Data-Driven Polynomial Chaos Expansions for Uncertainty Quantification

Zhanlin Liu

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

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Youngjun Choe, Chair

Ashis G. Banerjee

Shuai Huang

Christina Mastrangelo

Program Authorized to Offer Degree:
Department of Industrial & Systems Engineering

University of Washington

Abstract

Data-Driven Polynomial Chaos Expansions for Uncertainty Quantification

Zhanlin Liu

Chair of the Supervisory Committee:
Assistant Professor Youngjun Choe
Industrial & Systems Engineering

Uncertainties exist in both physics-based and data-driven models of systems. Understanding how system inputs affect a system output's uncertainty is essential to improve system outputs such as quality and productivity. Variance-based sensitivity analysis, which is widely used for uncertainty quantification, characterizes how the output variance is propagated from inputs. To estimate the variance-based sensitivity indices of the output with respect to inputs, polynomial chaos expansions (PCEs) are widely used. However, a majority of existing PCEs impose parametric distributional assumptions on inputs. Furthermore, existing sensitivity indices for dependent inputs impose strong assumptions on the dependence structure of the inputs or lack interpretability. Although recent studies proposed fully data-driven PCEs without strong assumptions on inputs, these PCEs are generally inefficient because the minimally required number of observations increases exponentially in the number of the inputs.

To address these challenges, three data-driven PCEs are proposed in this dissertation. We first propose the sparse network PCE (SN-PCE) model for a broad class of systems whose input-output relationships are expressed as directed acyclic graphs. The proposed SN-PCE model accurately estimates variance-based sensitivity indices with far fewer observations than state-of-the-art black-box methods. Next, we propose data-driven sensitivity indices by constructing ordered partitions of linearly independent polynomials of dependent

inputs for PCEs. The proposed sensitivity indices provide intuitive interpretations of how the dependent inputs affect the variance of the output without a priori knowledge of the dependence structure of the inputs. Finally, we propose a data-driven algorithm to build sparse PCEs for models with dependent inputs. The proposed algorithm not only reduces the number of minimally required observations but also improves upon the numerical stability and estimation accuracy of a state-of-the-art sparse PCE.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

DAG Direct acyclic graph.

GS-PCE Gram-Schmidt based polynomial chaos expansion.

GSA Global sensitivity analysis.

LAR Least angle regression.

mGS-PCE Modified Gram-Schmidt based polynomial chaos expansion.

PCE Polynomial chaos expansion.

SA Sensitivity analysis.

SN-PCE Sparse network polynomial chaos expansion.

UQ Uncertainty quantification.

ACKNOWLEDGMENTS

First, I want to express my deep and sincere appreciation to my advisor, Professor Youngjun Choe, for his endless patience and generous support to guide me through my entire Ph.D. study. His instructions not only strengthened my technical skills and writing ability but also helped me develop my career path.

I also want to express my special thanks to Professor Ashis G. Banerjee, who helped me improve both research skills and presentation abilities. It was also my great pleasure to work with him on Chapter 2 and have his great advice and suggestions.

Besides, I want to thank Professor Shuai Huang and Professor Christina Mastrangelo for taking the time to be reading committee members of my dissertation. I am also grateful that I had the opportunity to work as a research assistant for Professor Christina Mastrangelo on her interesting research project, which allowed me to learn more statistical knowledge besides my own research topic.

I want to thank all my friends who helped at UW including Dr. Jue Gong, Dr. Wei Guo, Dr. Yan Jin, Dr. Linpeng Wei, Tianshu Feng, James Starling, Qiang Meng, and all other people who helped me at UW.

Last but not least, I want to thank my mother for her selfless love in her financial support for my undergraduate study and master's study.

DEDICATION

to family

Chapter 1

INTRODUCTION

1.1 Motivation

Uncertainties exist in both physics-based and data-driven models of systems. Understanding how system inputs affect a system output's uncertainty is essential to improve system outputs such as quality and productivity. Uncertainty quantification (UQ) methods to characterize and reduce those uncertainties are increasingly popular in engineering studies. As an aspect of UQ, uncertainty propagation evaluating the low-order moments of the outputs, e.g. mean and variance, and estimating the probability distribution of the output allows us to predict the performance of the output considering uncertainties of the potential inputs. More importantly, sensitivity analysis (SA) characterizes and identifies how output uncertainties are propagated from input uncertainties. Two general ways of conducting SA are local sensitivity analysis (LSA) and global sensitivity analysis (GSA). LSA analyzes how a small perturbation near an input space value could influence the output. On the contrary, GSA investigates how the input variability influences the output variability over the entire input space. In recent studies, variance-based sensitivity analysis, as a form of GSA, is utilized to understand system uncertainties in various applications. For example, the Sobol index is one of the most widely used sensitivity indices for systems with independent inputs, which directly represents how the variance of the output is affected by the independent inputs. It has been used in various engineering applications including material mechanics [32], building energy [57], structure mechanics [33], hydrogeology [16], and manufacturing [19].

In practice, simple random sampling (from the actual process) or Monte Carlo sampling (from the simulation model of the process) is most commonly used to estimate the Sobol indices, where many observations of the inputs and output are available. A more resource-

efficient alternative is to construct a model (or, metamodel) of the actual process (or, its simulation model when the computational cost is expensive), and use the model (or, metamodel) to estimate the Sobol indices [63]. Among such models, *polynomial chaos expansion* (PCE) is particularly conducive to estimating the Sobol indices. In essence, the PCE expands a random variable (e.g., process output) in terms of orthonormal polynomials in other random variables (e.g., process inputs), and the expansion's coefficients directly yield convergent estimators of the Sobol indices thanks to the orthogonality of the polynomials in a Hilbert space. However, a majority of existing PCEs impose parametric distributional assumptions on inputs. Although the latest development of PCE models allows independent inputs to follow arbitrary distributions, these PCEs are generally inefficient because the minimally required number of observations increases exponentially in the number of the inputs. For example, consider a welding process that has several process variables whose relationships are depicted in Fig. 1.1. The process output of interest is the total minimum theoretical energy required for the welding process, E. This energy depends on the weld volume V, specific gravity ρ, heat capacity C_p, initial temperature T_i, final temperature T_f, and latent heat H. In turn, the weld volume V depends on six welding parameters (weld zone dimensions: e, g, h, l, and t; weld length L). In order to build a PCE model of E with respect to all the inputs, which are the inputs noted as blue nodes in Figure. 1.1, the existing PCE model requires at least $\binom{11+p}{p}$ random observations, where p is the pre-specified polynomial order for a PCE model and 11 is the number of the inputs. Therefore, it remains a challenge to efficiently obtain the sensitivity indices for a networked system which involves a large number of inputs.

In the meanwhile, recent research has extended the variance-based sensitivity analysis for systems with independent inputs to dependent inputs. Generalized Sobol sensitivity indices have been proposed in Chastaing et al. [12] based on the hierarchically orthogonal functional decomposition (HOFD). However, the unboundedness of the resulting sensitivity indices makes their interpretation for *dependent* inputs not as straightforward as the Sobol indices for models with *independent* inputs [48]. A different framework is proposed in [74] to

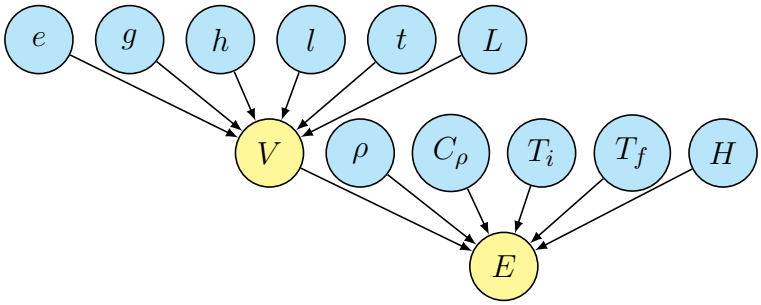


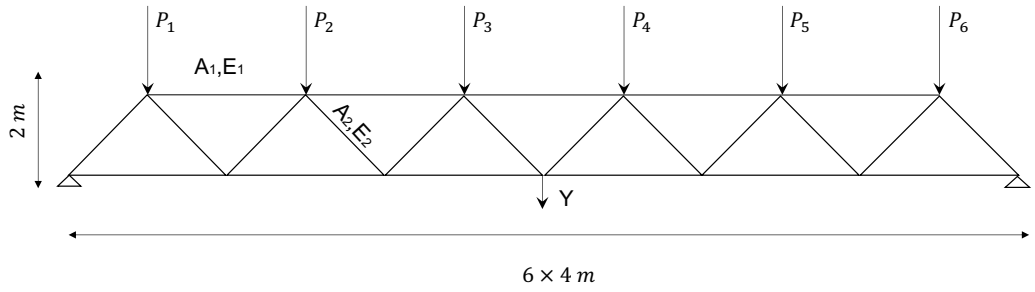
Figure 1.1: This network structure represents the relationships among the variables in the welding process [47]; the weld volume V depends on six welding parameters (weld zone dimensions: e, g, h, l, and t; weld length L), and the total energy E depends on V and additional parameters, ρ, C_p, T_i, T_f, and H.

obtain sensitivity indices for models with correlated inputs. However, it requires the knowledge of model structure between the inputs and the outputs. An alternative way of obtaining sensitivity indices for models with *dependent* inputs is to transform *dependent* inputs into *independent* inputs [40, 41, 65]. Even though the transformation-based methods generate interpretable sensitivity indices, they require strong assumptions on the dependency or distributions of the inputs. However, such assumptions might not be satisfied in a real-world application. For example, Figure. 1.2 shows an example of a truss model where the six loads fail to follow the assumptions proposed in [40, 41, 65]. Hence, defining interpretable sensitivity indices for systems with dependent inputs considering practical limitations requires further research.

1.2 Research objectives

The research objectives of this dissertation are as follows:

- Develop data-driven models to efficiently identify the inputs that significantly affect the variance of the output in a complex real-world system without using a large number of observations.



[h]

Figure 1.2: Scheme of the horizontal truss model modified from [37]. The downward vertical displacement at the mid span of the structure Y depends on Young modulus $E_i, i = 1, 2$, cross-sectional area $A_i, i = 1, 2$ for both horizontal and diagonal bars, and the random loads $P_i, i = 1, 2, \cdots, 6$.

- Develop a data-driven method to obtain interpretable variance-based sensitivity indices for systems with dependent inputs without any strong assumptions on the inputs.
- Develop a data-driven method to efficiently build sparse PCE models and accurately predict the output with dependent inputs.

1.3 Organization

This dissertation consists of multiple chapters, where Chapters 2 through 4 represent stand-alone research articles. Each chapter's notations are independent of the other chapters. This dissertation is organized as follows:

- **Chapter 2 Identifying the influential inputs for network output variance using sparse polynomial chaos expansion [1]:** In this chapter, we propose a sparse network polynomial chaos expansion model for a class of input-output relationships represented as directed acyclic networks. The model exploits the relationship structure by recursively relating a network node to its direct predecessors to trace the output

variance back to the inputs. It, thereby, estimates the Sobol indices, which measure the influence of each input on the output variance, accurately and efficiently. Theoretical analysis establishes the validity of the model as the prediction of the network output converges in probability to the true output under certain regularity conditions. Empirical evaluation on two manufacturing processes and a flooding process shows that the model estimates the Sobol indices accurately with far fewer observations than state-of-the-art black-box methods.

- **Chapter 3 Data-driven sensitivity indices for models with dependent inputs using the polynomial chaos expansion:** Data-driven sensitivity indices are proposed for models with dependent inputs in this chapter. We first construct ordered partitions of linearly independent polynomials of the inputs. The modified Gram-Schmidt algorithm is then applied to the ordered partitions to generate orthogonal polynomials with respect to the empirical measure based on observed data of model inputs and outputs. Using the polynomial chaos expansion with the orthogonal polynomials, we obtain the proposed data-driven sensitivity indices. The sensitivity indices provide intuitive interpretations of how the dependent inputs affect the variance of the output without a priori knowledge of the dependence structure of the inputs. Four numerical examples are used to validate the proposed approach.

- **Chapter 4 Data-driven sparse polynomial chaos expansion for models with dependent inputs:** In this chapter, we propose a data-driven approach to build sparse PCEs for models with dependent inputs. The proposed algorithm recursively constructs orthonormal polynomials using a set of initial polynomials based on their correlations with the output. The proposed algorithm on building sparse PCEs not only reduces the number of minimally required observations but also improves upon the numerical stability and estimation accuracy of a state-of-the-art sparse PCE. Four numerical examples are implemented to validate the proposed algorithm.

- **Chapter 5 Conclusions and future research:** We conclude the dissertation by

summarizing the major contributions and discussing future research directions.

Chapter 2

IDENTIFYING THE INFLUENTIAL INPUTS FOR NETWORK OUTPUT VARIANCE USING SPARSE POLYNOMIAL CHAOS EXPANSION

2.1 Abstract

Sensitivity analysis (SA) is an important aspect of process automation. It often aims to identify the process inputs that influence the process output's variance significantly. Existing SA approaches typically consider the input-output relationship as a black-box and conduct extensive random sampling from the actual process or its high-fidelity simulation model to identify the influential inputs. In this chapter, an alternate, novel approach is proposed using a sparse polynomial chaos expansion-based model for a class of input-output relationships represented as directed acyclic networks. The model exploits the relationship structure by recursively relating a network node to its direct predecessors to trace the output variance back to the inputs. It, thereby, estimates the Sobol indices, which measure the influence of each input on the output variance, accurately and efficiently. Theoretical analysis establishes the validity of the model as the prediction of the network output converges in probability to the true output under certain regularity conditions. Empirical evaluation on two manufacturing processes and a flooding process shows that the model estimates the Sobol indices accurately with far fewer observations than state-of-the-art black-box methods.

2.2 Introduction

Uncertainty quantification plays a critical role in controlling the uncertainties in process automation [18, 43]. Automated processes that neglect important uncertainties are, at minimum, unstable, and, in the worst case, have catastrophic process outcomes in terms of both

quality and productivity. To quantify how such uncertainties propagate through the process, sensitivity analysis is widely used in process automation. The sensitivity analysis of this study focuses on characterizing how the process output's variance is propagated from the inputs. This type of analysis is ubiquitous in different areas of automation science and engineering, such as thermodynamics [3], electromagnetism [13], power systems [49], building systems [38], and manufacturing [27].

How sensitive a random variable (e.g., process output) is with respect to another random variable (e.g., process input) is measured using a *sensitivity index*. Arguably, the most widely used sensitivity indices are the *Sobol indices* [59], which quantify how the independent inputs apportion the variance of the output. In practice, simple random sampling (from the actual process) or Monte Carlo sampling (from the simulation model of the process) is most commonly used to estimate the Sobol indices, where many observations of the inputs and output are available. A more resource-efficient alternative is to construct a model (or, metamodel) of the actual process (or, its simulation model when the computational cost is expensive), and use the model (or, metamodel) to estimate the Sobol indices [63].

Among such models, *polynomial chaos expansion* (PCE) is particularly conducive to estimating the Sobol indices, as detailed in Sec. 2.3. In essence, the PCE expands a random variable (e.g., process output) in terms of orthonormal polynomials in other random variables (e.g., process inputs), and the expansion's coefficients directly yield convergent estimators of the Sobol indices thanks to the orthogonality of the polynomials in a Hilbert space.

However, such models, including PCE, typically consider the input-output relationship of a process as a black-box for sensitivity analysis. This approach, while generally applicable to many processes, misses the opportunity to leverage the scientific/engineering knowledge of how the process actually works. Opening the black-box and utilizing the knowledge of the inner working can enable more effective modeling of the process, leading to more accurate and efficient sensitivity analysis.

To this end, this study considers a broad class of processes whose input-output relationships are expressed as *directed acyclic graphs* (DAGs), also known as directed acyclic

networks. Since network-structured processes are ubiquitous in practice, several real-world systems can potentially benefit from this study, such as biological networks [54], supply chain systems [45], manufacturing operations [23], and process industries, in general [2]. In particular, this study uses two manufacturing processes (welding and injection molding) and a flooding process in Sec. 2.5 for illustration purposes.

Consequently, the main contribution of this paper is in the development of a novel model, called sparse network PCE (SN-PCE), to accurately estimate the Sobol indices of the process output (sink node in the DAG) with respect to the process inputs (source nodes in the DAG). To the best of our knowledge, SN-PCE provides the most efficient way to estimate the Sobol indices for any process represented as a DAG. The estimated Sobol indices allow us to identify the inputs that significantly influence the output variance. The proposed model is validated through a theoretical analysis, which establishes that the prediction of the output converges in probability to the true output under certain regularity conditions (without imposing impractical restrictions such as parametric relationships among the network variables). The model is also empirically validated through sensitivity analysis of three real-world processes, where it is shown to accurately estimate the Sobol indices with much fewer observations of the process inputs and output as compared to state-of-the-art black-box methods.

The remainder of this paper is organized as follows. Sec. 2.3 briefly reviews the technical background on the PCE and Sobol indices. Sec. 2.4 starts with formal definitions regrading the DAG and presents the following three PCEs to model a process represented as a DAG: naïve PCE, network PCE, and SN-PCE. Sec. 2.4 also discusses the theoretical properties of the PCEs. In Sec. 2.5, the PCEs are empirically evaluated with two manufacturing applications and a flooding process. Sec. 2.6 concludes the paper with a discussion on future research directions.

2.3 Technical Background

In this section, we first formally define process inputs and output that bear uncertainties. Then, we present the PCE and sparse PCE. We review the construction of orthonormal

polynomials for the PCE, the Hoeffding decomposition, and the Sobol indices. We also review how to estimate the Sobol indices using the PCE.

2.3.1 Input random vector and output random variable

In this chapter, we generally use the same notations as [55], including $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. $\mathbb{A}^n \subseteq \mathbb{R}^n$, $n \in \mathbb{N}$, denotes a bounded or unbounded subdomain of $\mathbb{R}^n$. Let Ω be a sample space, and $\mathcal{F}$ be a σ-algebra on Ω. $(\Omega, \mathcal{F}, \mathcal{P})$ is a probability space with a probability measure $\mathcal{P} : \mathcal{F} \to [0,1]$. The Borel σ-algebra on $\mathbb{A}^n \subseteq \mathbb{R}^n$ is represented as $\mathcal{B}^n := \mathcal{B}(\mathbb{A}^n)$. An $\mathbb{A}^n$-valued input random vector $\boldsymbol{x} := (x_1, \ldots, x_n) : (\Omega, \mathcal{F}) \to (\mathbb{A}^n, \mathcal{B}^n)$ describes the uncertainties of n process inputs of interest.

To model a process output y using the PCE in $\boldsymbol{x}$, we assume that y is a function of $\boldsymbol{x}$ and is a square-integrable random variable defined on the same probability space $(\Omega, \mathcal{F}, \mathcal{P})$, written $y(\boldsymbol{x}) \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$.

2.3.2 Polynomial chaos expansion (PCE)

PCE approximates y using a finite number of orthonormal polynomials in $\boldsymbol{x}$ as follows:

$$\hat{y} = f(\boldsymbol{x}) \approx \sum_{i=0}^{P} \theta_i \psi_i(\boldsymbol{x}), \tag{2.1}$$

where θ_i and $\psi_i(\boldsymbol{x})$, $i = 0, 1, 2, \ldots, P$, denote the PCE coefficients and orthonormal polynomials in $\boldsymbol{x}$, respectively. $P+1$ is the number of the orthonormal polynomials and equals $\binom{n+p}{n}$ for the pre-specified highest polynomial order p and the dimension of $\boldsymbol{x}$, $\dim(\boldsymbol{x}) = n$. Thus, $P+1$ increases exponentially in n and p. As P increases, the approximation error in eq. (2.1) tends to zero [10].

In this chapter, we adopt a regression-based method to obtain the PCE coefficients by solving an overdetermined linear system of equations in the least-squares sense as follows [53]:

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^{P+1}}{\operatorname{argmin}} \sum_{j=1}^{m} \left(Y_j - \sum_{i=0}^{P} \theta_i \psi_i \left(\boldsymbol{X}_j \right) \right)^2, \tag{2.2}$$

where $\boldsymbol{\theta}$ denotes $(\theta_0, \theta_1, \ldots, \theta_P)$. Y_j and $\boldsymbol{X}_j$ represent the output and the input vector of the j^{th} observation, $j = 1, \ldots, m$, respectively.

Note that in contrast to typical regression models whose coefficients explain how the *expectation* of the output would conditionally change when the inputs are varied, the PCE coefficients are used to interpret how the *variance* of the output is apportioned to the inputs.

