2017 11th European conference on antennas and propagation (EUCAP)*, pages 793–797. IEEE, 2017.

[123] Timothy John Sullivan. *Introduction to uncertainty quantification*, volume 63. Springer, 2015.

[124] Qiang Sun, Wen-Xin Zhou, and Jianqing Fan. Adaptive huber regression. *Journal of the American Statistical Association*, 115(529):254–265, 2020.

[125] Yoshihiko Susuki and Igor Mezić. Nonlinear Koopman modes and coherency identification of coupled swing dynamics. *IEEE Trans. Power Syst.*, 26(4):1894–1904, 2011.

[126] Yoshihiko Susuki and Igor Mezić. Nonlinear Koopman modes and power system stability assessment without models. *IEEE Trans. Power Syst.*, 29(2):899–907, 2014.

[127] Yoshihiko Susuki, Igor Mezic, Fredrik Raak, and Takashi Hikihara. Applied koopman operator theory for power systems technology. *Nonlinear Theory and Its Applications, IEICE*, 7(4):430–459, 2016.

[128] Jalil Taghia and Arne Leijon. Variational inference for watson mixture model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1886–1900, 2015.

[129] Jalil Taghia, Zhanyu Ma, and Arne Leijon. Bayesian estimation of the von-mises fisher mixture model with variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 36(9):1701–1715, 2014.

[130] Robert J Tibshirani and Bradley Efron. An introduction to the bootstrap. *Monographs on statistics and applied probability*, 57(1):1–436, 1993.

[131] Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, 81(393):82–86, 1986.

[132] Huynh TT Tran, Hieu T Nguyen, Long T Vu, and Samuel T Ojetola. Solving differential-algebraic equations in power system dynamic analysis with quantum computing. *Energy Conversion and Economics*, 5(1):40–53, 2024.

[133] Rohit K Tripathy and Ilias Bilionis. Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of computational physics*, 375:565–588, 2018.

[134] Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.

[135] Joachim van der Herten, Ivo Couckuyt, Dirk Deschrijver, and Tom Dhaene. A fuzzy hybrid sequential design strategy for global surrogate modeling of high-dimensional computer experiments. *SIAM Journal on Scientific Computing*, 37(2):A1020–A1039, 2015.

[136] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[137] N. X. Vinh, M. Chetty, R. Coppel, and P. P. Wangikar. Globalmit: Learning globally optimal dynamic bayesian network with the mutual information test criterion. *Bioinformatics*, 27(19):2765–2766, 2011.

[138] Markus Walker, Marcel Reith-Braun, Peter Schichtel, Mirko Knaak, and Uwe D Hanebeck. Identifying trust regions of bayesian neural networks. In *2023 IEEE Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI)*, pages 1–8. IEEE, 2023.

[139] Xiaoliang Wan and George Em Karniadakis. An adaptive multi-element generalized

polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2):617–642, 2005.

[140] Hao Wang and Dit-Yan Yeung. A survey on bayesian deep learning. *ACM computing surveys (csur)*, 53(5):1–37, 2020.

[141] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4): 897–936, 1938.

[142] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[143] Matthew O Williams, Clarence W Rowley, and Ioannis G Kevrekidis. A kernel-based approach to data-driven koopman spectral analysis. *arXiv preprint arXiv:1411.2260*, 2014.

[144] Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *J. Nonlinear Sci.*, 25(6):1307–1346, 2015.

[145] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015.

[146] Winnie Xu, Ricky TQ Chen, Xuechen Li, and David Duvenaud. Infinitely deep bayesian neural networks with stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 721–738. PMLR, 2022.

[147] Yijun Xu, Zhixiong Hu, Lamine Mili, Mert Korkali, and Xiao Chen. Probabilistic power flow based on a gaussian process emulator. *IEEE Transactions on Power Systems*, 35 (4):3278–3281, 2020.

[148] Xi Ye, Zongxiang Lu, Ying Qiao, Yong Min, and Mark O'Malley. Identification and correction of outliers in wind farm time series power data. *IEEE Transactions on Power Systems*, 31(6):4197–4205, 11 2016. doi: 10.1109/TPWRS.2015.2512843.

[149] Zhi Yuan, Nima Khakzad, Faisal Khan, and Paul Amyotte. Risk analysis of dust explosion scenarios using bayesian networks. *Risk analysis*, 35(2):278–291, 2015.