Thanks to the orthogonality of the orthonormal polynomials, the lower order moments of the output y are approximated using the PCE coefficients in eq. (2.1) as follows:

$$\begin{aligned} &\mathbb{E}(y) \approx \theta_0, \\ &Var(y) \approx \sum_{i=1}^{P} \theta_i^2, \end{aligned} \tag{2.3}$$

where $\mathbb{E}$ is the expectation operator with respect to the probability measure $\mathcal{P}$. The approximation errors in eq. (2.3) tend to zero as P in the PCE in eq. (2.1) increases.

2.3.3 Sparse PCE

The sparse PCE is extensively studied in the recent literature [8, 9, 14, 31, 52, 58]. In this chapter, we use l_1 minimization, specifically the least absolute shrinkage and selection operator (LASSO), to obtain a more parsimonious model than the model from eq. (2.2), by solving the following problem:

$$\begin{aligned} &\underset{\boldsymbol{\theta} \in \mathbb{R}^{P+1}}{\operatorname{argmin}} \sum_{i=0}^{P} |\theta_i|, \\ &\text{such that } \frac{\sum_{j=1}^{m} \left(Y_j - \sum_{i=0}^{P} \theta_i \psi_i \left(\boldsymbol{X}_j \right) \right)^2}{\sum_{j=1}^{m} \left(Y_j - \bar{Y} \right)^2} \leq \Gamma, \end{aligned} \tag{2.4}$$

where $\bar{Y} = \sum_{j=1}^{m} Y_j / m$. The parameter Γ constrains the goodness-of-fit for the sparse PCE (e.g., $\Gamma = 0$ requires a perfectly fitting model and $\Gamma = 1$ allows the model to be as simple as a constant model). The two application examples in Sec. 2.5 use Γ of 0.001.

2.3.4 Construction of multivariate orthonormal polynomials

A variety of PCEs have been proposed to construct orthonormal polynomials considering different types of input distributions. The Wiener chaos expansion, known as the first PCE, uses Hermite polynomials for independent Gaussian-distributed inputs [69]. Later PCEs allow for different input distributions and include the generalized PCE (gPCE) [72], the multi-element gPCE (ME-gPCE) [68], the moment-based arbitrary PCE (aPCE) [50], and the Gram-Schmidt based PCE (GS-PCE) [70]. In particular, GS-PCE, which accounts for dependent inputs following arbitrary distributions [48], provides the basis for constructing the proposed model in this chapter.

In this work, process inputs are assumed to be mutually independent. However, the other variables in the process (represented as a network) are allowed to be dependent on others. When the variables in $\boldsymbol{x}$ are *mutually independent*, the orthonormal polynomials are directly constructed as the tensor products of the univariate orthonormal polynomials as follows:

$$\psi_i(\boldsymbol{x}) = \psi_{\boldsymbol{\alpha}_i}(\boldsymbol{x}) := \prod_{j=1}^{n} \psi_{\alpha_{ij}}(x_j), \tag{2.5}$$

where $\boldsymbol{\alpha}_i := (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in})$. $\psi_{\alpha_{ij}}(x_j)$ represents the α_{ij}-th order orthonormal polynomial in input x_j. $\boldsymbol{\alpha}_i$ is the i-th arbitrary vector satisfying $|\boldsymbol{\alpha}_i| := \sum_{j=1}^{n} \alpha_{ij} \leq p$, where p is the highest order of the polynomials in the PCE. When variables are *dependent on each other*, we construct orthonormal polynomials using the modified Gram-Schmidt algorithm, as proposed in [39].

2.3.5 Hoeffding decomposition and Sobol indices

Suppose the inputs in the n-dimensional vector $\boldsymbol{x}$ are mutually independent and $y = f(\boldsymbol{x}) \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$. Denote the probability density function of $\boldsymbol{x}$ as $\mu(\boldsymbol{x})$. The Hoeffding decomposition of $f(\boldsymbol{x})$ is then defined as follows [12, 59]:

$$f(\boldsymbol{x}) := \sum_{u \subseteq \{1,2,\ldots,n\}} f_u(\boldsymbol{x}_u), \tag{2.6}$$

where $f_\emptyset := f_0$ is a constant and $\boldsymbol{x}_u := (x_j)_{j\in u}$ for $u \neq \emptyset$. Such decomposition is unique if and only if the summands in eq. (2.6) are orthogonal to each other as follows [59]:

$$\int f_u(\boldsymbol{x}) f_v(\boldsymbol{x}) \mu(\boldsymbol{x}) d\boldsymbol{x} = 0, \forall u, v \subseteq \{1, 2, \ldots, n\}, v \neq u. \tag{2.7}$$

Due to the orthogonal property in eq. (2.7), the functional decomposition of $Var(y)$ is expressed as follows [15]:

$$\begin{aligned} Var(y) &= \int f^2(\boldsymbol{x}) \mu(\boldsymbol{x}) d\boldsymbol{x} - f_0^2 \\ &= \sum_{\substack{u \subseteq \{1,2,\ldots,n\} \\ u \neq \emptyset}} D_u(y), \end{aligned}$$

where

$$\begin{aligned} D_u(y) &:= \int f_u^2(\boldsymbol{x}_u) \mu(\boldsymbol{x}_u) d\boldsymbol{x}_u \\ &= Var\left(\mathbb{E}\left(y|\boldsymbol{x}_u\right)\right) - \sum_{\substack{v \subset u \\ v \neq u \\ v \neq \emptyset}} D_v(y). \end{aligned}$$

Based on the decomposition, our sensitivity analysis considers the *first-order* Sobol index S_{x_j} and *total* Sobol index ST_{x_j} of y with respect to x_j defined as follows:

$$\begin{aligned} S_{x_j} &:= \frac{D_{\{j\}}(y)}{Var(y)}, \\ ST_{x_j} &:= \sum_{u \ni x_j} S_u, \end{aligned}$$

where $S_u := D_u(y)/Var(y)$. The first-order Sobol index S_{x_j} measures the *main effect* of input x_j on the output variance $Var(y)$. The total Sobol index ST_{x_j} measures the *total contribution* of x_j to $Var(y)$ including its main effect *and* interactions with other inputs [25].

2.3.6 PCE-based Sobol indices

The Sobol indices can be estimated directly using the PCE coefficients in eq. (2.1), making PCE particularly useful for the sensitivity analysis [61]. The first-order Sobol index is

estimated as follows:

$$S_{x_j} \approx \frac{\sum_{\boldsymbol{\alpha}_i \in \mathscr{A}_{\{j\}}} \theta^2_{\boldsymbol{\alpha}_i}}{\sum_{i=1}^{P} \theta_i^2}, \tag{2.8}$$

where $\theta_{\boldsymbol{\alpha}_i}$ is the PCE coefficient with respect to $\psi_{\boldsymbol{\alpha}_i}(\boldsymbol{x})$ in eq. (2.5) and

$$\mathscr{A}_u := \{\boldsymbol{\alpha}_i \in \mathbb{N}^n : \alpha_{ij} \neq 0 \leftrightarrow j \in u, |\boldsymbol{\alpha}_i| \leq p\}.$$

The total Sobol index is estimated as follows:

$$ST_{x_j} \approx \sum_{\mathscr{A}_{u \ni j}} S_{\mathscr{A}_u}, \tag{2.9}$$

where

$$S_{\mathscr{A}_u} \approx \frac{\sum_{\boldsymbol{\alpha}_i \in \mathscr{A}_u} \theta^2_{\boldsymbol{\alpha}_i}}{\sum_{i=1}^{P} \theta_i^2}.$$

2.4 Methodology

In this section, we first define notations on the directed acyclic graph (DAG), which represents the network-structured process of interest. Then, we present three models to estimate the Sobol indices for the process output with respect to the process inputs. The first model called naïve PCE is a baseline model and directly approximates the process output as a function of the process inputs, viewing the process as a black-box. The second model called network PCE uses the network structure of the process to effectively approximate the process output in terms of the process inputs. The third model called sparse network PCE (SN-PCE) imposes sparsity on the second model to use even fewer observations for the sensitivity analysis than the other models.

In addition, we show that predicted outputs from naïve PCE and network PCE converge to the true network output in probability under certain regularity conditions. This validates the use of the PCEs for estimating Sobol indices to conduct a sensitivity analysis. Because SN-PCE is a sparsity-imposed version of network PCE, its validity follows from the validity of network PCE and the sprase PCE.

2.4.1 Directed acyclic graph

Let $G = DAG(V, E)$ be the directed acyclic graph that represents the network-structured process of interest. $V = \{v_1, v_2, \ldots, v_{|V|}\}$ is the collection of all the nodes in G, where $|V|$ denotes the number of the nodes. Let x_{v_i} denote the random variable represented by the node $v_i \in V$ and, for $\boldsymbol{v} \subseteq V$, $\boldsymbol{x}_{\boldsymbol{v}} := (x_{v_i})_{v_i \in \boldsymbol{v}}$. $E \subseteq V \times V$ is the collection of all the directed edges in G, encoding all dependencies between the nodes. For example, $\langle v_i, v_j \rangle \in E$ implies that there is an edge from node v_i to node v_j, and hence x_{v_j} depends on x_{v_i}. The adjacency matrix $\boldsymbol{A}$ is defined as follows:

$$A_{ij} := \begin{cases} 1 & \langle v_i, v_j \rangle \in E \\ 0 & \langle v_i, v_j \rangle \notin E. \end{cases}$$

If $A_{ij} = 1$, then v_i is called a *direct predecessor* of v_j. If $\sum_{j=1}^{|V|} A_{ji} = 0$, v_i is called a *source node*. If $\sum_{j=1}^{|V|} A_{ij} = 0$, v_i is termed as a *sink node*. Let $S(G)$ denote all the source nodes in G. We call the variables in $\boldsymbol{x}_{S(G)}$ *network inputs* and the variables represented by the sink nodes *network outputs*. The node corresponding to the network output y is denoted by v_y.

We say that there exists a *path* from v_i to v_j if and only if either $A_{ij} = 1$ or there exists a sequence of nodes $(v_{k_1}, \ldots, v_{k_\tau})$ for $1 \leq \tau \leq |V| - 2$ such that

$$A_{ik_1} \prod_{t=1}^{\tau-1} A_{k_t k_{t+1}} A_{k_\tau j} = 1.$$

If there is such a path, we define $\mathcal{E}(v_i, v_j) = 1$; otherwise, $\mathcal{E}(v_i, v_j) = 0$. This function is useful for naïve PCE, which uses the network inputs that influence y, denoted as $\boldsymbol{\xi} := \boldsymbol{x}_{S(G) \cap V_y}$ for

$$V_y := \{v_i \in V : \mathcal{E}(v_i, v_y) = 1\}.$$

We assume the variables in $\boldsymbol{\xi}$ are mutually independent hereafter to well-define the Sobol indices of y with respect to $\boldsymbol{\xi}$.

For network PCE, which relates each node to its direct predecessors, we define

$$\mathscr{P}(\boldsymbol{x}_{\boldsymbol{v}}) := \boldsymbol{x}_{\left\{v_i \in V : v_j \in \boldsymbol{v}, A_{ij} = 1\right\} \cup (\boldsymbol{v} \cap S(G))},$$

where $\{v_i \in V : v_j \in \boldsymbol{v}, A_{ij} = 1\}$ represents the direct predecessors, if any, of the nodes in $\boldsymbol{v}$. To well-define $\mathscr{P}(\boldsymbol{x}_{\boldsymbol{v}})$, $\boldsymbol{v} \cap S(G)$ represents the source nodes in $\boldsymbol{v}$, which do not have any direct predecessors. For $l \in \mathbb{N}$, $\mathscr{P}^l(\boldsymbol{x}_{\boldsymbol{v}})$ denotes applying the operator $\mathscr{P}(\cdot)$ on $\boldsymbol{x}_{\boldsymbol{v}}$ l times.

Network PCE can be applied to any DAG, which contains any of the four possible 3-node motifs [11]. Fig. 2.1 shows an example DAG, which contains all the four motifs (e.g., $\{v_1, v_2, v_7\}, \{v_2, v_7, v_8\}, \{v_4, v_5, v_9\}, \{v_6, v_{10}, v_{12}\}$). Note the network inputs $\boldsymbol{\xi} = (x_{v_1}, x_{v_3}, x_{v_4}, x_{v_5}, x_{v_6})$ are mutually independent. The node v_{13} is the sink node corresponding to the network output y. This example DAG will be used in the following subsections to illustrate naïve PCE and network PCE.

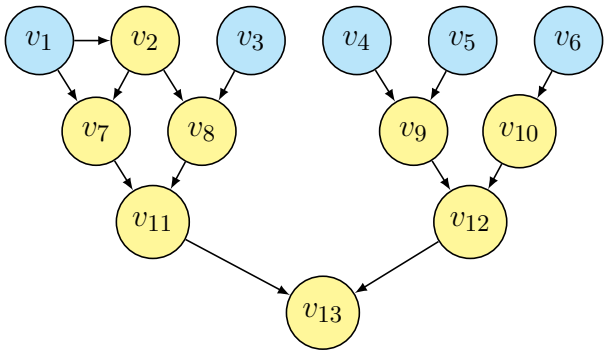


Figure 2.1: This example network contains all the four 3-node motifs of DAG. The network inputs represented by the blue-shaded nodes, $x_{v_1}, x_{v_3}, x_{v_4}, x_{v_5}$, and x_{v_6}, are mutually independent. The variable represented by node v_{13}, $x_{v_{13}}$, is the network output y. The arrows represent dependent relationships; e.g., $y = x_{v_{13}}$ directly depends on $x_{v_{11}}$ and $x_{v_{12}}$, while indirectly depending on all the network inputs.

2.4.2 Naïve PCE

Naïve PCE is the standard PCE in eq. (2.1) that approximates y directly as a function of the network inputs that influence y, $\boldsymbol{\xi} = \boldsymbol{x}_{S(G) \cap V_y}$, as follows:

$$\hat{y} = \sum_{i=0}^{P} \theta_i \psi_i(\boldsymbol{\xi}). \tag{2.10}$$

This PCE directly yields the estimated Sobol indices of y with respect to $\boldsymbol{\xi}$ based on eqs. (2.8) and (2.9). The naïve PCE algorithm is summarized in Algorithm 1.

Algorithm 1 Naïve PCE Algorithm

Input: $G = DAG(V, E)$; at least $P+1$ observations of the network output y and inputs $\boldsymbol{\xi} = \boldsymbol{x}_{S(G)\cap V_y}$.

Output: Sobol indices of y with respect to each input in $\boldsymbol{\xi}$.

1: Construct univariate orthonormal polynomials for each input in $\boldsymbol{\xi}$.
2: Construct multivariate orthonormal polynomials for $\boldsymbol{\xi}$ as the tensor products of the univariate orthonormal polynomials using eq. (2.5).
3: Estimate $\boldsymbol{\theta}$ in eq. (2.10) by solving an equivalent problem to eq. (2.2).
4: Estimate the Sobol indices based on the estimated $\boldsymbol{\theta}$ using eqs. (2.8) and (2.9).

As discussed in Sec. 2.3.2, the number of orthonormal polynomials, P, increases exponentially in $\dim(\boldsymbol{\xi})$. Thus, this naïve approach of taking the network as a black-box requires an exponentially increasing number of observations to solve an equivalent problem to eq. (2.2) as $\dim(\boldsymbol{\xi})$ increases. In other words, a sensitivity analysis with respect to a large number of network inputs requires a large number of observations for naïve PCE. This issue is mitigated by network PCE that explicitly considers the network structure.

2.4.3 Network PCE

We propose network PCE to leverage the known network structure of the process of interest to improve the efficiency and accuracy of sensitivity analysis. Intuitively speaking, this model recursively relates a network node to its direct predecessors to trace the output variance back to the network inputs. How uncertainties propagate through the network is effectively captured by the PCE coefficients in the model. The coefficients directly yield estimated Sobol indices of the output with respect to each input in the network.

The recursive modeling process for network PCE starts from the sink node v_y (e.g., v_{13} in

Fig. 2.1) and traces it back to the source nodes (e.g., v_1, v_3, v_4, v_5, v_6 in Fig. 2.1). The network output y is first modeled as the PCE in $\mathscr{P}(y)$ (e.g., $(x_{v_{11}}, x_{v_{12}})$ in Fig. 2.1), which includes the variables corresponding to the direct predecessors of v_y. Then, their direct predecessors are recursively found until y is modeled as the PCE in $\boldsymbol{\xi}$, or equivalently, $\mathscr{P}^L(y)$, where $L := \inf\{l \in \mathbb{N} : \mathscr{P}^l(y) = \mathscr{P}^{l+1}(y)\}$ denotes the total number of iterations.

In the l^{th} iteration of this recursive process for $l = 1, \ldots, L$, we need to find mutually independent vectors to use in a PCE (recall eq. (2.5)). Thus, we define the *mutually independent decomposition* of $\mathscr{P}^l(y)$ as follows:

$$\boldsymbol{x}_{\boldsymbol{v}^{(l)}} := \left(\boldsymbol{x}_{\boldsymbol{v}_1^{(l)}}, \boldsymbol{x}_{\boldsymbol{v}_2^{(l)}}, \ldots, \boldsymbol{x}_{\boldsymbol{v}_{n^{(l)}}^{(l)}} \right), \tag{2.11}$$

which has an arbitrary order of the elements and satisfies the following two conditions:

1. $\boldsymbol{x}_{\boldsymbol{v}_i^{(l)}} \perp\!\!\!\perp \boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}, \forall i \neq j$;
2. $x_{v_j} \not\perp\!\!\!\perp x_{v_k}, \forall v_j, v_k \in \boldsymbol{v}_i^{(l)}, \forall i$.

Note $n^{(l)} := \dim\left(\boldsymbol{x}_{\boldsymbol{v}^{(l)}}\right)$ is the number of elements of the tuple in eq. (2.11). For example, in Fig. 2.1, the mutually independent decompositions of $\mathscr{P}(y)$ and $\mathscr{P}^2(y)$ are $(x_{v_{11}}, x_{v_{12}})$ and $((x_{v_7}, x_{v_8}), x_{v_9}, x_{v_{10}})$, respectively. Hence, $n^{(1)} = 2$ and $n^{(2)} = 3$, not 4.

In the l^{th} iteration, network PCE approximates y using the PCE as follows:

$$\hat{y}^{(l)} := \sum_{i=0}^{P^{(l)}} \theta_i^{(l)} \psi_i^{(l)}\left(\mathscr{P}^l(y)\right), \tag{2.12}$$

where each orthonormal polynomial $\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)$ can be obtained using the mutually independent decomposition of $\mathscr{P}^l(y)$ in eq. (2.11) as follows:

$$\psi_i^{(l)}\left(\mathscr{P}^l(y)\right) = \psi_{\boldsymbol{\alpha}_i^{(l)}}^{(l)}\left(\mathscr{P}^l(y)\right) := \prod_{j=1}^{n^{(l)}} \psi_{\boldsymbol{\alpha}_{ij}^{(l)}}^{(l)}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right). \tag{2.13}$$

Here $\boldsymbol{\alpha}_i^{(l)} = \left(\boldsymbol{\alpha}_{i1}^{(l)}, \ldots, \boldsymbol{\alpha}_{in^{(l)}}^{(l)}\right)$ is the i-th arbitrary tuple satisfying $\sum_{j=1}^{n^{(l)}} \left|\boldsymbol{\alpha}_{ij}^{(l)}\right| \leq p^{(l)}$ for the pre-specified highest polynomial order $p^{(l)}$. $\boldsymbol{\alpha}_{ij}^{(l)}$ is the vector composed of the polynomial

orders with respect to the variables in $\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}$ for $j = 1, 2, \ldots, n^{(l)}$. $\psi^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$ is an $\left|\boldsymbol{\alpha}_{ij}^{(l)}\right|$-th order orthonormal polynomial in $\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}$ obtained using the modified Gram-Schmidt algorithm [39].

In the first iteration ($l = 1$), the PCE coefficients, $\theta_i^{(l)}, i = 0, 1, \ldots, P^{(l)}$, in eq. (2.12) are estimated by solving an equivalent problem to eq. (2.2) where only the notations are different. For the subsequent iterations ($l \geq 2$), the coefficients are calculated in a different way using the previous iteration's coefficients, as detailed next.

To model y as a function of $\mathscr{P}^{l+1}(y)$, or equivalently,

$$\left(\mathscr{P}\left(\boldsymbol{x}_{\boldsymbol{v}_1^{(l)}}\right), \mathscr{P}\left(\boldsymbol{x}_{\boldsymbol{v}_2^{(l)}}\right), \ldots, \mathscr{P}\left(\boldsymbol{x}_{\boldsymbol{v}_{n^{(l)}}^{(l)}}\right)\right),$$

network PCE substitutes $\psi^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$ in eq. (2.13) with

$$\hat{\psi}^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right) := \sum_{k=0}^{P_{ij}^{(l)}} \theta_{ijk}^{(l)} \psi_{ijk}^{(l)}\left(\mathscr{P}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)\right) \tag{2.14}$$

to obtain the approximation of eq. (2.13) as follows:

$$\hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right) := \prod_{j=1}^{n^{(l)}} \hat{\psi}^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right). \tag{2.15}$$

Without loss of generality, we let the highest polynomial order $p_{ij}^{(l)}$ of orthonormal polynomials in eq. (2.14) to be the constant $p^{(l+1)}$. Substituting $\hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right)$ in eq. (2.15) for $\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)$ in eq. (2.12) yields the approximation of y for the $(l+1)^{th}$ iteration as follows:

$$\hat{y}^{(l+1)} = \sum_{i=0}^{P^{(l)}} \theta_i^{(l)} \hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right) \tag{2.16}$$

$$:= \sum_{i=0}^{P^{(l+1)}} \theta_i^{(l+1)} \psi_i^{(l+1)}\left(\mathscr{P}^{l+1}(y)\right), \tag{2.17}$$

where the PCE coefficient $\theta_i^{(l+1)}$ in eq. (2.17) is calculated after estimating the coefficients in eq. (2.14) (by solving an equivalent problem to eq. (2.2)) and rearranging the terms in

eq. (2.16), which involve the previous iteration's coefficients. This recursive approach of calculating the PCE coefficient helps reduce the minimally required number of observations compared to naïve PCE, as explained later.

The above iteration ends when y is approximated as the PCE in $\mathscr{P}^L(y)$, or equivalently, $\boldsymbol{\xi}$ as follows:

$$\hat{y}^{(L)} := \sum_{i=0}^{P^{(L)}} \theta_i^{(L)} \psi_i(\boldsymbol{\xi}), \tag{2.18}$$

where $P^{(L)}$ depends on the highest polynomial orders, $p^{(1)}, \ldots, p^{(L)}$. This recursive approximation process is valid as shown in Theorem 3 below, which proves the convergence of $\hat{y}^{(L)}$ in eq. (2.18) to y in probability. The PCE coefficients, $\theta_i^{(L)}, i = 1, \ldots, P^{(L)}$, in eq. (2.18) directly yield the estimated Sobol indices of y with respect to each input in $\boldsymbol{\xi}$, as described in Sec. 2.3.6. The network PCE algorithm is summarized in Algorithm 2.

An advantage of network PCE over naïve PCE lies in its potential to use much fewer observations for solving equivalent problems of eq. (2.2). The minimally required number of observations for either model is equal to the largest number of orthonormal polynomials of any PCE used in the model. Thanks to the recursive decomposition procedure of network PCE, it tends to use a much smaller PCE than naïve PCE especially when the number of the inputs that influence y is large (i.e., $\dim(\boldsymbol{\xi}) \gg 1$). To illustrate this point, assume all PCEs in network PCE use the same highest polynomial order $p^{(l)} = p$, $l = 1, 2, \ldots, L$, as naïve PCE. Then, the minimally required number of observations for naïve PCE increases exponentially in $\dim(\boldsymbol{\xi})$, as explained below eq. (2.1). That for network PCE (see Input of Algorithm 2) increases exponentially in the largest number of variables used for any PCE in Steps 3 and 4 of Algorithm 2,

$$\max\left(|D_y|, \max_{1 \le l \le L-1} \max_{1 \le j \le n^{(l)}} \left|D_j^{(l)}\right|\right). \tag{2.19}$$

Here,

$$D_y := \{v_i \in V : \langle v_i, v_y \rangle \in E\} \tag{2.20}$$

denotes the set of all direct predecessors of the output node v_y and

$$D_j^{(l)} := \{v_i \in V : v_k \in \boldsymbol{v}_j^{(l)}, \langle v_i, v_k \rangle \in E\} \tag{2.21}$$

Algorithm 2 Network PCE Algorithm

Input: $G = DAG(V, E)$; at least

$$\max\left(\begin{array}{c} P^{(1)}, \max\limits_{\substack{1\le l\le L-1\\ 0\le i\le P^{(l)}\\ 1\le j\le n^{(l)}}} P_{ij}^{(l)} \end{array}\right)+1$$

observations of the network output y and all the variables in $\boldsymbol{x}_{V_y}$; Iteration counter $l = 1$.

Output: Sobol indices of y with respect to each input in $\boldsymbol{\xi} = \boldsymbol{x}_{S(G)\cap V_y}$.

1: Construct $\psi^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$ in eq. (2.13) using the modified Gram-Schmidt algorithm for $i = 0, 1, \ldots, P^{(l)}$ and $j = 1, 2, \ldots, n^{(l)}$.

2: Construct $\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)$ for $i = 0, 1, \ldots, P^{(l)}$ in eq. (2.12) as the tensor product of $\psi^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$ for $j = 1, 2, \ldots, n^{(l)}$ in eq. (2.13).

3: If $l = 1$, estimate the PCE coefficients, $\theta_i^{(l)}, i = 0, 1, \ldots, P^{(l)}$, in eq. (2.12) by solving an equivalent problem to eq. (2.2). If $L = 1$, skip to Step 6.

4: Estimate the PCE coefficients in eq. (2.14) and substitute eq. (2.15) into eq. (2.16).

5: Rearrange the terms in eq. (2.16) to yield the expression in eq. (2.17). Increment l by 1 and if $l < L$, go to Step 1.

6: Estimate the Sobol indices using eqs. (2.8) and (2.9) based on the PCE coefficients $\theta_i^{(L)}, i = 0, 1, \ldots, P^{(L)}$, in eq. (2.18).

denotes the set of all direct predecessors of the nodes in $\boldsymbol{v}_j^{(l)}$ in eq. (2.14). Fig. 2.2 visualizes the minimally required number of observations with respect to p and the ratio

$$\lambda := \frac{\max\left(|D_y|, \max\limits_{1\le l\le L-1}\max\limits_{1\le j\le n^{(l)}}\left|D_j^{(l)}\right|\right)}{\dim(\boldsymbol{\xi})}. \tag{2.22}$$

The saving of network PCE (with $\lambda < 1$) over naïve PCE ($\lambda = 1$) increases as p increases or λ decreases.

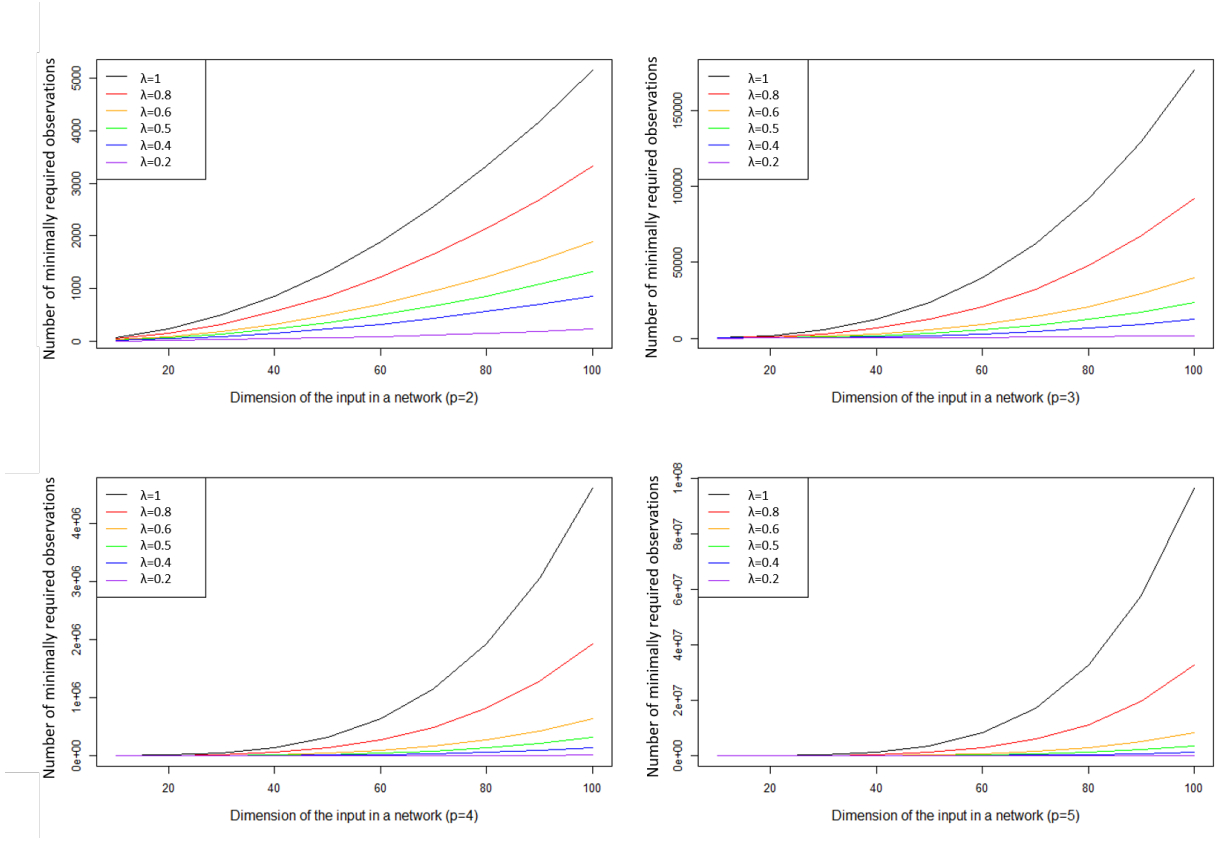


Figure 2.2: Network PCE ($\lambda < 1$) requires fewer observations to model the network output than naïve PCE ($\lambda = 1$) for a larger p (the highest polynomial order used in both PCEs) and a smaller λ. λ, defined in eq. (2.22), represents the ratio of the largest number of variables used in any PCE for network PCE to the number of network inputs used in naïve PCE.

2.4.4 Sparse network PCE (SN-PCE)

While the proposed network PCE already provides substantial parsimony over naïve PCE, it may still need to use many orthonormal polynomials to capture all the main effects and the major interaction effects of the network variables on the output variance. However, in practice, many of such effects are insignificant. Therefore, we propose SN-PCE, which imposes l_1-sparsity on the PCE coefficients to capture only significant pathways of uncertainty propagation in the network. Specifically, SN-PCE solves an equivalent problem of eq. (2.4)

(with the difference lying only in the notations) instead of eq. (2.2) in Steps 3 and 4 of Algorithm 2. The resulting parsimonious model can use even fewer observations than network PCE to estimate the PCE coefficients and, in turn, the Sobol indices. Also, because only significant effects appear in the model (with non-zero PCE coefficients), the model is easier to interpret than the other models.

2.4.5 Theoretical result

Since naïve PCE is a direct application of standard PCE in eq. (2.1), the corresponding theoretical result (Theorem 1) directly follows from [55] by appropriately invoking Assumption 1 below. The result is still presented here mainly for comparison with the theoretical result on network PCE (Theorem 3).

Assumption 1 (adapted from [55])**.** *The random vector* $\boldsymbol{x} \coloneqq (x_1, \ldots, x_n)^T : (\Omega, \mathcal{F}) \to (\mathbb{A}^n, \mathcal{B}^n)$

1. *has a continuous joint probability density function* $\mu(\boldsymbol{x})$ *with a bounded or unbounded support* $\mathbb{A}^n \subseteq \mathbb{R}^n$*;*

2. *possesses absolute finite moments of all orders, that is,* $\forall \mathbf{j} \coloneqq (j_1, j_2, \ldots, j_n) \in \mathbb{N}_0^n$*,*

$$\mathbb{E}\left(\left|\boldsymbol{x}^{\mathbf{j}}\right|\right) := \int_{\Omega} \left|\boldsymbol{x}^{\mathbf{j}}(w)\right| d\mathcal{P}(w) < \infty,$$

   where $\boldsymbol{x}^{\mathbf{j}} := x_1^{j_1} \cdots x_n^{j_n}$*; and*

3. *has a joint probability density function* $\mu(\boldsymbol{x})$*, which*

   (a) *has a compact support, that is, there exists a compact subset* $\mathbb{A}^n \subset \mathbb{R}^n$ *such that* $\mathcal{P}(\boldsymbol{x} \in \mathbb{A}^n) = 1$*, or*

   (b) *is exponentially integrable, that is, there exists a real number* $\alpha > 0$ *such that*

$$\int_{\mathbb{A}^n} \exp\left(\alpha ||\boldsymbol{x}||\right) \mu(\boldsymbol{x})\, d\boldsymbol{x} < \infty,$$

   where $||\cdot|| : \mathbb{A}^n \to \mathbb{R}_0^+$ *is an abitrary norm.*

Theorem 1. *Suppose that for the network output* $y(\boldsymbol{\xi}) \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$, $\boldsymbol{\xi} = \boldsymbol{x}_{S(G)\cap V_y}$ *fulfills Assumption 1. Then,* $\hat{y}$ *in eq.* (2.10) *converges to* y *in probability as* $P \to \infty$.

Proof. This is a direct result of Theorem 11 in [55]. □

Theorem 1 imposes regularity conditions on the network inputs $\boldsymbol{\xi}$ and output y, but not on the other network variables. This black-box approach of naïve PCE still provides a convergent prediction of y as $P \to \infty$. In practice, increasing P requires increasing the number of observations. Thus, the finite-sample inefficiency of naïve PCE compared to network PCE (described in Sec. 2.4.3) is critical when the sample size is limited.

Theoretical results on network PCE impose regularity conditions on all the network variables as the knowledge of network structure provides network PCE's advantage over naïve PCE. Lemma 2 validates the recursive modeling of a network node using its direct predecessors (described in Steps 1–5 in Algorithm 2). Building on Lemma 2, Theorem 3 proves the convergence of network PCE output in eq. (2.18).

Lemma 2. *Suppose that* $\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}$ *is a function of* $\mathscr{P}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$ *for* $l = 1, \ldots, L-1$ *and* $j = 1, \ldots, n^{(l)}$, *and that for the network output* y, $\boldsymbol{x}_{V_y}$ *fulfills Assumption 1. Then,*

1. $\hat{\psi}^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$ *in eq.* (2.14) *converges to* $\psi^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$ *in eq.* (2.13) *in probability as* $P_{ij}^{(l)} \to \infty$; *and*
2. $\hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right) = \prod_{j=1}^{n^{(l)}} \hat{\psi}^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$ *in eq.* (2.15) *converges to* $\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)$ *in eq.* (2.13) *in probability as* $P_{ij}^{(l)} \to \infty$

for all $l = 1, \ldots, L-1$ *and all* $\boldsymbol{\alpha}_i^{(l)} = \left(\boldsymbol{\alpha}_{i1}^{(l)}, \ldots, \boldsymbol{\alpha}_{in^{(l)}}^{(l)}\right)$ *satisfying* $\sum_{j=1}^{n^{(l)}} \left|\boldsymbol{\alpha}_{ij}^{(l)}\right| \leq p^{(l)}$.

Proof. Under the stated assumptions, $\psi^{(l)}_{\boldsymbol{\alpha}_{ij}^{(l)}}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$ in eq. (2.13) is a square-integrable function of $\mathscr{P}\left(\boldsymbol{x}_{\boldsymbol{v}_j^{(l)}}\right)$. The first statement follows from Theorem 11 in [55]. The second statement follows from the first statement by the continuous mapping theorem. □

Theorem 3. *Suppose that the assumptions in Lemma 2 hold for the network output* $y\left(\boldsymbol{x}_{\boldsymbol{v}^{(1)}}\right) \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$. *Then, there exists an increasing function* $\gamma^{(l)} : \mathbb{N} \mapsto \mathbb{N}$ *such that if* $p^{(l+1)} = \gamma^{(l)}\left(P^{(l)}\right)$ *for* $l = 1, \ldots, L-1$, $\hat{y}^{(L)}$ *in eq.* (2.18) *converges to* y *in probability as* $p^{(1)} \to \infty$, *or equivalently,* $P^{(1)} \to \infty$.

The proof is provided in Appendix A.2.

The increasing function $\gamma^{(l)}$ in Theorem 3 prescribes how fast the highest polynomial order $p^{(l+1)} = \gamma^{(l)}\left(P^{(l)}\right)$ should grow in relation to the number of orthonormal polynomials, $P^{(l)}$, for $l = 1, \ldots, L-1$ so that $\hat{y}^{(L)}$ in eq. (2.18) converges to y in probability. Furthermore, to make the relation between $p^{(l+1)}$ and $p^{(l)}$ more explicit, using eqs. (2.20) and (2.21), let $d^{(1)} := |D_y|$ and $d^{(l)} := \left| \bigcup_{1 \le j \le n^{(l)}} D_j^{(l-1)} \right|$, $l = 2, \ldots, L$, denote the number of nodes represented by $\mathscr{P}(y)$ and $\mathscr{P}^l(y)$ in eq. (2.12), respectively. Because

$$P^{(l)} + 1 = \binom{d^{(l)} + p^{(l)}}{d^{(l)}},$$

for $l = 1, \ldots, L$, Stirling's approximation yields

$$p^{(l+1)} = \gamma^{(l)}\left(O\left(\left(p^{(l)}\right)^{d^{(l)}}\right)\right)$$

using the big O notation.

For example, if $\gamma^{(l)}$ is linear (or superlinear), then $p^{(l+1)}$ should grow as fast as (or faster than) $d^{(l)}$-th power of $p^{(l)}$, or equivalently, $\prod_{l'=1}^{l} d^{(l')}$-th power of $p^{(1)}$ for $l = 1, \ldots, L-1$. Intuitively speaking, we should control the approximation errors for the upstream nodes in the network more tightly to ensure that the approximation errors for the downstream nodes (especially the output node) are arbitrarily small. We note that it is beyond the scope of this paper to establish more detailed characteristics of $\gamma^{(l)}$.

SN-PCE, which tends to use much smaller $P^{(l)}, l = 1, \ldots, L$, than network PCE, mitigates the need for rapidly increasing $p^{(l)}, l = 1, \ldots, L$, while maintaining a similar approximation accuracy.

2.5 Applications

For empirical evaluation of the proposed methodology, we conduct sensitivity analysis of two manufacturing processes (welding and injection molding) in [47]. We use the same modeling equations and notations therein (hence, some notation conflicts arise although their meanings are clear from the context). Note that the proposed methodology leverages the known directional relationships between inputs and the output (expressed as a DAG), not modeling equations, which are often unknown in practice.

We compare four methods, namely, random sampling [51], orthogonal array sampling [66], naïve PCE, and SN-PCE, for estimating the first-order and total Sobol indices. The estimation is replicated 50 times to calculate the sample mean and standard error of the estimated Sobol indices for each method.

As discussed in Sec. 2.3.5, the Sobol indices measure the influence of each network input on the variance of the network output. Henceforth, we call some network inputs *influential inputs* if their first-order Sobol indices are 10^{-1} or larger; in other words, the influential inputs explain 10% or more of the output variance without considering their interactions with the other inputs.

We use a large sample for random and orthogonal array sampling, namely, 10,000 observations unless specified otherwise. We treat the sample mean from 50 replications using the orthogonal array sampling as the ground truth if the standard error is below 1% of the sample mean. Even with a large sample, such Monte Carlo sampling methods tend to have large standard errors for estimating small *total* Sobol indices [46, 51], as also observed in this study.

2.5.1 Welding process

The welding process has several process variables whose relationships are depicted in Fig. 2.3. The process output of interest is the total minimum theoretical energy required for the welding process, E. This energy depends on the weld volume V, specific gravity ρ, heat

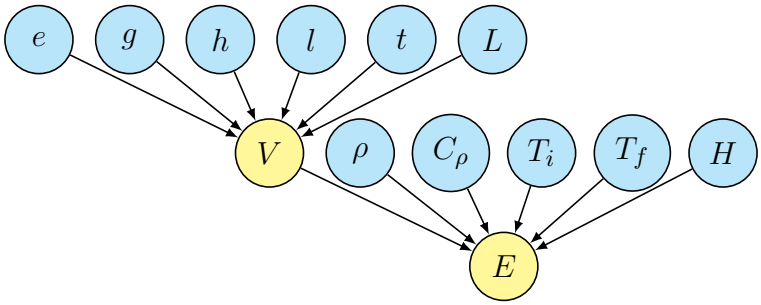


Figure 2.3: This DAG represents the relationships among the variables in the welding process [47]; the weld volume V depends on six welding parameters (weld zone dimensions: e, g, h, l, and t; weld length L), and the total energy E depends on V and additional parameters, ρ, C_p, T_i, T_f, and H.

capacity C_p, initial temperature T_i, final temperature T_f, and latent heat H as follows:

$$E = \rho V \left(C_p(T_f - T_i) + H \right).$$

In turn, the weld volume V depends on six welding parameters (weld zone dimensions: e, g, h, l, and t; weld length L) as follows:

$$V = L\left(0.75lh + gt + 0.5(l-9)(t-e)\right).$$

The process inputs' distributions are presented in Table 2.1.