[150] Christophe Zhang and Enrique Zuazua. A quantitative analysis of Koopman operator methods for system identification and predictions. *Comptes Rendus. Mécanique*, 351 (S1):1–31, 2023.

[151] Jiaxin Zhang. Modern monte carlo methods for efficient uncertainty quantification and propagation: A survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 13 (5):e1539, 2021.

[152] Yinhao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394: 56–81, 2019.

[153] J. D. Ziebarth, A. Bhattacharya, and Y. Cui. Bayesian network webserver: A comprehensive tool for biological network modeling. *Bioinformatics*, 29(21):2801–2803, 2013.

Appendices

Appendix A

Gaussian process

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. A Gaussian process is completely specified by its mean function and covariance function. The mean function $m(x)$ and covariance function $k(x, x')$ of a real process $f(x)$ are defined as:

$$m(x) = \mathbb{E}[f(x)], \quad k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))].$$

We write the Gaussian process as:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')).$$

The covariance function specifies a distribution over functions. By choosing a set of input points X_*, we can compute the covariance matrix $K(X_*, X_*)$ elementwise and sample a random Gaussian vector from the distribution:

$$\mathbf{f}_* \sim \mathcal{N}(0, K(X_*, X_*)). \tag{1}$$

In realistic scenarios, we do not observe the function values directly but rather noisy observations, modeled as $y = f(x) + \varepsilon$, where ε is Gaussian noise with variance σ_n^2. The covariance

of the noisy observations is:

$$\text{cov}(y_p, y_q) = k(x_p, x_q) + \sigma_n^2 \delta_{pq}, \quad \text{or equivalently,} \quad \text{cov}(\mathbf{y}) = K(X, X) + \sigma_n^2 I.$$

Here, δ_{pq} is the Kronecker delta, which is 1 if $p = q$ and 0 otherwise. Adding the noise term modifies the covariance structure, as seen in the diagonal addition to the covariance matrix.

The joint distribution of the observed targets $\mathbf{y}$ and the function values $\mathbf{f}_*$ at test points X_* is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right). \tag{2}$$

Using this joint distribution, we can derive the conditional distribution for Gaussian process regression. The posterior distribution of the function values $\mathbf{f}_*$ at the test points, given observations $\mathbf{y}$ and inputs X, is:

$$\mathbf{f}_* \mid X, \mathbf{y}, X_* \sim \mathcal{N}(\mathbb{E}[\mathbf{f}_*], \text{cov}(\mathbf{f}_*)),$$

where the predictive mean and covariance are given by:

$$\mathbb{E}[\mathbf{f}_* \mid X, \mathbf{y}, X_*] = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}\mathbf{y}, \tag{3}$$

$$\text{cov}(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}K(X, X_*). \tag{4}$$

These equations form the foundation of Gaussian process regression, enabling us to make predictions with uncertainty quantification. The mean provides the expected prediction, while the covariance quantifies the uncertainty around those predictions.

Appendix B

GP-Huber additional experiments

B.1 Selection of the threshold b

The Huber estimator is a maximum likelihood estimator associated with the least favorable density function given by

$$\tilde{g}(r) = \frac{1-\varepsilon}{\sqrt{2\pi}\sigma} e^{-\rho\left(\frac{r}{\sigma}\right)}, \tag{B.1}$$

which can be further elaborated as

$$\tilde{g}(r) = \begin{cases} \frac{1-\varepsilon}{\sqrt{2\pi}} e^{-\frac{r^2}{2}} & \text{for } |r| \leq b \\ \frac{1-\varepsilon}{\sqrt{2\pi}} e^{-|b||r|-\frac{b^2}{2}} & \text{for } |r| > b \end{cases} \tag{B.2}$$

This distribution is Gaussian in the center and Laplacian in the tails. The threshold b is related to the fraction of contamination ε against which we want to be protected. This relation is obtained by setting

$$\int_{-\infty}^{\infty} \tilde{g}(r)\, dr = 1 \tag{B.3}$$

yielding

$$\int_{-b}^{b} \frac{(1-\varepsilon)}{\sqrt{2\pi}} e^{-\frac{r^2}{2}}\, dr + 2(1-\varepsilon) \int_{b}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\left(-br+\frac{b^2}{2}\right)}\, dr = 1 \tag{B.4}$$

$$(1-\varepsilon) \int_{-b}^{b} \frac{1}{\sqrt{2\pi}} e^{-\frac{r^2}{2}}\, dr = (1-\varepsilon)[1-2(1-\Phi(b))] = (1-\varepsilon)(2\Phi(b)-1); \tag{B.5}$$