We first compare the four methods with respect to the mean squared errors of estimating the Sobol indices when using the same sample size. Fig. 2.4 shows that SN-PCE significantly outperforms random and orthogonal array sampling. Also, SN-PCE estimates the Sobol indices using a much smaller sample size than naïve PCE. In this example, we use the highest polynomial order of 3 for both naïve PCE and SN-PCE; naïve PCE requires at least $364 = \binom{11+3}{3}$ observations for $\dim(\boldsymbol{\xi}) = 11$ and network PCE (i.e., SN-PCE without sparsity) requires at least $84 = \binom{6+3}{3}$ observations because eq. (2.19) equals 6. SN-PCE could use even fewer observations because it identifies only 14 orthonormal polynomials (4% of those used in naïve PCE) as necessary to approximate the network output of this particular process across all the 50 replications.

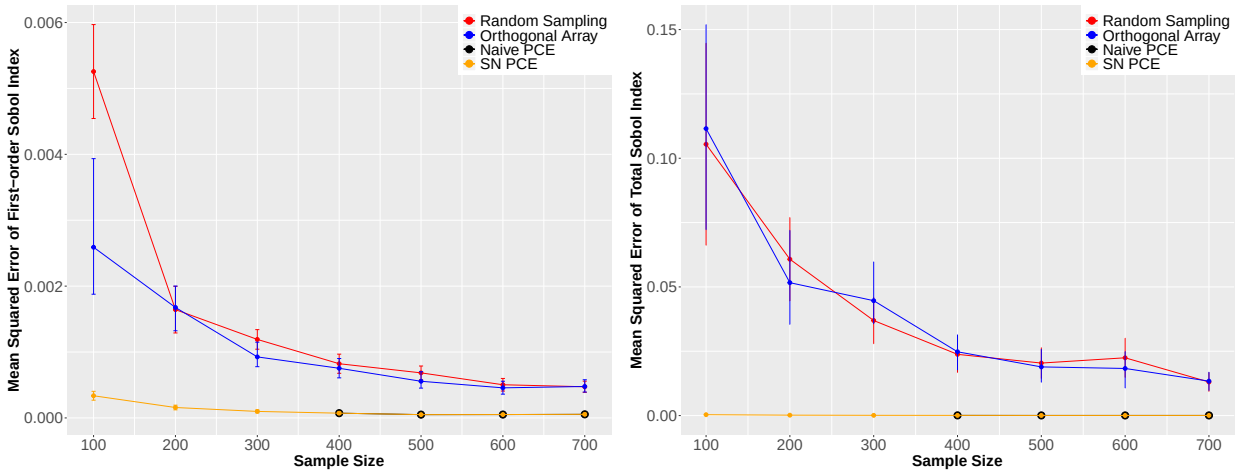


Figure 2.4: SN-PCE yields a smaller mean squared error of estimating the first-order Sobol indices (left) and the total Sobol indices (right) than the three other methods for the welding process. Also, SN-PCE requires much fewer observations than naïve PCE to estimate the Sobol indices.

Table 2.1 shows the sample means and standard errors of the estimated Sobol indiceses for the four methods, where naïve PCE and SN-PCE use 500 and 100 observations, respectively. Despite the vastly different sample sizes used for each method, the sample means are nearly identical across the methods (except for the non-influential inputs, which have total Sobol indices smaller than 10^{-1}). This indicates the estimation bias is nearly zero for these methods. The standard errors are also very similar across the methods for the influential inputs, indicating that SN-PCE achieves a similar accuracy as the other methods with a much smaller sample size.

Fig. 2.5 presents a Pareto chart of the first-order Sobol indices estimated from SN-PCE. Along with Table 2.2, the chart confirms that the weld zone dimensions (h, g, t, e, l) are the most influential inputs for the variance of the process output E. Also, the cumulative sum of the first-order Sobol indices approaches 100% in the chart, implying that the interactions between the inputs do not have significant effects on the variance of E. It echoes the finding from Table 2.2 that the first-order Sobol indices are approximately equal to the total Sobol

indices across all the influential inputs.

Table 2.1: Sample means and standard errors (rounded to two decimal places) of the estimated first-order and total Sobol indices based on 50 replications for the welding process. The inputs in the first column are sorted in descending order of the first-order Sobol indices estimated from the Monte Carlo method. For each replication,†random sampling, ‡orthogonal array sampling, ††Naïve PCE, and ‡‡SN-PCE use 10,000, 10,000, 500, and 100 observations, respectively. For the influential inputs, SN-PCE attains similar standard errors as the other methods despite using the smallest sample size.

Input	Distribution	Random Sampling†		Orthogonal Array‡		Naïve PCE††		SN-PCE‡‡	
		First-order	Total	First-order	Total	First-order	Total	First-order	Total
h	$N(2.6, 0.5)$	2.76×10^{-1} $\pm 0.02 \times 10^{-1}$	2.85×10^{-1} $\pm 0.04 \times 10^{-1}$	2.78×10^{-1} $\pm 0.01 \times 10^{-1}$	2.81×10^{-1} $\pm 0.03 \times 10^{-1}$	2.78×10^{-1} $\pm 0.01 \times 10^{-1}$	2.80×10^{-1} $\pm 0.01 \times 10^{-1}$	2.80×10^{-1} $\pm 0.02 \times 10^{-1}$	2.81×10^{-1} $\pm 0.02 \times 10^{-1}$
g	$N(2, 0.1)$	2.31×10^{-1} $\pm 0.01 \times 10^{-1}$	2.30×10^{-1} $\pm 0.03 \times 10^{-1}$	2.34×10^{-1} $\pm 0.01 \times 10^{-1}$	2.32×10^{-1} $\pm 0.03 \times 10^{-1}$	2.32×10^{-1} $\pm 0.01 \times 10^{-1}$	2.33×10^{-1} $\pm 0.01 \times 10^{-1}$	2.34×10^{-1} $\pm 0.02 \times 10^{-1}$	2.35×10^{-1} $\pm 0.02 \times 10^{-1}$
t	$N(15, 0.6)$	2.29×10^{-1} $\pm 0.01 \times 10^{-1}$	2.24×10^{-1} $\pm 0.03 \times 10^{-1}$	2.27×10^{-1} $\pm 0.01 \times 10^{-1}$	2.32×10^{-1} $\pm 0.04 \times 10^{-1}$	2.29×10^{-1} $\pm 0.01 \times 10^{-1}$	2.29×10^{-1} $\pm 0.01 \times 10^{-1}$	2.25×10^{-1} $\pm 0.02 \times 10^{-1}$	2.26×10^{-1} $\pm 0.02 \times 10^{-1}$
e	$N(11, 1)$	1.46×10^{-1} $\pm 0.01 \times 10^{-1}$	1.48×10^{-1} $\pm 0.04 \times 10^{-1}$	1.46×10^{-1} $\pm 0.01 \times 10^{-1}$	1.50×10^{-1} $\pm 0.04 \times 10^{-1}$	1.45×10^{-1} $\pm 0.01 \times 10^{-1}$	1.47×10^{-1} $\pm 0.01 \times 10^{-1}$	1.46×10^{-1} $\pm 0.01 \times 10^{-1}$	1.48×10^{-1} $\pm 0.01 \times 10^{-1}$
l	$N(8.5, 0.5)$	1.07×10^{-1} $\pm 0.01 \times 10^{-1}$	1.08×10^{-1} $\pm 0.04 \times 10^{-1}$	1.08×10^{-1} $\pm 0.01 \times 10^{-1}$	1.12×10^{-1} $\pm 0.04 \times 10^{-1}$	1.07×10^{-1} $\pm 0.00 \times 10^{-1}$	1.12×10^{-1} $\pm 0.00 \times 10^{-1}$	1.08×10^{-1} $\pm 0.01 \times 10^{-1}$	1.12×10^{-1} $\pm 0.01 \times 10^{-1}$
L	$N(500, 10)$	1.96×10^{-3} $\pm 0.01 \times 10^{-3}$	4.45×10^{-4} $\pm 43.4 \times 10^{-4}$	1.95×10^{-3} $\pm 0.01 \times 10^{-3}$	2.57×10^{-3} $\pm 4.25 \times 10^{-3}$	1.96×10^{-3} $\pm 0.01 \times 10^{-3}$	2.00×10^{-3} $\pm 0.01 \times 10^{-3}$	1.94×10^{-3} $\pm 0.03 \times 10^{-3}$	1.95×10^{-3} $\pm 0.03 \times 10^{-3}$
C_ρ	$N(500, 5)$	9.67×10^{-4} $\pm 0.07 \times 10^{-4}$	-9.76×10^{-4} $\pm 43.02 \times 10^{-4}$	9.66×10^{-4} $\pm 0.05 \times 10^{-4}$	4.94×10^{-6} $\pm 0.04 \times 10^{-2}$	9.69×10^{-4} $\pm 0.05 \times 10^{-4}$	9.89×10^{-4} $\pm 0.05 \times 10^{-4}$	9.77×10^{-4} $\pm 0.15 \times 10^{-4}$	9.77×10^{-4} $\pm 0.15 \times 10^{-4}$
T_f	$N(1628, 10)$	2.75×10^{-4} $\pm 0.02 \times 10^{-4}$	-1.64×10^{-3} $\pm 4.35 \times 10^{-3}$	2.76×10^{-4} $\pm 0.05 \times 10^{-4}$	1.50×10^{-6} $\pm 4.25 \times 10^{-3}$	2.75×10^{-4} $\pm 0.01 \times 10^{-4}$	2.81×10^{-4} $\pm 0.01 \times 10^{-4}$	7.55×10^{-5} $\pm 1.02 \times 10^{-5}$	7.55×10^{-5} $\pm 1.02 \times 10^{-5}$
T_i	$N(303, 0.3)$	8.32×10^{-6} $\pm 0.04 \times 10^{-6}$	-1.74×10^{-3} $\pm 0.04 \times 10^{-3}$	8.23×10^{-6} $\pm 0.05 \times 10^{-6}$	2.76×10^{-4} $\pm 4.25 \times 10^{-3}$	8.28×10^{-6} $\pm 0.04 \times 10^{-6}$	8.45×10^{-6} $\pm 0.04 \times 10^{-6}$	0	0
ρ	$N(8238, 10)$	7.23×10^{-6} $\pm 0.05 \times 10^{-6}$	-1.78×10^{-3} $\pm 4.31 \times 10^{-3}$	7.20×10^{-6} $\pm 4.06 \times 10^{-8}$	4.06×10^{-8} $\pm 4.25 \times 10^{-3}$	7.16×10^{-6} $\pm 0.03 \times 10^{-6}$	7.30×10^{-6} $\pm 0.03 \times 10^{-6}$	0	0
H	$N(2270, 3)$	3.32×10^{-10} $\pm 0.02 \times 10^{-10}$	-1.77×10^{-3} $\pm 4.32 \times 10^{-3}$	3.30×10^{-10} $\pm 0.02 \times 10^{-10}$	1.94×10^{-12} $\pm 4.25 \times 10^{-3}$	3.31×10^{-10} $\pm 4.25 \times 10^{-3}$	3.41×10^{-10} $\pm 0.02 \times 10^{-10}$	0	0

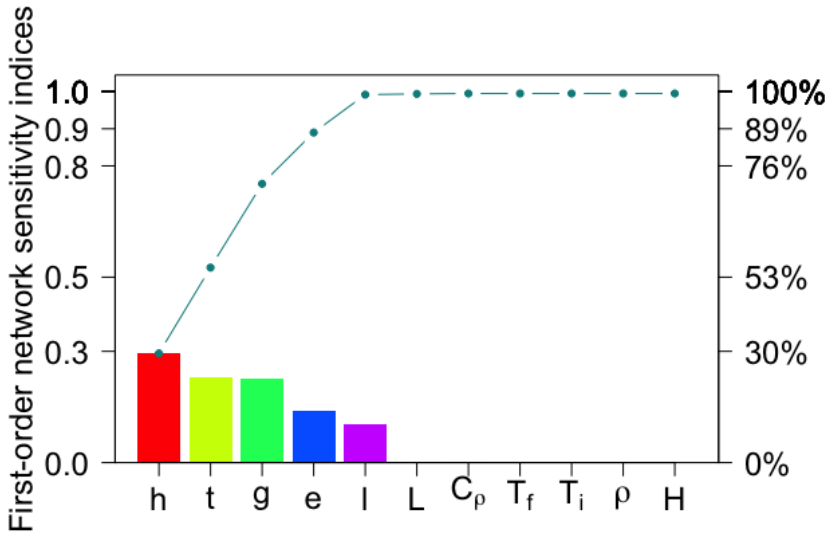


Figure 2.5: Pareto chart of the first-order Sobol indices estimated using SN-PCE for the welding process. Higher bar indicates that the input has a more influence (excluding interactions with other inputs) on the variance of the process output E.

2.5.2 Injection molding process

The injection molding process has more intricate relationships between process variables than the welding process, as depicted in Fig. 2.6. The process output of interest is the energy consumed in resetting the process, E_{reset}. This energy depends on the melting energy E_{melt}, the injection energy E_{inj}, and the cooling energy E_{cool} as follows: $E_{reset} = 0.25\,(E_{melt} + E_{inj} + E_{cool})$. Here, $E_{melt} = P_{melt} \times V_{shot}/Q$, where

$$P_{melt} = \frac{1}{2}\left(\rho \times Q \times C_p \times \left(T_{inj} - T_{pol}\right) + \rho \times Q \times H_f\right),$$
$$V_{shot} = V_{part} \times \left(1 + \frac{\epsilon}{100} + \frac{\Delta}{100}\right).$$

Here, ρ (specific gravity), C_p (heat capacity), T_{pol} (initial polymer temperature), ϵ (shrinkage parameter), and T_{inj} (injection temperature) are the network inputs. Q (flow rate), H_f

(polymer heat of fusion), V_{part} (volume of mold), and Δ (buffer) are constant parameters. On the other hand, $E_{inj} = P_{inj} \times V_{part}$ and

$$E_{cool} = \frac{\rho \times V_{part} \times (C_p \times (T_{inj} - T_{ej}))}{COP},$$

where P_{inj} (injection pressure), T_{ej} (ejection temperature), and T_{inj} (injection temperature) are the network inputs. COP (coefficient of performance of the cooling equipment) is a constant parameter. The network inputs' distributions are presented in Table 2.2.

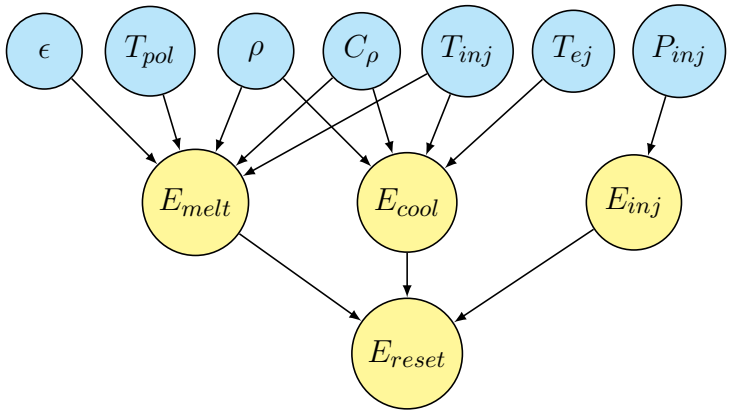


Figure 2.6: This DAG represents the relationships among the variables in the injection molding process [47]; the yellow shaded nodes E_{melt}, E_{cool}, and E_{inj} are the energies consumed in three subprocesses that depend on different blue shaded network inputs. The network output E_{reset} depends on E_{melt}, E_{cool}, and E_{inj}.

To adequately model the more intricate network structure of the injection molding process, we use the highest polynomial order of 4 for both naïve PCE and SN-PCE, compared to 3 used for the welding process; naïve PCE requires at least $330 = \binom{7+4}{4}$ observations for $\dim(\boldsymbol{\xi}) = 7$ and network PCE (i.e., SN-PCE without sparsity) requires at least $126 = \binom{5+4}{4}$ observations because eq. (2.19) equals 5. Thus, we use 500 and 200 observations for naïve PCE and SN-PCE, respectively, compared to 500 and 100 used for the welding process. Yet, again, SN-PCE could use even fewer observations because it identifies only 9 orthonor-

mal polynomials (3% of those used in naïve PCE) as necessary to approximate the network output of this particular process across all 50 replications.

Table 2.2: Sample means and standard errors (rounded to two decimal places) of the estimated first-order Sobol indices based on 50 replications for the injection molding process. For each replication, ‡‡SN-PCE uses 200 observations. All the other setups are the same as in Table 2.1.

Input	Distribution	Random Sampling†	Orthogonal Array‡	Naive PCE††	SN-PCE‡‡
T_{inj}	$N(210,3)$	4.76×10^{-1} $\pm 0.02 \times 10^{-1}$	4.75×10^{-1} $\pm 0.02 \times 10^{-1}$	4.77×10^{-1} $\pm 0.01 \times 10^{-1}$	4.78×10^{-1} $\pm 0.02 \times 10^{-1}$
T_{ej}	$N(35,3)$	2.61×10^{-1} $\pm 0.01 \times 10^{-1}$	2.61×10^{-1} $\pm 0.01 \times 10^{-1}$	2.62×10^{-1} $\pm 0.01 \times 10^{-1}$	2.61×10^{-1} $\pm 0.01 \times 10^{-1}$
ρ	$U(950,990)$	2.27×10^{-1} $\pm 0.01 \times 10^{-1}$	2.27×10^{-1} $\pm 0.01 \times 10^{-1}$	2.26×10^{-1} $\pm 0.00 \times 10^{-1}$	2.26×10^{-1} $\pm 0.01 \times 10^{-1}$
T_{pol}	$N(40,3)$	3.22×10^{-2} $\pm 0.02 \times 10^{-2}$	3.22×10^{-2} $\pm 0.02 \times 10^{-2}$	3.21×10^{-2} $\pm 0.02 \times 10^{-2}$	3.20×10^{-2} $\pm 0.02 \times 10^{-2}$
C_ρ	$U(2250,2260)$	2.61×10^{-3} $\pm 0.01 \times 10^{-3}$	2.62×10^{-3} $\pm 0.01 \times 10^{-3}$	2.61×10^{-3} $\pm 0.01 \times 10^{-3}$	2.61×10^{-3} $\pm 0.01 \times 10^{-3}$
ϵ	$U(0.018,0.021)$	7.76×10^{-9} $\pm 0.04 \times 10^{-9}$	7.74×10^{-9} $\pm 0.04 \times 10^{-9}$	7.76×10^{-9} $\pm 0.03 \times 10^{-9}$	0
P_{inj}	$N(90,4)$	4.75×10^{-14} $\pm 0.02 \times 10^{-14}$	4.73×10^{-14} $\pm 0.02 \times 10^{-14}$	4.77×10^{-14} $\pm 0.02 \times 10^{-14}$	0

Like the welding process, the injection molding process turns out to have the first-order Sobol indices approximately equal to the total Sobol indices for the influential inputs. Thus, we only report the first-order Sobol indices in Table 2.2. Again, SN-PCE attains similar standard errors as the other methods for the influential inputs despite using fewer observations. Fig. 2.7 shows that T_{inj} determines nearly 50% of the variance of network output E_{reset}. T_{ej} and ρ have comparable effects (26% and 23%, respectively), while other inputs,

T_{pol}, C_ρ, ϵ, and P_{inj}, barely influence the variance of E_{reset}.

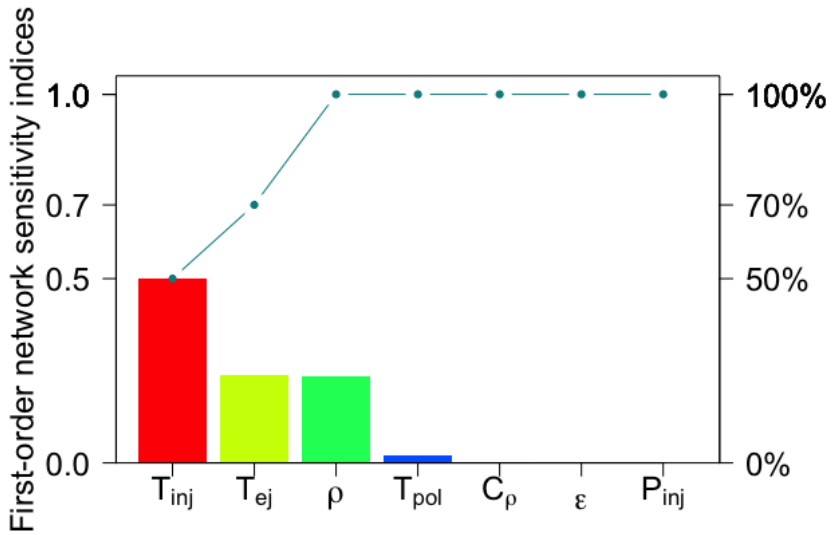


Figure 2.7: Pareto chart of the first-order Sobol indices estimated using SN-PCE for the injection molding process. Higher bar indicates that the input has a more influence (excluding interactions with other inputs) on the variance of process output E_{reset}.

2.5.3 Flooding process

In flooding process modeling, the two outputs of interest are the maximal annual overflow (S) and the associated cost (C_p) of the dyke to prevent the flood [28]. The relationship between the variables is presented in Figure 2.8. The maximal annul overflow S depends on the maximal annual height of the river H and additional parameters, H_d, Z_v, and C_b as follows:

$$S = Z_v + H - H_d - C_b,$$

where H depends on the maximal annual flowrate (Q), Strickler coefficient (K_s), river downstream level (Z_v), river upstream level (Z_m), river width (B), and length of the river stretch

(L) as follows:

$$H = \left(\frac{Q}{BK_s \sqrt{\frac{Z_m - Z_v}{L}}} \right)^{0.6}.$$

Lastly, the associated cost C_p is a *discontinuous* function of S and H_d as follows:

$$C_p = \mathbb{1}_{S>0} + [0.2 + 0.8(1 - e^{-\frac{1000}{S^4}})]\mathbb{1}_{S\leq 0} + \frac{1}{20}(H_d \mathbb{1}_{H_d>8} + 8\mathbb{1}_{H_d\leq 8}).$$

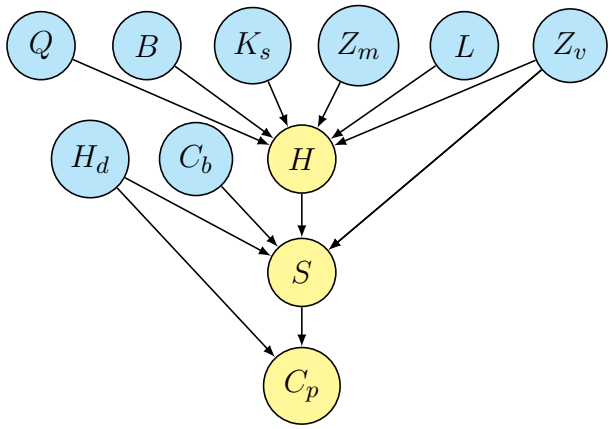


Figure 2.8: This DAG represents the relationships among the variables in the flooding process [28]; the maximal annual height of the river H depends on six parameters (Q, B, K_s, Z_m, L, and Z_v), and the maximal annul overflow S depends on H, H_d, Z_v, and C_b. The associated cost C_p depends on S and H_d.

To estimate the Sobol indices of S and C_p with respect to the inputs, we use the highest polynomial order of 3 for both naïve PCE and SN-PCE. Naïve PCE requires at least $495 = \binom{8+3}{3}$ observations for $\dim(\boldsymbol{\xi}) = 8$ and network PCE (i.e., SN-PCE without sparsity) requires at least $252 = \binom{6+3}{3}$ observations because eq. (2.19) equals 6. Thus, we use 500 and 300 observations for naïve PCE and SN-PCE, respectively, to estimate the Sobol indices. Yet, again, SN-PCE could use even fewer observations because it identifies less than 30 orthonormal polynomials (6% of those used in naïve PCE) as necessary to approximate the network output of this particular process across all the 50 replications.

Table 2.3 and Table 2.4 show the estimated Sobol indices of S and C_p, respectively, using the four methods. As for S, SN-PCE yields similarly accurate estimates as the other methods for the influential inputs despite using fewer observations. As for C_p, PCE-based methods are not as accurate. This is an expected limitation of PCE because C_p is a discontinuous function of the inputs. Yet, the estimates are still close to those from the Monte Carlo sampling methods using 10,000 observations.

Fig. 2.9 and Fig. 2.10 show that Q, H_d, Z_v, and K_s are the most influential inputs for the variance of S and C_p. Note that in Fig. 2.10, the cumulative sum of the first-order Sobol indices approaches 90%, not 100%, implying that the interactions among the inputs have significant effects on the variance of C_p. It echoes the finding from Table 2.4 that the total Sobol indices for Q, Z_v, and K_s are significantly larger than the corresponding first-order Sobol indices.

Table 2.3: Sample means and standard errors (rounded to two decimal places) of the estimated first-order and total Sobol indices based on 50 replications for the maximal annual overflow S. For each replication, ‡‡SN-PCE uses 300 observations. All the other setups are the same as in Table 2.1.

Input	Distribution	Random Sampling†		Orthogonal Array‡		Naïve PCE††		SN-PCE‡‡	
		First-order	Total	First-order	Total	First-order	Total	First-order	Total
Q	$\mathscr{G}_{trunc}(1013, 558)$ on $[500, 3000]$	3.46×10^{-1} $\pm 0.02 \times 10^{-1}$	3.52×10^{-1} $\pm 0.03 \times 10^{-1}$	3.46×10^{-1} $\pm 0.02 \times 10^{-1}$	3.57×10^{-1} $\pm 0.03 \times 10^{-1}$	3.43×10^{-1} $\pm 0.02 \times 10^{-1}$	3.51×10^{-1} $\pm 0.02 \times 10^{-1}$	3.40×10^{-1} $\pm 0.03 \times 10^{-1}$	3.49×10^{-1} $\pm 0.03 \times 10^{-1}$
H_d	$U(7, 9)$	2.85×10^{-1} $\pm 0.01 \times 10^{-1}$	2.85×10^{-1} $\pm 0.03 \times 10^{-1}$	2.84×10^{-1} $\pm 0.02 \times 10^{-1}$	2.77×10^{-1} $\pm 0.04 \times 10^{-1}$	2.86×10^{-1} $\pm 0.02 \times 10^{-1}$	2.86×10^{-1} $\pm 0.02 \times 10^{-1}$	2.87×10^{-1} $\pm 0.02 \times 10^{-1}$	2.87×10^{-1} $\pm 0.02 \times 10^{-1}$
Z_v	$T(49, 50, 51)$	1.89×10^{-1} $\pm 0.01 \times 10^{-1}$	1.84×10^{-1} $\pm 0.04 \times 10^{-1}$	1.88×10^{-1} $\pm 0.01 \times 10^{-1}$	1.82×10^{-1} $\pm 0.04 \times 10^{-1}$	1.88×10^{-1} $\pm 0.01 \times 10^{-1}$	1.89×10^{-1} $\pm 0.01 \times 10^{-1}$	1.94×10^{-1} $\pm 0.04 \times 10^{-1}$	1.95×10^{-1} $\pm 0.04 \times 10^{-1}$
K_s	$N_{trunc}(30, 8)$ on $[15, \infty)$	1.34×10^{-1} $\pm 0.01 \times 10^{-1}$	1.45×10^{-1} $\pm 0.04 \times 10^{-1}$	1.34×10^{-1} $\pm 0.01 \times 10^{-1}$	1.43×10^{-1} $\pm 0.04 \times 10^{-1}$	1.34×10^{-1} $\pm 0.01 \times 10^{-1}$	1.42×10^{-1} $\pm 0.01 \times 10^{-1}$	1.30×10^{-1} $\pm 0.02 \times 10^{-1}$	1.38×10^{-1} $\pm 0.02 \times 10^{-1}$
C_b	$T(55, 55.5, 56)$	3.59×10^{-2} $\pm 0.02 \times 10^{-2}$	3.77×10^{-2} $\pm 0.43 \times 10^{-2}$	3.56×10^{-2} $\pm 0.02 \times 10^{-2}$	2.93×10^{-2} $\pm 0.47 \times 10^{-2}$	3.60×10^{-2} $\pm 0.03 \times 10^{-2}$	3.60×10^{-2} $\pm 0.03 \times 10^{-2}$	3.63×10^{-2} $\pm 0.05 \times 10^{-2}$	3.63×10^{-2} $\pm 0.05 \times 10^{-2}$
Z_m	$T(54, 55, 56)$	3.46×10^{-3} $\pm 0.02 \times 10^{-3}$	5.35×10^{-3} $\pm 4.66 \times 10^{-3}$	3.46×10^{-3} $\pm 0.02 \times 10^{-3}$	-8.27×10^{-4} $\pm 53.64 \times 10^{-4}$	3.40×10^{-3} $\pm 0.04 \times 10^{-3}$	3.75×10^{-3} $\pm 0.05 \times 10^{-3}$	3.39×10^{-3} $\pm 0.05 \times 10^{-3}$	3.75×10^{-3} $\pm 0.06 \times 10^{-3}$
B	$T(295, 300, 305)$	9.22×10^{-5} $\pm 0.06 \times 10^{-5}$	1.38×10^{-3} $\pm 4.73 \times 10^{-3}$	9.23×10^{-5} $\pm 0.06 \times 10^{-5}$	-4.76×10^{-3} $\pm 5.22 \times 10^{-3}$	9.33×10^{-5} $\pm 0.49 \times 10^{-5}$	1.29×10^{-4} $\pm 0.06 \times 10^{-4}$	1.09×10^{-4} $\pm 0.10 \times 10^{-4}$	1.71×10^{-4} $\pm 0.17 \times 10^{-4}$
L	$T(4900, 5000, 5010)$	3.38×10^{-7} $\pm 0.02 \times 10^{-7}$	1.32×10^{-3} $\pm 4.74 \times 10^{-3}$	3.32×10^{-7} $\pm 0.02 \times 10^{-7}$	-4.83×10^{-3} $\pm 5.20 \times 10^{-3}$	5.17×10^{-6} $\pm 0.67 \times 10^{-6}$	3.71×10^{-5} $\pm 0.31 \times 10^{-5}$	1.13×10^{-5} $\pm 0.19 \times 10^{-5}$	5.64×10^{-5} $\pm 0.56 \times 10^{-5}$

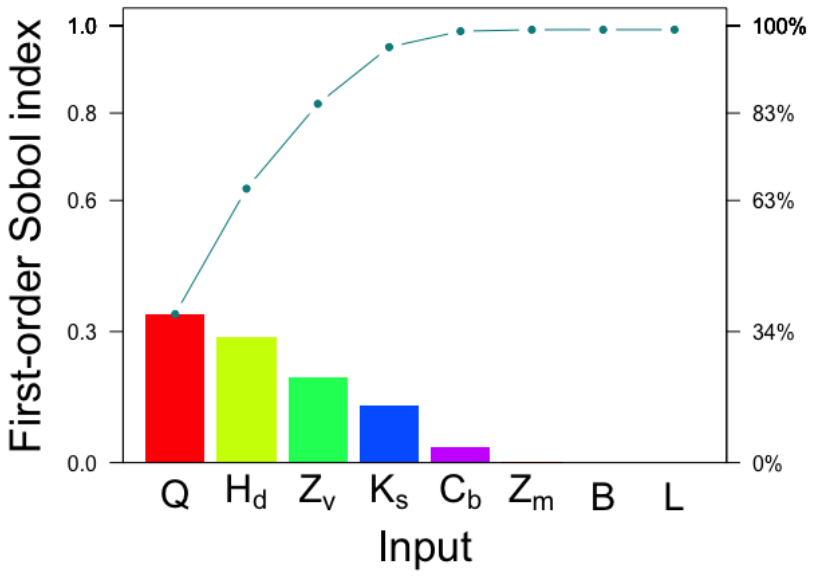


Figure 2.9: Pareto chart of the first-order Sobol indices of S estimated using SN-PCE for the flooding process. Higher bar indicates that the input has more influence (excluding interactions with other inputs) on the variance of the process output S.

2.6 Conclusion

This paper proposes SN-PCE to model uncertainty propagation in a broad class of processes represented as DAGs. A DAG encodes the dependencies between the variables in a process, including the process output (sink node in the DAG) and inputs (source nodes in the DAG). SN-PCE effectively captures how the inputs influence the output variance. Theoretically, it is shown that network PCE (equivalent to SN-PCE without sparsity) is valid in the sense that its prediction of the output converges in probability to the true output under reasonable assumptions. Empirically, SN-PCE is shown to accurately estimate the Sobol indices of the output with respect to the inputs for two manufacturing processes and a flooding process.

Table 2.4: Sample means and standard errors (rounded to two decimal places) of the estimated first-order and total Sobol indices based on 50 replications for the associated cost C_p. All the setups are the same as in Table 2.3.

Input	Distribution	Random Sampling†		Orthogonal Array‡		Naïve PCE††		SN-PCE‡‡	
		First-order	Total	First-order	Total	First-order	Total	First-order	Total
Q	$\mathscr{G}_{trunc}(1013, 558)$ on $[500, 3000]$	3.57×10^{-1} $\pm 0.03 \times 10^{-1}$	4.87×10^{-1} $\pm 0.04 \times 10^{-1}$	3.58×10^{-1} $\pm 0.03 \times 10^{-1}$	4.84×10^{-1} $\pm 0.03 \times 10^{-1}$	3.70×10^{-1} $\pm 0.02 \times 10^{-1}$	4.68×10^{-1} $\pm 0.03 \times 10^{-1}$	3.82×10^{-1} $\pm 0.04 \times 10^{-1}$	4.68×10^{-1} $\pm 0.05 \times 10^{-1}$
Z_v	$T(49, 50, 51)$	1.70×10^{-1} $\pm 0.01 \times 10^{-1}$	2.25×10^{-1} $\pm 0.06 \times 10^{-1}$	1.68×10^{-1} $\pm 0.01 \times 10^{-1}$	2.08×10^{-1} $\pm 0.06 \times 10^{-1}$	1.75×10^{-1} $\pm 0.02 \times 10^{-1}$	2.19×10^{-1} $\pm 0.02 \times 10^{-1}$	1.83×10^{-1} $\pm 0.03 \times 10^{-1}$	2.22×10^{-1} $\pm 0.04 \times 10^{-1}$
K_s	$N_{trunc}(30, 8)$ on $[15, \infty)$	1.58×10^{-1} $\pm 0.01 \times 10^{-1}$	2.56×10^{-1} $\pm 0.05 \times 10^{-1}$	1.57×10^{-1} $\pm 0.01 \times 10^{-1}$	2.58×10^{-1} $\pm 0.06 \times 10^{-1}$	1.61×10^{-1} $\pm 0.01 \times 10^{-1}$	2.29×10^{-1} $\pm 0.02 \times 10^{-1}$	1.68×10^{-1} $\pm 0.02 \times 10^{-1}$	2.41×10^{-1} $\pm 0.05 \times 10^{-1}$
H_d	$U(7, 9)$	1.18×10^{-1} $\pm 0.01 \times 10^{-1}$	1.88×10^{-1} $\pm 0.06 \times 10^{-1}$	1.20×10^{-1} $\pm 0.01 \times 10^{-1}$	1.64×10^{-1} $\pm 0.08 \times 10^{-1}$	1.22×10^{-1} $\pm 0.02 \times 10^{-1}$	1.69×10^{-1} $\pm 0.02 \times 10^{-1}$	1.29×10^{-1} $\pm 0.04 \times 10^{-1}$	1.29×10^{-1} $\pm 0.04 \times 10^{-1}$
C_b	$T(55, 55.5, 56)$	3.06×10^{-2} $\pm 0.03 \times 10^{-2}$	5.40×10^{-2} $\pm 0.69 \times 10^{-2}$	3.04×10^{-2} $\pm 0.02 \times 10^{-2}$	3.12×10^{-2} $\pm 0.07 \times 10^{-2}$	3.24×10^{-2} $\pm 0.07 \times 10^{-2}$	4.12×10^{-2} $\pm 0.13 \times 10^{-2}$	3.20×10^{-2} $\pm 0.12 \times 10^{-2}$	3.41×10^{-2} $\pm 0.01 \times 10^{-2}$
Z_m	$T(54, 55, 56)$	3.85×10^{-3} $\pm 0.04 \times 10^{-3}$	2.19×10^{-2} $\pm 0.68 \times 10^{-2}$	3.84×10^{-3} $\pm 0.05 \times 10^{-3}$	7.12×10^{-4} $\pm 78.32 \times 10^{-4}$	3.88×10^{-3} $\pm 0.17 \times 10^{-3}$	7.03×10^{-3} $\pm 0.04 \times 10^{-3}$	4.27×10^{-3} $\pm 0.16 \times 10^{-3}$	8.60×10^{-3} $\pm 0.05 \times 10^{-3}$
B	$T(295, 300, 305)$	9.84×10^{-5} $\pm 0.11 \times 10^{-5}$	1.47×10^{-2} $\pm 0.07 \times 10^{-2}$	9.77×10^{-5} $\pm 0.10 \times 10^{-5}$	-6.98×10^{-3} $\pm 7.88 \times 10^{-3}$	2.90×10^{-4} $\pm 0.04 \times 10^{-4}$	1.43×10^{-3} $\pm 0.19 \times 10^{-3}$	1.94×10^{-4} $\pm 0.25 \times 10^{-4}$	1.86×10^{-3} $\pm 0.24 \times 10^{-3}$
L	$T(4900, 5000, 5010)$	3.63×10^{-7} $\pm 0.04 \times 10^{-7}$	1.44×10^{-2} $\pm 0.07 \times 10^{-2}$	3.55×10^{-7} $\pm 0.04 \times 10^{-7}$	-7.46×10^{-3} $\pm 7.87 \times 10^{-3}$	2.11×10^{-4} $\pm 0.03 \times 10^{-4}$	1.28×10^{-3} $\pm 0.01 \times 10^{-3}$	1.28×10^{-4} $\pm 0.25 \times 10^{-4}$	2.23×10^{-3} $\pm 0.46 \times 10^{-3}$

SN-PCE uses substantially fewer observations than the black-box approaches (Monte Carlo sampling methods and naïve PCE) to accurately identify the influential inputs, showing promise for efficient sensitivity analysis in process automation.

Future research may investigate extension of the proposed model to even more general networks. One direction could be to relax the assumption of mutual independence of the network inputs. While their dependencies would complicate the estimation and interpretation of the sensitivity indices, the indices proposed in [39] might help decode how the dependent network inputs influence the output. Another research direction would be to allow cycles (or, feedback loops) in the network, thereby, extending this work beyond directed acyclic networks. Lastly, while domain experts can often identify DAGs of their systems, machine learning can more efficiently identify complex DAGs from data under the causal Markov and causal faithfulness conditions [26, 44]. Future research may investigate the best way to combine network identification methods with the proposed method in this chapter.

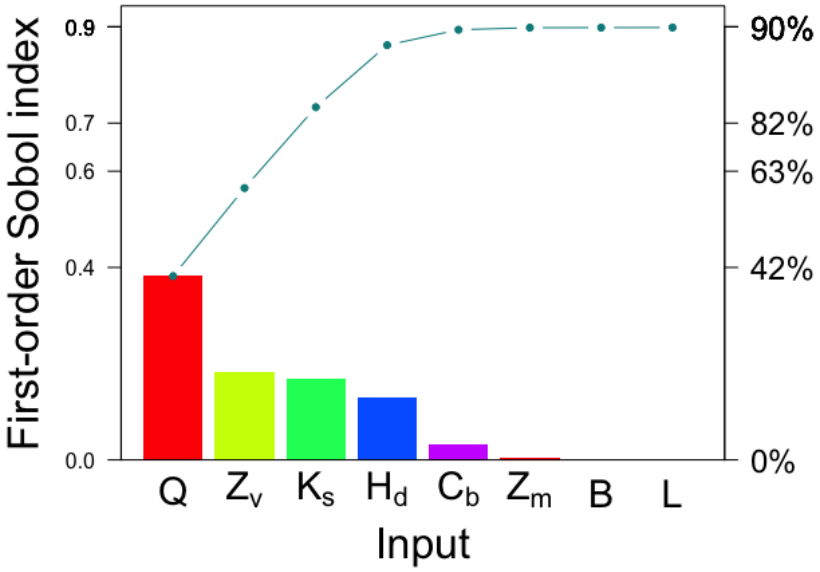


Figure 2.10: Pareto chart of the first-order Sobol indices of C_p estimated using SN-PCE for the flooding process. Higher bar indicates that the input has more influence (excluding interactions with other inputs) on the variance of the process output C_p.