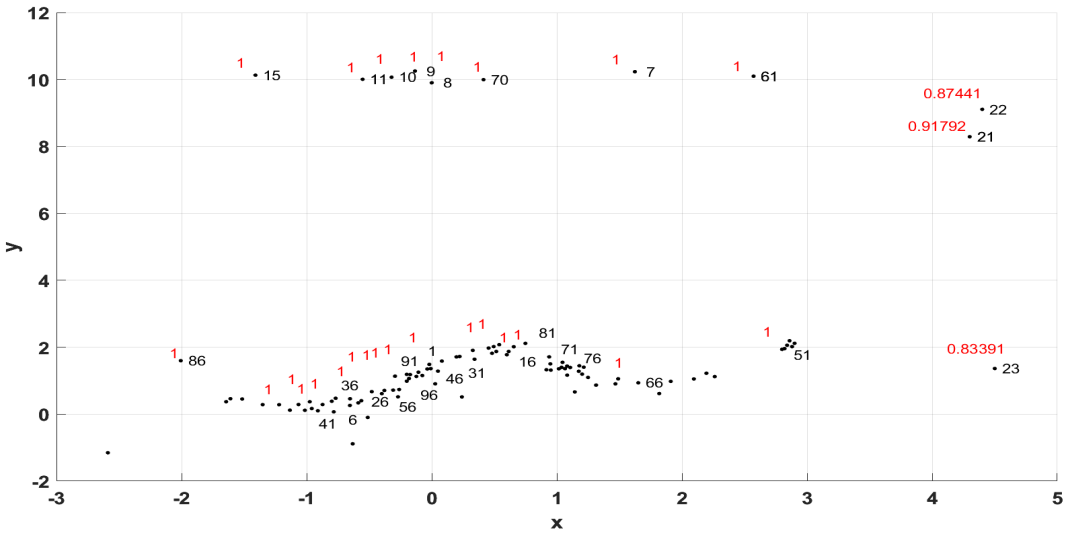


Figure B.1: Weights based on PS for the Neal data. The numbers right to the data points indicate index numbers and the ones to the left in red color indicate the weights associated with that data point.

and

$$2(1-\varepsilon)\int_b^\infty \frac{1}{\sqrt{2\pi}} e^{\left(-br+\frac{b^2}{2}\right)}\, dr = 2(1-\varepsilon)\frac{1}{\sqrt{2\pi}}\left[-\frac{1}{b}e^{\left(-br+\frac{b^2}{2}\right)}\right]_b^\infty \tag{B.6}$$

$$= \frac{2(1-\varepsilon)}{\sqrt{2\pi}}\frac{1}{b}e^{-\frac{b^2}{2}} = \frac{2(1-\varepsilon)}{b}\phi(b); \tag{B.7}$$

Solving further, we get

$$2\Phi(b) - 1 + \frac{2}{b}\Phi(b) = \frac{1}{1-\varepsilon} \tag{B.8}$$

We observe that b decreases to 0 as ε increases to 1. At $b = 1.5$, the Huber loss can handle roughly $\varepsilon = 0.1$ i.e. 10% of outliers.

B.2 Additional experiments

B.2.1 Neal dataset

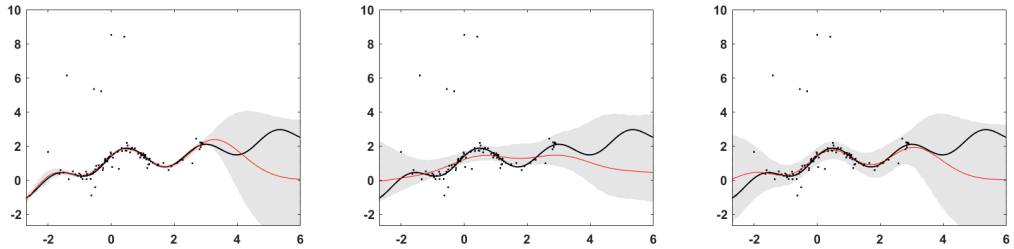
[87] proposed the following artificial model:

$$g(\boldsymbol{x}_i) = 0.3 + 0.4x + 0.5sin(2.7x) + 1.1/(1 + x^2). \tag{B.9}$$

A sample of $n = 100$ points constitutes the training data set, $(\mathbf{X}, \mathbf{y})$. The predictions of the vector function, $\mathbf{f}^*$, are made at $n^* = 541$ test covariates contained in $\boldsymbol{x}^*$ over the interval $[-2.7, 5]$. Since the projection statistics require at least a two-dimensional covariate space, they are calculated on the regressors' vector, $\mathbf{x}$ combined with the column of ones, i.e., on the matrix $\mathbf{H} = [\mathbf{1}, \mathbf{x}]$. Specifically, for a test point $\mathbf{h}_i = [1, x_i]$, $\text{PS}(\mathbf{h}_i)$ is calculated using (4.5). The training covariate, x_i, is flagged as an outlier if the associated projection pursuit weights, $w_i = \min\left(1, \frac{c}{\text{PS}(\mathbf{h}_i)^2}\right)$, has a value less than one.

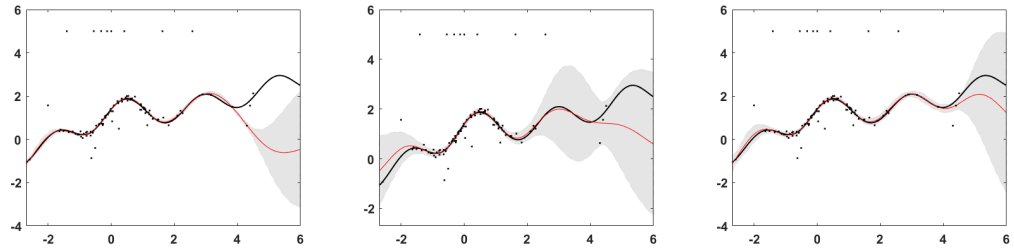
We demonstrate the proposed GP-Huber in four cases of error probability distribution: (i) $\mathcal{N}(0.01, 0.08)$; (ii) the Student's t-distribution with 10 degrees of freedom; (iii) Laplace$(0, 0.1)$; and (iv) the Cauchy distribution. For each of these error distributions, we introduce extreme output outliers $y^l = \{90.5, 8.6, 98.1, 5.3, 5.2, 6.1, 1, 8\}$ at locations $j = \{7, 8, 9, 10, 11, 15, 61, 70\}$, extreme covariate data points $x^{(l)} = \{4.3, 4.4, 4.5\}$ at locations $i = \{21, 22, 23\}$. We also add large magnitudes to introduce group of good data points to the covariates $\{x_{50}, x_{51}, x_{52}, x_{53}, x_{54}, x_{55}\}$ for which $y_i = g(\boldsymbol{x}_i)$.

We observe that the projection pursuit weights based on the PS corresponding to the bad leverage points are $\{0.9179, 0.8744, 0.8339\}$ while those corresponding to the good leverage points are equal to 1 (see Figure B.1).



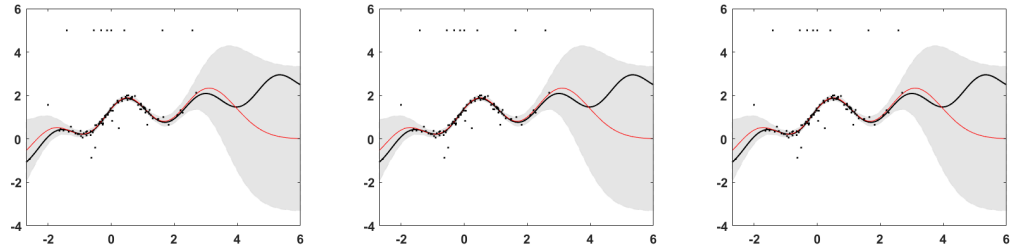
(a) (b) (c)

Figure B.2: Predicted values for (a) tLA; (b) HuberMCMC; (c) HuberLA with standard deviations for the Case 2 with error following Student's t distribution on Neal dataset.



(a) (b) (c)

Figure B.3: Predicted values for (a) tLA; (b) HuberMCMC; (c) HuberLA with standard deviations for the Case 3 with error following Student's t distribution on Neal dataset.



(a) (b) (c)

Figure B.4: Predicted values for (a) tLA; (b) HuberMCMC; (c) HuberLA with standard deviations for the Case 4 with error following Student's t distribution on Neal dataset.