Chapter 3

DATA-DRIVEN SENSITIVITY INDICES FOR MODELS WITH DEPENDENT INPUTS USING THE POLYNOMIAL CHAOS EXPANSION

3.1 Abstract

Uncertainties exist in both physics-based and data-driven models. Variance-based sensitivity analysis characterizes how the variance of a model output is propagated from the model inputs. The Sobol index is one of the most widely used sensitivity indices for models with independent inputs. For models with dependent inputs, different approaches have been explored to obtain sensitivity indices in the literature. Typical approaches are based on procedures of transforming the dependent inputs into independent inputs. However, such transformation requires additional information about the inputs, such as the dependency structure or the conditional probability density functions. In this chapter, data-driven sensitivity indices are proposed for models with dependent inputs. We first construct ordered partitions of linearly independent polynomials of the inputs. The modified Gram-Schmidt algorithm is then applied to the ordered partitions to generate orthogonal polynomials with respect to the empirical measure based on observed data of model inputs and outputs. Using the polynomial chaos expansion with the orthogonal polynomials, we obtain the proposed data-driven sensitivity indices. The sensitivity indices provide intuitive interpretations of how the dependent inputs affect the variance of the output without a priori knowledge on the dependence structure of the inputs. Four numerical examples are used to validate the proposed approach.

3.2 Introduction

Uncertainties exist in both physics-based and data-driven models. Uncertainty quantification (UQ) methods to characterize and reduce those uncertainties are increasingly popular in engineering studies. As an aspect of UQ, sensitivity analysis (SA) quantifies how output uncertainties are propagated from input uncertainties. Two general ways of conducting SA are local sensitivity analysis (LSA) and global sensitivity analysis (GSA). LSA analyzes how a small perturbation near an input space value could influence the output. On the contrary, GSA investigates how the input variability influences the output variability over the entire input space. In recent studies, variance-based sensitivity analysis, as a form of GSA, is utilized to understand system uncertainties in various applications such as material mechanics [32], building energy [57], structural mechanics [73], hydrogeology [16], and manufacturing [19].

Conducting variance-based sensitivity analysis for models with *independent* inputs has been studied widely. Monte Carlo simulation and surrogate models are two general ways to obtain sensitivity indices for models with independent inputs. Surrogate models have been shown to be more computationally efficient compared with Monte Carlo simulation [64]. Polynomial chaos expansion (PCE) and Kriging (also known as Gaussian process regression) are the two surrogate models which have been used to compute sensitivity indices most commonly in the literature [36, 64]. Thanks to the orthogonal property of a PCE model, sensitivity indices for *independent* inputs can be directly obtained using PCE coefficients [36, 60, 62]. PCE-based sensitivity indices appear in various fields including fluid dynamics [70], structural reliability [42], and vehicle dynamics [34].

For models with *dependent* inputs, a limited number of approaches are available in the literature to conduct variance-based sensitivity analyses. Generalized Sobol sensitivity indices have been proposed in Chastaing et al. [12] based on the hierarchically orthogonal functional decomposition (HOFD). However, the unboundedness of the resulting sensitivity indices makes their interpretation for *dependent* inputs not as straightforward as the

Sobol indices for models with *independent* inputs [48]. A different framework is proposed in [74] to obtain sensitivity indices for models with correlated inputs. However, it requires the knowledge of model structure between the inputs and the outputs. An alternative way of obtaining sensitivity indices for models with *dependent* inputs is to transform *dependent* inputs into *independent* inputs [40, 41, 65]. Even though the transformation-based methods generate interpretable sensitivity indices, they require strong assumptions on the dependency or distributions of the inputs.

The main contribution of this paper is the development of a data-driven method to obtain interpretable sensitivity indices for models with dependent inputs without invoking any assumptions on the inputs. We first propose the modified Gram-Schmidt based polynomial chaos expansion (mGS-PCE). The mGS-PCE increases the numerical robustness of constructing orthogonal polynomials for arbitrarily distributed inputs compared with the GS-PCE in [70]. Then, we propose a method to obtain data-driven sensitivity indices for models with dependent inputs by constructing ordered partitions of orthonormal polynomials of the inputs. This method estimates some of the sensitivity indices in [40] and [41] without invoking the assumptions therein. Lastly, we propose conditional order-based sensitivity indices, which explain the model output variability in a hierarchical manner.

The remainder of the paper is organized as follows: Section 3.3 reviews the background knowledge about Sobol indices and PCE models. Section 3.4 introduces the modified Gram-Schmidt algorithm and our data-driven method to obtain sensitivity indices for models with dependent inputs using PCE models. In Section 3.5, four numerical examples, where the inputs are dependent, are used to validate our proposed method. Section 3.6 provides a few concluding remarks and a discussion on future research directions.

3.3 Technical background

This section briefly reviews sensitivity indices in the existing literature for models with independent inputs and those with dependent inputs. We first introduce the Hoeffding functional decomposition and the Sobol indices for independent inputs. We then review

the *full* sensitivity indices and the *uncorrelated* sensitivity indices defined for models with *dependent* inputs. Lastly, we introduce PCE models and explain how PCE coefficients can be used to calculate sensitivity indices.

3.3.1 Hoeffding decomposition and sensitivity indices for independent inputs

Suppose we have n independent random inputs $\boldsymbol{X} = (X_1, X_2, \cdots, X_n)$ with their density $\mu(\boldsymbol{X})$. For the output $Y = f(\boldsymbol{X})$ that is square-integrable with respect to $\mu(\boldsymbol{X})$, its Hoeffding decomposition is defined as follows [12, 59]:

$$f(\boldsymbol{X}) = \sum_{u \subseteq \{1,2,\ldots,n\}} f_u(\boldsymbol{X}_u), \tag{3.1}$$

where $f_\emptyset = f_0$ and f_0 is a constant and $\boldsymbol{X}_u = (X_j)_{j \in u}$. Each summand $f_u(\boldsymbol{X}_u), u \neq \emptyset$, in Eq. (3.2) satisfies

$$\int f_u(\boldsymbol{X}_u)\mu(X_i)\,\mathrm{d}X_i = 0, \quad \forall i \in u.$$

such that

$$f_0 = \int f(\boldsymbol{X})\mu(\boldsymbol{X})\,\mathrm{d}\boldsymbol{X}.$$

In addition, the summands in Eq. (3.2) are orthogonal to each other as follows:

$$\int f_u(\boldsymbol{X}_u) f_v(\boldsymbol{X}_v)\mu(\boldsymbol{X})\,\mathrm{d}\boldsymbol{X} = 0, \quad \forall u, v \subseteq \{1, 2, \ldots, n\}, v \neq u.$$

Based on the Hoeffding decomposition, the variance of Y is decomposed as follows [60, 62]:

$$\begin{aligned} Var(Y) &= \int f^2(\boldsymbol{X})\mu(\boldsymbol{X})\,\mathrm{d}\boldsymbol{X} - f_0^2 \\ &= \sum_{\substack{u \subseteq \{1,2,\ldots,n\} \\ u \neq \emptyset}} D_u(Y), \end{aligned}$$

where

$$\begin{aligned} D_u(Y) &= \int f_u^2(\boldsymbol{X}_u)\mu(\boldsymbol{X}_u)\,\mathrm{d}\boldsymbol{X}_u \\ &= Var(E(Y \mid \boldsymbol{X}_u)) - \sum_{\substack{v \subset u \\ v \neq u \\ v \neq \emptyset}} D_v(Y). \end{aligned}$$

For example, $D_i(Y) = Var(E(Y \mid X_i))$ and $D_{ij}(Y) = Var(E(Y \mid X_i, X_j)) - D_i(Y) - D_j(Y)$.

Based on the variance decomposition, the Sobol index for set u is defined as

$$S_u = \frac{D_u(Y)}{Var(Y)},$$

which measures the sensitivity of the output variance with respect to the inputs in $\boldsymbol{X}_u$. For a particular input variable X_i, the *first-order* Sobol index S_{X_i} and *total* Sobol index ST_{X_i} are defined as follows:

$$S_{X_i} = \frac{D_i(Y)}{Var(Y)},$$

$$ST_{X_i} = \sum_{u \ni i} S_u.$$

S_{X_i} represents the percentage of the output variance that is propagated from the input X_i. ST_{X_i} represents the percentage of the output variance that is propagated from the input X_i and its interactions with the other variables.

3.3.2 Sensitivity indices for dependent inputs

This study focuses on sensitivity indices proposed in [40, 41, 65] because they are bounded and do not require the knowledge of the model structure between the inputs and the output in contrast to those considered in [12, 35, 74], as discussed earlier.

In [40], the Gram-Schmidt algorithm is employed to decorrelate the inputs when the dependences are characterized solely by the inputs' first-order conditional moments. Then the *full* sensitivity indices and the *uncorrelated* sensitivity indices (also called *independent* sensitivity indices in [41]) are defined. On the other hand, in order to calculate these sensitivity indices when conditional probability density functions (cPDFs) of the inputs are known, the inverse Rosenblatt transformation or the inverse Nataf transformation is applied to transform the *dependent* inputs into the *independent* inputs [41, 65].

Suppose dependent inputs $(X_1, X_2, \ldots, X_n)$ are transformed into independent inputs $(\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_n)$, for example, under the assumptions of [40] such that $\bar{X}_1 = X_1$ and $\bar{X}_i = X_i - E\Big(X_i \mid \bar{X}_1, \ldots, \bar{X}_{i-1}\Big)$, $\forall i = 2, \ldots, n$. Intuitively speaking, $\bar{X}_1$ keeps all information concerning

X_1 including its dependent part with the other inputs. $\bar{X}_i$ contains all information concerning X_i *except* its dependent part with $\bar{X}_1, \ldots, \bar{X}_{i-1}$. Thus, $\bar{X}_n$ only contains information of X_n *excluding* its dependent part with all the other inputs. These constructed *independent* inputs allow for calculating the first-order Sobol indices ($S_{\bar{X}_i}$) and total Sobol indices ($ST_{\bar{X}_i}$). Then the sensitivity indices with respect to the *dependent* inputs are defined as follows [40]:

$\bar{S}_{X_1} = S_{\bar{X}_1}$ is the first-order full contribution of X_1 to the variance of the output.

$\overline{ST}_{X_1} = ST_{\bar{X}_1}$ is the total full contribution of X_1 to the variance of the output.

$S^u_{X_n} = S_{\bar{X}_n}$ is the first-order uncorrelated contribution of X_n to the variance of the output.

$ST^u_{X_n} = ST_{\bar{X}_n}$ is the total uncorrelated contribution of X_n to the variance of the output.

By permuting the order of the inputs, different sensitivity indices can be further calculated. Suppose the initial input variables are ordered as $(X_i, X_{i+1}, \ldots, X_n, X_1, \ldots, X_{i-1})$, and the constructed independent inputs are $(\bar{X}_i, \bar{X}_{i+1}, \ldots, \bar{X}_n, \bar{X}_1, \ldots, \bar{X}_{i-1})$. Then the *full* sensitivity indices ($\bar{S}_{X_i} = S_{\bar{X}_i}$ and $\overline{ST}_{X_i} = ST_{\bar{X}_i}$) and the *uncorrelated* sensitivity indices ($S^u_{X_{i-1}} = S_{\bar{X}_{i-1}}$ and $ST^u_{X_i} = ST_{\bar{X}_{i-1}}$) are defined. $\bar{S}_{X_i}$ is called the first-order *full* sensitivity index and $\overline{ST}_{X_i}$ is called the total *full* sensitivity index. $S^u_{X_i}$ is called the first-order *uncorrelated* sensitivity index and $ST^u_{X_i}$ is called the total *uncorrelated* sensitivity index.

3.3.3 PCE and PCE-based sensitivity indices

As a way of calculating sensitivity indices, PCE is known to be more computationally efficient than Monte Carlo simulations [60, 62]. The original PCE, which is proposed in [69], provides Hermite polynomials for independent Gaussian random variables. Several types of PCE have been proposed under the assumption of *independence* between model inputs, including the generalized PCE (gPCE) [72], the multi-element generalized PCE (ME-gPCE) [68], the moment-based arbitrary PCE (aPCE) [50] and the Gram-Schmidt based PCE (GS-PCE) [70].

The GS-PCE for models with *independent* inputs is extended to models with multivariate *dependent* inputs in Navarro et al. [48]. It is regarded as the pioneering work in constructing an orthogonal polynomial basis for arbitrary *dependent* inputs. Rahman [55] theoretically validates the Gram-Schmidt orthogonalization process to construct an orthogonal polynomial basis for the PCE with dependent inputs.

PCE model

PCE uses a finite number of orthonormal polynomial terms of n random inputs in $\boldsymbol{X}$ to approximate the output Y as follows:

$$Y = f(\boldsymbol{X}) \approx \sum_{i=0}^{P} \theta_i \psi_i(\boldsymbol{X}), \tag{3.2}$$

where θ_i, $i = 0,1,2,\ldots,P$, are called PCE coefficients and ψ_i, $i = 1,2,\ldots,P$ are orthonormal polynomials.

$$P+1 = \binom{n+p}{n} \tag{3.3}$$

is the number of polynomial terms, where p is the highest polynomial degree in the PCE model. As p increases, the accuracy of approximating a complex output function improves. In this chapter, we estimate the PCE coefficients by solving an overdetermined linear system of equations in the least-squares sense as proposed in [4].

Thanks to the properties of orthonormal polynomials, we can approximate the lower order moments of output Y directly using the PCE coefficients in (3.2) as follows:

$$\begin{aligned} E(Y) &\approx \theta_0, \\ Var(Y) &\approx \sum_{i=1}^{P} \theta_i^2. \end{aligned} \tag{3.4}$$

The approximation errors converge to zero as P increases [10].

PCE-based sensitivity indices

For *independent* inputs, the multivariate orthonormal polynomials $\psi_i(\boldsymbol{X})$ can be directly constructed as the products of univariate orthonormal polynomials as follows [36, 60, 62]:

$$\psi_i(\boldsymbol{X}) = \psi_{\boldsymbol{\alpha}_i}(\boldsymbol{X}) = \prod_{j=1}^{n} \psi_{\alpha_{ij}}(X_j),$$

where $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in})$ and $\psi_{\alpha_{ij}}(X_j)$ represents the $\alpha_{ij}-$th order orthonormal polynomial in input X_j.

Define $\mathscr{A}_u$ as the set of multi-indices depending exactly on the subset of variables $\boldsymbol{X}_u, u \subseteq \{1, 2, \ldots, n\}$ as follows:

$$\mathscr{A}_u = \{\boldsymbol{\alpha}_i \in \mathbb{N}^n : \alpha_{ij} \neq 0 \Leftrightarrow j \in u, |\boldsymbol{\alpha}_i| \leq p\},$$

where

$$|\boldsymbol{\alpha}_i| = \sum_{j=1}^{n} \alpha_{ij}.$$

Suppose $\theta_{\boldsymbol{\alpha}_i}$ is the PCE coefficient with respect to the polynomial term corresponding to $\boldsymbol{\alpha}_i$. Then the first-order Sobol index for X_j and the total Sobol index for X_j can be estimated for $j = 1, \ldots, n$ as follows [36, 60, 62]:

$$S_{X_j} \approx \frac{\sum_{\boldsymbol{\alpha}_i \in \mathscr{A}_{\{j\}}} \theta^2_{\boldsymbol{\alpha}_i}}{\sum_{i=1}^{P} \theta_i^2},$$

$$ST_{X_j} \approx \sum_{\mathscr{A}_{u \ni j}} S_{\mathscr{A}_u},$$

where

$$S_{\mathscr{A}_u} \approx \frac{\sum_{\boldsymbol{\alpha}_i \in \mathscr{A}_u} \theta^2_{\boldsymbol{\alpha}_i}}{\sum_{i=1}^{P} \theta_i^2}.$$

Using these PCE-based Sobol indices, we can also obtain the sensitivity indices for *dependent* inputs, which were described earlier.

3.4 Methodology

In the previous section, we discussed the current methods proposed in [40] and [41] of obtaining the sensitivity indices for models with *dependent* inputs under certain assumptions on the inputs.

In this section, we propose a data-driven method to estimate the sensitivity indices for models with *dependent* inputs using a PCE model based on the orthonormal polynomials constructed from the modified Gram-Schmidt algorithm. First, we show how to construct orthonormal polynomials using the modified Gram-Schmidt algorithm. Then we propose a data-driven method to estimate the *first-order full* sensitivity indices and the *total uncorrelated* sensitivity indices for models with *dependent* inputs. Then we propose an alternative *total full* sensitivity index and an alternative *first-order uncorrelated* sensitivity index, which can be also calculated using the proposed method. These alternative indices have different interpretations than those in [40] and [41] because our decorrelation process does not eliminate dependences in inputs. In addition, we propose conditional order-based sensitivity indices and illustrate how they can be used to reduce the PCE model complexity by excluding higher order interaction terms.

3.4.1 Modified Gram-Schmidt algorithm

In [48], orthonormal polynomials are constructed using the Gram-Schmidt algorithm for general multivariate correlated variables. Even though the Gram-Schmidt algorithm behaves the same as the modified Gram-Schmidt algorithm mathematically, the modified Gram-Schmidt algorithm is less sensitive to numeric rounding errors and performs more stably than the Gram-Schmidt algorithm [5]. Therefore, we propose to use the modified Gram-Schmidt algorithm to construct orthonormal polynomial basis $\{\psi_i(\boldsymbol{X})\}_{i=1}^{P}$ based on the initial P linearly independent polynomials $(e_i)_{i\in\{1,2,\ldots,P\}}$ as follows [6]:

Algorithm 3 Modified Gram-Schmidt Algorithm

1: **for** $i = 1, 2, \ldots, P$ **do**
2: $\quad \phi_i(\boldsymbol{X}) \leftarrow e_i(\boldsymbol{X})$
3: $\quad$ **for** $k = 1, 2, \ldots, i-1$ **do**
4: $\quad\quad \phi_i(\boldsymbol{X}) \leftarrow \phi_i(\boldsymbol{X}) - \langle \phi_i(\boldsymbol{X}), \psi_k(\boldsymbol{X}) \rangle \psi_k(\boldsymbol{X})$
5: $\quad$ **end for**
6: $\quad \psi_i(\boldsymbol{X}) \leftarrow \frac{\phi_i(\boldsymbol{X})}{||\phi_i(\boldsymbol{X})||_2}$
7: **end for**

The inner product in the algorithm is defined with respect to the empirical measure in this chapter. The inner product is numerically evaluated using the observations of $\boldsymbol{X}$ in a given dataset. Note that the proposed data-driven method assumes neither any distributional knowledge of $\boldsymbol{X}$ nor the ability to easily sample from its distribution. Thus, we do not use a Monte Carlo approach to evaluate the inner product although it may be an option for the problems that permit the sampling.

The difference between the standard Gram-Schmidt algorithm and the modified Gram-Schmidt algorithm is at the line 4 in Algorithm 1, where the standard Gram-Schmidt algorithm performs

$$\phi_i(\boldsymbol{X}) \leftarrow \phi_i(\boldsymbol{X}) - \langle e_i(\boldsymbol{X}), \psi_k(\boldsymbol{X}) \rangle \, \psi_k(\boldsymbol{X}).$$

Note that different orthonormal polynomials are constructed from different permutations of the initial polynomials. In the following section, we discuss how to permute the order of the initial polynomials in order to obtain data-driven sensitivity indices for models with *dependent* inputs.

3.4.2 Sensitivity indices

As we discussed in the previous section, PCE models can be constructed for models with *dependent* inputs based on the modified Gram-Schmidt algorithm. In this section, we first propose how to use PCE models to estimate the *full* sensitivity indices and the *uncorrelated*

sensitivity indices based on data. Then we define the conditional order-based sensitivity indices and present how they can be used to exclude higher order interaction terms in a PCE model. For easy reference, we include in Appendix A.1 a list of sensitivity index symbols used in this chapter.

Full sensitivity indices

Constructing orthonormal polynomials using the modified Gram-Schmidt algorithm requires a linearly independent set of polynomials. A PCE model with n inputs and the highest polynomial order p is composed of $P+1$ terms of polynomials as we defined in Eqs. (3.2) and (3.3). Assume polynomials in the set

$$S = \left\{ \prod_{l=1}^{n} X_l^{j_l} : j_l \in \{0,1,\dots p\}, \sum_{l=1}^{n} j_l \leq p \right\} \tag{3.5}$$

are linearly independent.

Orthonormal polynomials can be constructed using the modified Gram-Schmidt algorithm with respect to a specific order of the polynomials. Suppose we order the polynomials in S as $(St_0, St_{11}, St_1 \backslash St_{11}, St_2, St_3, \dots, St_n)$, where $St_0 = \{1\}$, St_{11} and St_i are defined as follows:

$$\begin{aligned} St_{11} &= \left\{ X_1^{j_1} : j_1 \in \{1,\dots p\} \right\}, \\ St_i &= \left(S \backslash \bigcup_{j=0}^{i-1} St_j \right) \bigcap \left\{ \prod_{l=1}^{n} X_l^{j_l} : j_i \in \{1,2,\dots,p\}, j_{l\neq i} \in \{0,1,\dots,p\}, \sum_{l=1}^{n} j_l \leq p \right\}. \end{aligned} \tag{3.6}$$

$(St_0, St_{11}, St_1 \backslash St_{11}, St_2, St_3, \dots, St_n)$ is an ordered partition of the set S. In the partition, St_0, St_{11}, $St_1 \backslash St_{11}$, and $St_i, i = 2,3,\dots,n$ are ordered in sequence but the polynomials in each set can be in any arbitrary order. Note that St_1 contains all the polynomial functions of X_1 and the interaction terms between X_1 and the rest of the inputs. St_2 contains all the polynomial functions of X_2 and the interaction terms between X_2 and the rest of the inputs *except* X_1. St_3 contains all the polynomial functions of X_3 and the interaction terms between X_3 and the rest of the inputs *except* X_1 and X_2.