	SCtMCMC	tLA	HuberMCMC	HuberLA	RCGP	GP	LaplaceMCMC
			$\varepsilon \sim \mathcal{N}(0.01, 0.08)$				
RMSE	1.41	1.30	1.40	1.36	2.04	1.74	1.48
MAE	0.90	0.81	0.99	0.95	1.93	1.51	0.98
			$\varepsilon \sim$ Student-t(10)				
RMSE	1.22	1.14	0.91	1.12	2.04	1.66	1.01
MAE	0.63	0.56	0.62	0.67	1.85	1.34	0.92
			$\varepsilon \sim$ Laplace$(0, 0.1)$				
RMSE	1.38	2.73	1.33	1.37	2.06	1.73	1.33
MAE	0.88	1.82	0.97	0.96	1.95	1.51	0.95
			$\varepsilon \sim$ Student-t(1) (Cauchy)				
RMSE	4.74	2.11	1.33	1.38	2.11	1.75	1.33
MAE	1.67	1.36	0.96	0.98	1.84	1.50	0.95

Table B.1: Results for Case 2

	SCtMCMC	tLA	HuberMCMC	HuberLA	RCGP	GP	LaplaceMCMC
			$\varepsilon \sim \mathcal{N}(0.01, 0.08)$				
RMSE	1.02	1.01	1.48	1.50	1.10	1.17	0.98
MAE	0.51	0.52	0.79	0.54	0.76	0.78	0.53
			$\varepsilon \sim$ Student-t(10)				
RMSE	1.58	1.02	1.17	1.13	1.11	1.17	0.61
MAE	1.28	0.52	0.53	0.78	0.76	0.85	0.35
			$\varepsilon \sim$ Laplace$(0, 0.1)$				
RMSE	1.04	1.01	1.06	1.18	1.16	1.08	1.16
MAE	0.51	0.52	0.53	0.78	0.66	0.66	0.58
			$\varepsilon \sim$ Student-t(1) (Cauchy)				
RMSE	1.58	1.02	1.18	1.02	1.10	1.07	1.04
MAE	1.28	0.52	0.63	0.78	0.56	0.56	0.52

Table B.2: Neal results for the Case 4.

B.2.2 Transmission spectroscopy

Transmission spectroscopy records the relative change in the stellar flux, which is the incident photons per unit area, as a planet travels in front of the star around which it revolves. When

the planet faces the star directly, known as a transit, it occludes a fraction of the stellar flux emitted by the star equal to the sky-projected area of the planet as compared to the area of the star, which is referred to as transit depth. The measurement of the total flux over time is known as the light curve. The property on which the transmission spectroscopy relies to estimate the transit curve parameters is the planet's transit depth, which dependents on the wavelengths of the transmitted flux. For the wavelengths where the planet's atmosphere is opaque due to the absorption of the emitted electromagnetic waves by constituent atoms or molecules, the planet blocks slightly more stellar flux. The variations are measured by binning the light curve into spectrophotometric channels of different wavelengths and by fitting the light curve from each channel separately with a transit model [63].

The sources of error, such as photon noise and instrumental and astrophysical systematics, raise many potential challenges for precise atmosphere characterization. Pointing drift or modifications in the telescope focus influence the spectrum position on the detector to a small degree during transit due to instrumental systematics. Note that instrumental systematics are nothing but what is popularly known as systematic errors in statistics, which are here attributed to the atmospheric effects on the physical properties of an instrument. The optical state parameters are metered via auxiliary measurements of the spectral trace such as position, width, angle, or other parameters, indicating the state of the detector and optics, which are thought to be the cause of instrumental systematics. Instead of modeling the latter as a linear function of the optical state parameters, [43] proposed a non-parametric model by leveraging GPs.

The observation set obtained from HST- NICMOS includes the light curves for 18 wavelength channels extracted from n=638 spectra along with six optical state parameters, namely the position of the spectral trace along the dispersion axis ,ΔX, the average position of the spectral trace along the cross-dispersion axis, ΔY, the angle of the spectral trace with

the x-axis, W, the average width of the spectral trace, ψ^s, the temperature, T, and the orbital phase, ψ^H. The flux measurements contained in the vector, $\boldsymbol{f} = [f_1, f_2, \ldots, f_n]^T$, are recorded at n time instants, $\{t_1, t_2, \ldots, t_n\}$, contained in the time vector, $\boldsymbol{t}$, and the optical state parameters are given by $\mathbf{x}_i = [\Delta X_i, \Delta Y_i, W_i, \psi_i^H, T_i, \psi_i^s]^T$ at time instant, t_i, collected in the matrix $\mathbf{X} \in \mathbb{R}^{6\times N}$ given by $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$.

The observed transit flux modeled in the GP framework follows a normal distribution, that is,

$$\boldsymbol{f}(\boldsymbol{t}, \mathbf{X}) \sim \mathcal{N}(\boldsymbol{T}(\boldsymbol{t}, \boldsymbol{\phi}), \mathbf{K}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta})). \tag{B.10}$$

where the parameter vector, $\boldsymbol{\phi}$, include the parameter of interest, ρ_{radius}, and other parameters, namely out-of-transit flux, f_{oot}, time gradient, T_{grad}, fixed central transit time, T_0, orbital period, P, limb darkening coefficient, c_1, limb darkening coefficient, c_2. The transit vector function, $\boldsymbol{T}(\boldsymbol{t}, \boldsymbol{\phi})$, is hereafter referred to as mean function parameter vector. The non-variable mean function parameters are fixed or calculated as stated in [43]. Along with the planet-to-star radius ratio, the other mean function parameters are the parameters of a linear baseline model, f_{oot} and T_{grad}. The covariance matrix, $\boldsymbol{\Sigma}(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta})$, is the covariance between two output flux measurements defined as a function of the distance between optical state parameters, $(\mathbf{x}_i, \mathbf{x}_j)$, given by

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij}\sigma^2, \tag{B.11}$$

where $k(\cdot, \cdot)$ is a Gaussian kernel. The threshold parameter, b, is set to 1.5 to achieve good robustness and efficiency at data distributed normally.

The joint un-normalized log-posterior function of $\boldsymbol{\phi}$, $\boldsymbol{\beta}$, and $\boldsymbol{\theta}$ is given by

$$\log P(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2, \boldsymbol{\beta} | \boldsymbol{f}, \mathbf{X}, \boldsymbol{\zeta}) = \log \left(\mathcal{L}(\mathbf{r} | \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2) \right) - \frac{\tau}{l_\tau} - \sum_{i=1}^{d} \left(\frac{1}{s_i l_i} \right) + \log(\boldsymbol{\beta}) - \boldsymbol{\beta}^T \boldsymbol{\sigma}^2 + \log(p(\boldsymbol{\beta} | \boldsymbol{\zeta})) + \mathrm{C}. \quad \text{(B.12)}$$

Here, we lay the gamma a priori probability density function, $p(\boldsymbol{\theta}) = \frac{1}{\boldsymbol{l}} \exp\left(\frac{-\boldsymbol{\theta}}{\boldsymbol{l}}\right)$ over the covariance function hyperparameters $\boldsymbol{\theta}$. The parameter l_τ is of the gamma a priori associated with hyperparameter τ and C represents additional constant terms. The samples of β_l are generated from log uniform distribution to lay a non-informative prior with parameter vector, $\boldsymbol{\zeta}$, whereas $p(\beta_g)$ is a degenerate probability density function.

The values of the planet-to-star radius ratio ρ_{radius} for each wavelength obtained from the GP-Huber model are shown in Table B.3 along with those obtained from the model described in [43] referred to as Gibson2012, where $\Delta\rho_{radius}$ represents the estimated uncertainty.

Table B.3: Results of the planet-to-star radius ratio obtained from Gibson (2012) and GP-Huber.

Wavelength	Results from model in Gibson2012		Results obtained from GP-Huber	
(μm)	ρ_{radius}	$\Delta\rho_{radius}$	ρ_{radius}	$\Delta\rho_{radius}$
2.468	0.15545	0.00077	0.15525	0.00071
2.411	0.15520	0.00052	0.15771	0.0008911
2.353	0.15455	0.00044	0.15488	0.0004021
2.296	0.15513	0.00057	0.15825	0.0006526
2.238	0.15512	0.00041	0.1542	0.0005276
2.181	0.15504	0.00051	0.15297	0.0007462
2.124	0.15417	0.00066	0.15928	0.0007869
2.066	0.15508	0.00066	0.15525	0.000399
2.009	0.15393	0.00036	0.15259	0.0004077
1.951	0.15595	0.00051	0.15602	0.0005586
1.894	0.15549	0.0006	0.15466	0.0005988
1.837	0.15513	0.00053	0.15433	0.0004704
1.779	0.15534	0.00051	0.1537	0.0003601
1.722	0.15447	0.00087	0.14937	0.0006938
1.665	0.15429	0.00064	0.1517	0.000871
1.607	0.15266	0.00062	0.15213	0.0008045
1.55	0.15359	0.00073	0.15276	0.0007583
1.492	0.15367	0.00118	0.15256	0.0010653