For example, St_{11} and $St_i, i=1,2,3$ for constructing a PCE model with inputs $\{X_1, X_2, X_3\}$ and the highest polynomial order $p=3$ are defined as follows:

$$St_{11} = \left\{X_1, X_1^2, X_1^3\right\},$$
$$St_1 = \left\{X_1, X_1^2, X_1^3, X_1X_2, X_1^2X_2, X_1X_2^2, X_1X_3, X_1^2X_3, X_1X_3^2, X_1X_2X_3\right\},$$
$$St_2 = \left\{X_2, X_2^2, X_2^3, X_2X_3, X_2^2X_3, X_2X_3^2\right\},$$
$$St_3 = \left\{X_3, X_3^2, X_3^3\right\}.$$

After constructing the orthonormal polynomials using the ordered partition (St_0, St_{11},-$St_1 \backslash St_{11}, St_2, St_3, \ldots, St_n$), the first-order full sensitivity index $\bar{S}_{X_1}$ for X_1 can be estimated as follows:

$$\bar{S}_{X_1} \approx \frac{\sum_{j \in St_{11}} \theta_j^2}{Var(Y)}, \tag{3.7}$$

where θ_j's are the PCE coefficients corresponding to the orthonormal polynomials in the set St_{11}.

In addition, we propose an alternative total full sensitivity index

$$\overline{ST}_{X_1} = \frac{\sum_{u \ni X_1} D_u(Y)}{Var(Y)}$$

and estimate it using

$$\overline{ST}_{X_1} \approx \frac{\sum_{j \in St_1} \theta_j^2}{Var(Y)}. \tag{3.8}$$

This total full sensitivity index is different from the one defined in [40], which is obtained after transforming the dependent inputs into the independent inputs. Instead, the total full sensitivity index in Eq. (3.8) has dependent effects of X_1 with other inputs. By permuting the order of the input $(X_1, X_2, X_3, \ldots, X_n)$ as $(X_i, X_{i+1}, \ldots, X_n, X_1, \ldots, X_{i-1})$, any $\bar{S}_{X_i}$ and $\overline{ST}_{X_i}$ can be estimated.

We also define the conditional total sensitivity indices for models with *dependent* inputs as follows:

- $\overline{ST}_{X_2|X_1} = \frac{\sum_{u \ni \{X_1, X_2\}} D_u(Y) - \sum_{u \ni X_1} D_u(Y)}{Var(Y)}$ is the total contribution of input X_2 to the variance of output Y after taking account of the total full contribution of X_1.

$\overline{ST}_{X_3|X_1,X_2} = \frac{\sum_{u \ni \{X_1,X_2,X_3\}} D_u(Y) - \sum_{u \ni \{X_1,X_2\}} D_u(Y)}{Var(Y)}$ is the total contribution of input X_3 to the variance of output Y after taking account of the total full contributions of X_1 and X_2.

$\vdots$

$\overline{ST}_{X_n|X_1,X_2,\ldots,X_{n-1}} = \frac{\sum_{u \ni \{X_1,X_2,\ldots,X_n\}} D_u(Y) - \sum_{u \ni \{X_1,X_2,\ldots,X_{n-1}\}} D_u(Y)}{Var(Y)}$ is the total contribution of input X_n to the variance of output Y after taking account of the total full contributions of X_1, X_2, ..., X_{n-1}.

We estimate the conditional total sensitivity indices using

$$\overline{ST}_{X_i|X_1,X_2,\ldots,X_{i-1}} \approx \frac{\sum_{j \in St_i} \theta_j^2}{Var(Y)}$$

for $i = 2, 3, \ldots, n$

When inputs can be grouped such that inputs from different groups have neither dependence nor interaction across groups, the total Sobol index of a group can be estimated based on the conditional total sensitivity indices. For example, if the first d inputs are independent of and have no interactions with the rest of the inputs, $\sum_{i=1}^{d} \overline{ST}_{X_i}$ estimates the total Sobol index of the first d inputs. Eq. (3.10) for Example 3 in Section 3.5 illustrates how this sensitivity index can be used in practice.

Uncorrelated sensitivity indices

In order to estimate the total *uncorrelated* sensitivity index of X_1, we consider the ordered partition of S as $(St_0, St_{-1}, St_{11}, St_1 \backslash St_{11})$, where $St_0 = \{1\}$ and $St_{-1} = \bigcup_{j=2}^{n} St_j$. St_{11} and $St_i, i = 1, 2, \ldots, n$ are defined in Eq. (3.6). Then the orthonormal polynomials can be constructed with respect to this ordered partition. As we obtain the PCE coefficients corresponding to the orthonormal polynomials in the set St_1, the total *uncorrelated* sensitivity index ($ST_{X_1}^u$) can be estimated using Eq. (3.8).

In addition, we propose an alternative first-order *uncorrelated* sensitivity index ($S_{X_1}^u$) and estimate it using Eq. (3.7). The proposed first-order *uncorrelated* sensitivity index is

different from the one defined in [40] because the latter is estimated after decorrelating X_1 with all the other inputs. Note that the proposed first-order *uncorrelated* sensitivity index is estimated by decorrelating the polynomials of the inputs. By permuting the order of the inputs $(X_1, X_2, X_3, \ldots, X_n)$ as $(X_i, X_{i+1}, \ldots, X_n, X_1, \ldots, X_{i-1})$, any $S^u_{X_i}$ and $ST^u_{X_i}$ can be estimated.

Note that the first-order full (resp. uncorrelated) sensitivity indices are always smaller or equal to the total full (resp. uncorrelated) sensitivity indices, but the first-order full (resp. uncorrelated) sensitivity indices are not necessarily larger or smaller than the total uncorrelated (resp. full) sensitivity indices.

Conditional order-based sensitivity indices

In order to reduce the complexity of a PCE model and select appropriate interaction terms in the PCE model, we propose the conditional order-based sensitivity indices.

Suppose we order the polynomials in the polynomial set S in Eq. (3.5) as $(Sc_0, Sc_{11}, Sc_{12}, \ldots, Sc_{1p}, Sc_{22}, Sc_{23}, \ldots, Sc_{2p}, \ldots, Sc_{kk}, Sc_{kk+1}, \ldots, Sc_{kp}\Big)$, where $k = \min(n,p)$, $Sc_0 = \{1\}$, and $Sc_{ij}, i = 1, 2, \ldots, k;\ j = i, i+1, \ldots, p$, are defined as follows:

$$Sc_{ij} = \left\{ \prod_{l=1}^{n} X_l^{j_l} : j_l \in \{0, 1, \ldots p\}, \sum_{l=1}^{n} \mathbb{1}_{j_l \neq 0} = i, \sum_{l=1}^{n} j_l = j \right\},$$

where

$$\mathbb{1}_{j_l \neq 0} = \begin{cases} 1 & j_l \neq 0 \\ 0 & j_l = 0 \end{cases}.$$

Define $Sc_i = \cup_{j=i}^{p} Sc_{ij}, i \leq \min(n,p)$, then Sc_1 contains all the polynomial functions of $X_i, i = 1, 2, \ldots, n$. Sc_2 contains all the two-way interaction terms. Sc_3 contains all the three-way interaction terms. Note that $(Sc_0, Sc_{11}, Sc_{12}, \ldots, Sc_{1p}, Sc_{22}, Sc_{23}, \ldots, Sc_{2p}, \ldots, Sc_{kk}, Sc_{kk+1}, \ldots, Sc_{kp}\Big)$, where $k = \min(n,p)$, is an ordered partition of the set S. In the partition, $Sc_0, Sc_{11}, Sc_{12}, \ldots, Sc_{1p}, Sc_{22}, Sc_{23}, \ldots, Sc_{2p}, \ldots, Sc_{kk}, Sc_{kk+1}, \ldots, Sc_{kp}$, are ordered in sequence but the polynomials in each set Sc_{ij} can be in any arbitrary order.

For example, $Sc_{11}, Sc_{12}, Sc_{13}, Sc_{22}, Sc_{23}$, and Sc_{33} for constructing a PCE model with inputs $\{X_1, X_2, X_3\}$ and the highest polynomial order $p = 3$ are defined as follows:

$$
\begin{aligned}
Sc_{11} &= \{X_1, X_2, X_3\}, \\
Sc_{12} &= \left\{X_1^2, X_2^2, X_3^2\right\}, \\
Sc_{13} &= \left\{X_1^3, X_2^3, X_3^3\right\}, \\
Sc_{22} &= \{X_1X_2, X_2X_3, X_1X_3\}, \\
Sc_{23} &= \left\{X_1^2X_2, X_1^2X_3, X_1X_2^2, X_1X_3^2, X_2^2X_3, X_2X_3^2, \right\}, \\
Sc_{33} &= \{X_1X_2X_3\}.
\end{aligned}
$$

We define the conditional order-based sensitivity indices as follows:

$\tilde{S}_1 = \frac{\sum_{i=1}^{n} D_i(Y)}{Var(Y)}$ is the first order sensitivity index of the output Y with respect to the inputs $\boldsymbol{X}$.

$\tilde{S}_{2|1} = \frac{\sum_{i<j} D_{ij}(Y)}{Var(Y)}$ is the second order sensitivity index of the output Y with respect to the inputs $\boldsymbol{X}$ after taking account of the first order sensitivity index.

$\tilde{S}_{3|1,2} = \frac{\sum_{i<j<k} D_{ijk}(Y)}{Var(Y)}$ is the third order sensitivity index of the output Y with respect to the inputs $\boldsymbol{X}$ after taking account of the first order sensitivity index and the second order sensitivity index.

$\vdots$

$\tilde{S}_{k|1,2,\ldots,k-1} = \frac{D_{1,2,3,\ldots,k}(Y)}{Var(Y)}$ is the k^{th} order sensitivity index of the output Y with respect to the inputs $\boldsymbol{X}$ after taking account of the first order sensitivity index through the $(k-1)^{th}$ order sensitivity index.

As we can obtain the PCE coefficients corresponding to the orthonormal polynomials constructed from $(Sc_0, Sc_{11}, Sc_{12}, \ldots, Sc_{1p}, Sc_{22}, Sc_{23}, \ldots, Sc_{2p}, \ldots,$

$Sc_{kk}, Sc_{kk+1}, \ldots, Sc_{kp}\Big)$, where $k = \min(n,p)$, using the modified Gram-Schmidt algorithm, the above sensitivity indices can be estimated as follows:

$$\tilde{S}_1 \approx \frac{\sum_{j \in Sc_1} \theta_j^2}{Var(\boldsymbol{Y})},$$
$$\tilde{S}_{i|1,\ldots,i-1} \approx \frac{\sum_{j \in Sc_i} \theta_j^2}{Var(\boldsymbol{Y})},\ i = 2, \ldots, \min(n,p),$$

where θ_j's are the PCE coefficients corresponding to the orthonormal polynomials in the set Sc_i for $i \leq \min(n,p)$. Note that for a full PCE model, Sc_i contains $\binom{n}{i}\binom{p}{i}$ polynomial terms.

The conditional order-based sensitivity indices serve the purpose of identifying up to which *order of interaction* of inputs significantly influences the output variance. Specifically, if the cumulative sum of conditional order-based sensitivity indices, $\sum_{i=1}^{d} \tilde{S}_i$, is close to one for a certain polynomial order, $d \leq \min(n,p)$, it indicates that the interaction terms of orders higher than d can be excluded in the PCE model. Using a simple procedure of inspecting the cumulative sum for different d's, we can identify and remove unnecessary high-order interaction terms from the PCE model. In constrast to the existing methods of constructing a sparse PCE model [7, 9, 31, 52], this procedure keeps the effect hierarchy principle [71] while improving the parsimony of the PCE model. Example 3 in Section 3.5 illustrates how this procedure can be used to determine the highest polynomial order.

While the conditional order-based sensitivity indices are useful for effective PCE modeling and, in turn, PCE-based sensitivity analyses, the new sensitivity indices do not directly serve the traditional purposes of sensitivity indices. In contrast, for example, the first-order full sensitivity indices and the total uncorrelated sensitivity indices directly help determine influential and non-influential inputs, respectively, in terms of their contributions to the output variance (also known as factor prioritization and factor fixing, respectively, in [56]; see also [40]).

3.5 Numerical examples

To validate the proposed data-driven sensitivity indices, this section presents four examples where inputs are dependent. We present our experiment results based on 500 replications with 95% confidence intervals wherever applicable. The confidence intervals are computed using 10,000 bootstrap samples of the 500 replications to improve upon the accuracy of the empirical confidence interval computed from the 500 replications [17].

3.5.1 Example 1

We use a benchmark example in [40] as our first validation case. In this case, inputs follow a three-dimensional multivariate normal distribution as follows:

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{12} & 1 & \rho_{23} \\ \rho_{13} & \rho_{23} & 1 \end{pmatrix} \right].$$

The output Y is simply modeled using a linear model $Y = X_1 + X_2 + X_3$.

Table 3.1 shows the first-order full sensitivity index $\bar{S}_{X_i}$ and total uncorrelated sensitivity index $ST^u_{X_i}$ for each input based on the analytical method [40]. These true indices are compared with the estimated indices from the proposed method. This example validates that the proposed method can estimate the first-order full sensitivity index and total uncorrelated sensitivity index based only on data without the knowledge of the distribution of dependent inputs and the model structure. Note that in this example, the *total* full sensitivity index is the same as the *first-order* full sensitivity index (i.e., $\overline{ST}_{X_i} = \bar{S}_{X_i}$) and that the *first-order* uncorrelated sensitivity index equals the *total* uncorrelated sensitivity index (i.e., $S^u_{X_i} = ST^u_{X_i}$) because there is no interaction effect. Thus, $\overline{ST}_{X_i}$ and $S^u_{X_i}$ are not presented.

3.5.2 Example 2

In this example from [65], the output Y is a non-linear function of four dependent inputs: $Y = X_1X_2 + X_3X_4$. Here, $(X_1, X_2) \in \left[0,1\right]^2$ is uniformly distributed within the triangle

Table 3.1: Sample mean of sensitivity indices and 95% confidence intervals.

$(\rho_{12}, \rho_{13}, \rho_{23})$	Input	$\bar{S}_{X_i}$		$ST^u_{X_i}$	
		Analytical method†	Proposed method‡	Analytical method†	Proposed method‡
(0.5,0.8,0)	X_1	0.945	0.945 (0.945, 0.945)	0.020	0.020 (0.020, 0.020)
	X_2	0.402	0.401 (0.400, 0.402)	0.055	0.055 (0.054, 0.055)
	X_3	0.579	0.579 (0.578, 0.579)	0.026	0.026 (0.026, 0.026)
(-0.5,0.2,-0.7)	X_1	0.490	0.491 (0.490, 0.492)	0.706	0.707 (0.706, 0.707)
	X_2	0.040	0.041 (0.040, 0.041)	0.375	0.374 (0.373, 0.374)
	X_3	0.250	0.250 (0.249, 0.251)	0.480	0.480 (0.479, 0.481)
(-0.49,-0.49,-0.49)	X_1	0.007	0.007 (0.007, 0.007)	0.974	0.974 (0.974, 0.974)
	X_2	0.007	0.007 (0.006, 0.007)	0.974	0.974 (0.974, 0.974)
	X_3	0.007	0.007 (0.007, 0.007)	0.974	0.974 (0.973, 0.974)

Note: †The value presented in [40] differs by up to 0.01 due to rounding. ‡The proposed method does not require any distributional assumption. The sample means of the sensitivity indices and 95% bootstrap confidence intervals (using 10,000 bootstrap samples) are calculated based on 500 simulation replications where each replication uses 500 random observations.

$X_1 + X_2 \leq 1$ and $(X_3, X_4) \in \left[0,1\right]^2$ is uniformly distributed within the triangle $X_1 + X_2 \geq 1$. Due to the symmetry of the model, the sensitivity indices of Y with respect to X_1 and X_3 are equal to those with respect to X_2 and X_4, respectively.

As shown in Table 3.2, the proposed method yields the estimates that are close to the analytical values of $\bar{S}_{X_i}$ and $ST^u_{X_i}$. In contrast to the benchmark method in [65] that requires the knowledge of joint probability distribution of the inputs, the proposed method is purely data-driven. $ST_{\{X_1,X_2\}}$ (or, $ST_{\{X_3,X_4\}}$) is estimated using $\sum_{i=1}^{2} \overline{ST}_{X_i}$ by permuting the inputs as (X_1, X_2, X_3, X_4) (or, (X_3, X_4, X_1, X_2)).

Figure 3.1 shows intricate working of the model by revealing how each input influences the output variance. The *total* full (uncorrelated) sensitivity index $\overline{ST}_{X_i}$ ($ST^u_{X_i}$) can be decomposed into the *first-order* full (uncorrelated) sensitivity index $\bar{S}_{X_i}$ ($S^u_{X_i}$) and the *rest of the total* effect, $\overline{ST}_{X_i} - \bar{S}_{X_i}$ ($ST^u_{X_i} - S^u_{X_i}$), which accounts for all the interactions of X_i.

Table 3.2: Sample mean of sensitivity indices and 95% confidence intervals.

Method	$\bar{S}_{X_1}$	$ST^u_{X_2}$	$ST_{\{X_1,X_2\}}$	$\bar{S}_{X_3}$	$ST^u_{X_4}$	$ST_{\{X_3,X_4\}}$
Analytical method†	0.033	0.067	0.100	0.233	0.666	0.900
Benchmark method‡	0.032	0.071	0.103	0.226	0.669	0.895
	(0.028, 0.037)	(0.066,0.077)	(0.095,0.114)	(0.209,0.248)	(0.639,0.705)	(0.848, 953)
Proposed method*	0.035	0.066	0.101	0.233	0.663	0.896
	(0.034, 0.036)	(0.066,0.067)	(0.100,0.103)	(0.231,0.235)	(0.661,0.666)	(0.892, 901)

Note: †The value is provided in [65]. ‡The value is estimated using the method proposed in [65] and the confidence interval is calculated based on 16,380 random observations under the assumption that the joint probability distribution of the inputs is *known*. *The proposed method does not require any distributional assumption. The sample means of sensitivity indices and 95% bootstrap confidence intervals (using $10,000$ bootstrap samples) are calculated based on 500 replications. In each replication, sensitivity indices are calculated using 500 random observations.

The gap between the two lines on the left (right) graph in Figure 3.1 shows the magnitude of $\overline{ST}_{X_i} - \bar{S}_{X_i}$ $(ST^u_{X_i} - S^u_{X_i})$, indicating how much of the total effect of X_i is attributed to the interaction effects compared to the first-order effect when we consider the full (uncorrelated) contribution of X_i.

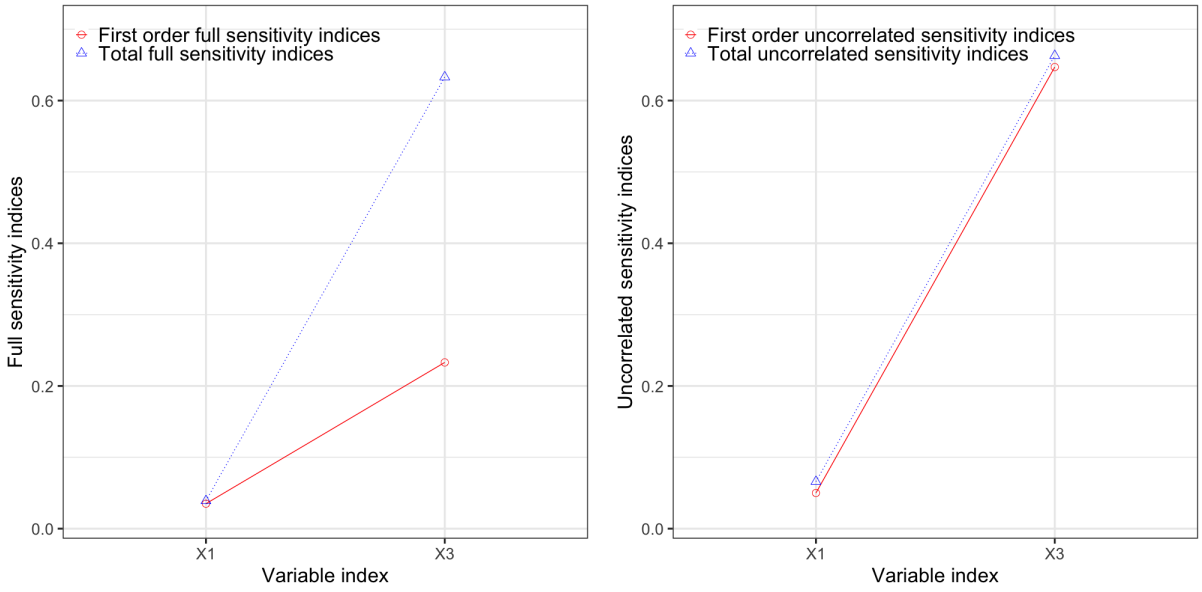


Figure 3.1: The left-hand side graph shows the *full* sensitivity indices for dependent input variables and the right-hand side graph shows the *uncorrelated* sensitivity indices for dependent input variables in Example 2.

3.5.3 Example 3

For the third example, we modified the example in [30] to have a more complex structure and involve multiple types of probability distributions as follows:

$$\begin{aligned}
\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} &\sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.3 \\ 0 & 0 & 0.3 & 1 \end{pmatrix}\right], \\
X &\sim \mathcal{U}(0,1), \\
X_5 &= \theta_1 X + \mathcal{U}(0,1), \\
X_6 &= \theta_2 X + \theta_3 X^2 + \mathcal{U}(0,1), \\
Y &= X_1 X_2 + X_3 X_4 + X_5 X_6.
\end{aligned} \tag{3.9}$$

Here, (X_1, X_2, X_3, X_4) follows a multivariate Gaussian distribution with the parameters as above. The inputs X_5 and X_6 are dependent on each other, but their dependency cannot be explained by their first-order conditional moments. In this experiment, we set $(\theta_1, \theta_2, \theta_3) = (0.4, 0.6, 1)$ and obtain 10,000 random observations. Because the cumulative sum of the first two conditional order-based sensitivity indices is $\sum_{i=1}^{2} \tilde{S}_i = 1$, we exclude third- and higher-order interaction terms in the PCE model to make it sparse. As for the two-way interaction terms, as shown in Figure 3.2, most of the corresponding PCE coefficients are nearly zero except for the polynomial terms, X_1X_2, X_3X_4, and X_5X_6. It indicates that X_1X_2, X_3X_4, and X_5X_6 are the only interaction terms in the true model. Various sparse PCE approaches [7, 9, 31, 52] can be additionally applied here to construct a sparser PCE model with only the important orthonormal polynomials of inputs.

Because $\{X_1, X_2\}$, $\{X_3, X_4\}$, and $\{X_5, X_6\}$ are mutually independent, we can infer from the conditional order-based sensitivity indices that the output Y is composed of three non-interacting functions $f_{12}(X_1, X_2)$, $f_{34}(X_3, X_4)$, and $f_{56}(X_5, X_6)$. Thanks to this special structure, we can directly calculate the total sensitivity indices for $\{X_1, X_2\}$, $\{X_3, X_4\}$, and $\{X_5, X_6\}$. Without permuting the order of the input variables, the total Sobol indices can

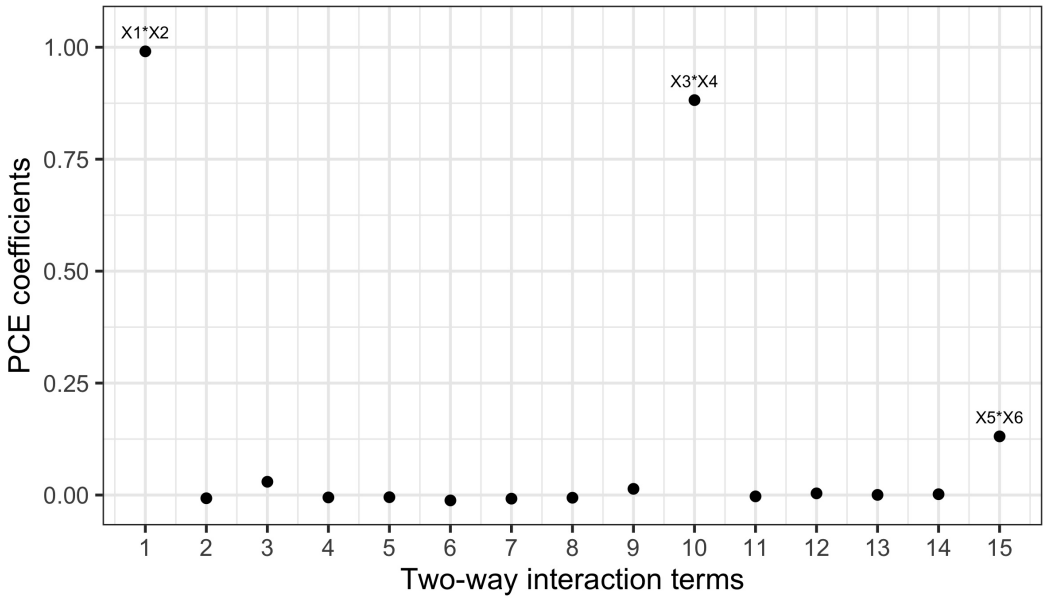


Figure 3.2: PCE coefficients v.s. the two-way interaction terms in the PCE model in Example 3. The significant interaction terms, X_1X_2, X_3X_4, and X_5X_6, are identified.

be calculated as follows:

$$\begin{aligned} ST_{\{X_1,X_2\}} &= \overline{ST}_{X_1} + \overline{ST}_{X_2}, \\ ST_{\{X_3,X_4\}} &= \overline{ST}_{X_3} + \overline{ST}_{X_4}, \\ ST_{\{X_5,X_6\}} &= \overline{ST}_{X_5} + \overline{ST}_{X_6}, \end{aligned} \tag{3.10}$$

where $\overline{ST}_{X_i}$ is the conditional *total full* sensitivity index defined in Section 3.4.2.

We validate the sensitivity indices from the proposed method with the values from an analytical method (see Appendix B.2). We also calculate $ST_{\{X_1,X_2\}}$ and $ST_{\{X_3,X_4\}}$ using the benchmark method in [40] assuming the knowledge that (X_1, X_2, X_3, X_4) is multivariate Gaussian distributed and $\{X_1, X_2\}$, $\{X_3, X_4\}$, and $\{X_5, X_6\}$ are mutually independent. Then, $ST_{\{X_5,X_6\}}$ is calculated based on the fact that $ST_{\{X_1,X_2\}} + ST_{\{X_3,X_4\}} + ST_{\{X_5,X_6\}} = 1$.

As shown in Table 3.3, the sensitivity indices from our method are close to . those from the benchmark method and the analytical values. Note that, in contrast to the benchmark method, the proposed method is a data-driven approach that does not impose any assumption

Table 3.3: Sample means of sensitivity indices and 95% confidence intervals.

Method \ Input set	$ST_{\{X_1,X_2\}}$	$ST_{\{X_3,X_4\}}$	$ST_{\{X_5,X_6\}}$
Analytical method	0.402	0.438	0.160
Proposed method*	0.402	0.438	0.160
	(0.401, 0.404)	(0.437, 0.439)	(0.159, 0.160)
Benchmark method[40]	$0.403^{\dagger}$	$0.439^{\dagger}$	$(0.158)^{\ddagger}$
	(0.402, 0.404)	(0.438, 0.440)	(0.157, 0.160)

Note: †The value is obtained using the sample variance to estimate $Var(Y)$ in the denominator of the sensitivity index instead of using the PCE coefficients from the benchmark method (see Eq. (3.4)) because the latter estimation suffers a non-negligible bias in this example that does not satisfy the assumption of the benchmark method. ‡The value cannot be obtained directly from the benchmark method, but we calculate the value based on the assumption that the user knows that X_5 and X_6 are independent of the rest of the inputs and that (X_1, X_2, X_3, X_4) follows a multivariate Gaussian distribution. *The proposed method does not require any assumption. The sample means of sensitivity indices and 95% bootstrap confidence intervals (using 10,000 bootstrap samples) are calculated based on 500 replications. In each replication, sensitivity indices are calculated using 5,000 random observations of the inputs and output in Eq. (3.9).

on the inputs.

3.5.4 Example 4

As the fourth example, we consider the 23-bar horizontal truss example in [67]. The output of interest, Y, is a downward vertical displacement at the mid span of the structure subject to random loads. As depicted in Figure 3.3, the uncertainty of Y depends on the ten random inputs in $\boldsymbol{X} = (E_1, E_2, A_1, A_2, P_1, \ldots, P_6)$: namely, uncertain Young modulus $E_i, i = 1, 2$, and

uncertain cross-sectional area $A_i, i = 1, 2$, for two different groups of bars (horizontal for $i = 1$ and diagonal for $i = 2$), and the random loads $P_i, i = 1, 2, \cdots, 6$. The inputs $E_i, i = 1, 2$, and $A_i, i = 1, 2$, are assumed to be mutually independent and follow the following distributions:

$$\begin{aligned} E_1, E_2 &\sim \mathcal{LN}(2.1 \times 10^{11}, 2.1 \times 10^{10}) \text{ [Pa]}, \\ A_1 &\sim \mathcal{LN}(2.0 \times 10^{-3}, 2.0 \times 10^{-4}) \text{ [m}^2\text{]}, \\ A_2 &\sim \mathcal{LN}(1.0 \times 10^{-3}, 1.0 \times 10^{-4}) \text{ [m}^2\text{]}, \end{aligned}$$

where $\mathcal{LN}(\mu, \sigma)$ denotes the lognormal distribution with mean μ and standard deviation σ.

The dependent inputs $P_i, i = 1, 2, \cdots, 6$, have the following Gumbel marginal distribution function with mean $\mu = 5 \times 10^4$ [N] and standard deviation $\sigma = 7.5 \times 10^3$ [N]:

$$F_i(x; \alpha, \beta) = e^{-e^{-(x-\alpha)/\beta}}, i = 1, 2, \ldots, 6,$$

where $\beta = \sqrt{6}\sigma/\pi$, $\alpha = \mu - \gamma\beta$, and $\gamma \approx 0.5772$ is the Euler-Mascheroni constant. The dependence between the inputs $P_i, i = 1, \ldots, 6$, is encoded in the C-vine copula with the following density:

$$c_{\boldsymbol{X}}^{(\mathcal{G})}(u_1, \ldots, u_6) = \prod_{j=2}^{6} c_{1j;\theta=1.1}^{(\mathcal{GH})}(u_1, u_j), \tag{3.11}$$

where $c_{1j;\theta=1.1}^{(\mathcal{GH})}$ is the density of the pair-copula between P_1 and P_j, $j = 2, \ldots, 6$. $\mathcal{GH}$ represents the Gumbel-Hougaard family whose bivariate copula is

$$C_\theta^{(\mathcal{GH})}(u, v) = \exp\left(-\left((-\log u)^\theta + (-\log v)^\theta\right)^{1/\theta}\right), \quad \theta \in [1, \infty).$$

The parameter $\boldsymbol{\theta}$ decides the dependence between two loads (i.e., the larger parameter $\boldsymbol{\theta}$ the stronger dependence). In this case, P_1 is equally and positively correlated with each of $P_2, \ldots, P_6$. Thus, $P_2, \ldots, P_6$ are positively correlated with each other although they are conditionally independent given P_1 (see a sample correlation matrix for the loads in Appendix A.3). The output Y is simulated using the response surface model (see Appendix A.4) in [37], which was constructed based on a finite element analysis. The explicit relationship between Y and $\boldsymbol{X}$ in the response surface model allows us to evaluate the estimated sensitivity indices.

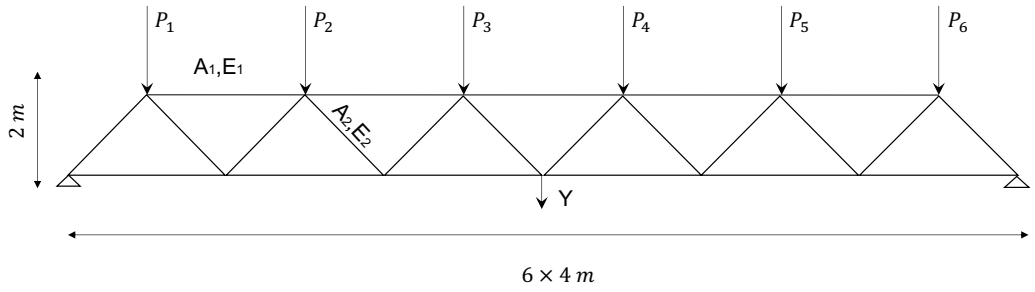


Figure 3.3: Scheme of the horizontal truss model modified from [37]. The downward vertical displacement at the mid span of the structure Y depends on Young modulus $E_i, i = 1, 2$, cross-sectional area $A_i, i = 1, 2$ for both horizontal and diagonal bars, and the random loads $P_i, i = 1, 2, \cdots, 6$.

This realistic problem with dependent inputs has neither analytically known sensitivity indices nor any benchmark methods that attempted to estimate the sensitivity indices (see [9] for a related sensitivity analysis with *independent* inputs). Implementing a brute-force Monte Carlo approach is computationally challenging, if not infeasible, because the analytical expressions of the sensitivity indices involve variances of conditional expectations that condition on (multiple combinations of) multiple inputs. A similar challenge lies in even estimating Sobol indices for *independent* inputs and is studied extensively in the literature [46, 51, 66]. No Monte Carlo method has satisfactorily addressed the computational challenge yet. Extending the existing Monte Carlo methods for independent inputs to handle dependent inputs is left for future work.

In this study, we examine the estimated sensitivity indices to confirm that they agree with their expected physical interpretations. As it is shown in Table 3.4, P_2 (resp. P_3) and P_5 (resp. P_4) have almost the same sensitivity indices for $\bar{S}$, $\overline{ST}$, S^u, and ST^u. This can be explained by a) the physical symmetry between P_2 (resp. P_3) and P_5 (resp. P_4) with respect to the location at which the output Y is measured (see Figure 3.3 and the response

surface model in Appendix A.4) and b) their symmetric correlations with other inputs (recall the copula density in Eq. (3.11) and see Appendix A.3). On the other hand, the differences between the *full* sensitivity indices (i.e., $\bar{S}$ and $\overline{ST}$) for P_1 and P_6 can be explained by the fact that P_1 has over 5 times stronger correlations than P_6 with $P_2, \ldots, P_5$ (see Appendix A.3). In contrast, the *uncorrelated* sensitivity indices (i.e., S^u and ST^u) for P_1 and P_6 are the same (up to 4 decimal places) because the uncorrelated effects of P_1 and P_6 on Y should be very similar (see the coefficients of the response surface model in Appendix A.4.). The sensitivity indices for all the other inputs are similarly confirmed to be consistent with their expected physical interpretations based on the response surface model, which reflects the physical relationship between Y and $\boldsymbol{X}$, and the dependence structure of $\boldsymbol{X}$.

In addition, although not directly comparable due to different settings, still the first and total uncorrelated sensitivity indices (i.e., S^u and ST^u) have similar magnitudes as the first and total Sobol indices (i.e., S and ST) reported in Table 5 of [36] and Table 2 of [9], respectively, for all ten inputs. Both articles [9, 36] assumed that P_1 through P_6 are *independent*, and directly computed Y using a finite element model.

3.6 Conclusion

In this chapter, data-driven sensitivity indices for a model with *dependent* inputs are proposed using the PCE without imposing any strong assumptions on the model inputs. The modified Gram-Schmidt algorithm with the empirical measure is utilized to construct orthonormal polynomials for a PCE model on the merit of numerical stability. The proposed data-driven method yields the *full* sensitivity indices and the *uncorrelated* sensitivity indices by constructing ordered partitions of orthonormal polynomials of inputs for a PCE model. The proposed conditional order-based sensitivity indices for a model with *dependent* inputs help reduce the complexity of a PCE model while keeping the effect hierarchy principle. Four numerical examples validate the proposed method.

The proposed method requires polynomials of inputs, which are fed into the modified Gram-Schmidt algorithm, to be linearly independent. This suggests a future research direc-

Table 3.4: Sample means of sensitivity indices and 95% confidence intervals.

Input \ Sensitivity indices	$\bar{S}$	$\overline{ST}$	S^u	ST^u
E_1	0.324 (0.321, 0.327)	0.371 (0.367, 0.374)	0.286 (0.284, 0.288)	0.312 (0.310, 0.314)
E_2	0.013 (0.012, 0.014)	0.036 (0.035, 0.037)	0.009 (0.008, 0.009)	0.009 (0.008, 0.009)
A_1	0.325 (0.322, 0.328)	0.370 (0.367, 0.374)	0.285 (0.283, 0.287)	0.310 (0.308, 0.312)
A_2	0.013 (0.012, 0.014)	0.037 (0.036, 0.038)	0.008 (0.008, 0.008)	0.008 (0.008, 0.008)
P_1	0.065 (0.063, 0.068)	0.096 (0.093, 0.099)	0.004 (0.004, 0.004)	0.004 (0.004, 0.004)
P_2	0.060 (0.058, 0.062)	0.088 (0.085, 0.090)	0.033 (0.033, 0.033)	0.035 (0.035, 0.036)
P_3	0.105 (0.103, 0.108)	0.135 (0.132, 0.138)	0.068 (0.067, 0.068)	0.073 (0.072, 0.073)
P_4	0.102 (0.010, 0.105)	0.130 (0.128, 0.133)	0.068 (0.067, 0.068)	0.073 (0.072, 0.073)
P_5	0.057 (0.055, 0.059)	0.084 (0.082, 0.087)	0.033 (0.033, 0.033)	0.035 (0.035, 0.036)
P_6	0.017 (0.016, 0.018)	0.043 (0.042, 0.045)	0.004 (0.004, 0.004)	0.004 (0.004, 0.004)

Note: The sample means of sensitivity indices and 95% bootstrap confidence intervals (using $10,000$ bootstrap samples) are calculated based on 500 replications. In each replication, sensitivity indices are calculated using 500 random observations of the inputs and outputs in Eq. (B.1).

tion because there are multiple ways of constructing linearly independent polynomials from a linearly dependent polynomial basis. How to build a theoretically and practically desirable basis warrants more investigation.

Chapter 4

DATA-DRIVEN SPARSE POLYNOMIAL CHAOS EXPANSION FOR MODELS WITH DEPENDENT INPUTS

4.1 Abstract

Polynomial chaos expansions (PCEs) have been used in many real-world engineering applications to quantify how the uncertainty of the output is propagated from the inputs. PCEs for models with independent inputs have been extensively explored in the literature. Recently, different approaches have been proposed for models with dependent inputs to expand the use of PCEs to more real-world applications. Typical approach is building PCEs based on the Gram-Schmidt algorithm or to transforming the dependent inputs into independent inputs. However, the two approaches have their limitations regarding computational efficiency and additional assumptions about the input distributions, respectively. In this chapter, we propose a data-driven approach to build sparse PCEs for models with dependent inputs. The proposed algorithm recursively constructs orthonormal polynomials using the initial polynomials based on the correlations with the output. The proposed algorithm on building sparse PCEs not only reduces the number of minimally required observations but also improves the numerical stability and computational efficiency. Four numerical examples are implemented to validate the proposed algorithm.

4.2 Introduction

Uncertainty quantification plays a critical role in many domains of real-world engineering applications as it characterizes and reduces the uncertainties of the system outputs in those applications. Surrogate models are often served as mathematical models to describe how the uncertainty of a system output is propagated from the inputs. Among the surrogate

models, polynomial chaos expansions (PCEs) have been widely used in the literature to conduct the uncertainty quantification on the outputs in many industrial applications including thermodynamics [3], electromagnetism [13], power systems [49], building systems [38], and manufacturing [27].

To accurately conduct uncertainty quantification for models with different types of inputs, a variety of PCEs have been developed in the literature. For models with independent inputs, the Wiener chaos expansion, which is known as the first PCE in the literature, uses to Hermite polynomials to construct PCE models for Gaussian-distributed inputs [69]. Later PCEs, including the generalized PCE (gPCE) [72], the multi-element gPCE (ME-gPCE) [68], the moment-based arbitrary PCE (aPCE) [50], and the Gram-Schmidt based PCE (GS-PCE) [70], are developed for independent inputs following non-Gaussian distributions.

Even though the GS-PCE can also be served to construct PCEs for models with dependent inputs, the procedure of using Gram-Schmidt algorithm is computationally demanding as the number of the inputs increases or the expansion order increases [67]. Therefore, [67] provides an alternative method to construct PCE for models with dependent inputs by transforming the dependent inputs into independent inputs using the Rosenblatt transformation. However, this approach might not be applicable to many engineering applications due to the fact that the Rosenblatt transformation requires the knowledge about the conditional probability density functions about the inputs, which is usually not the case in practice. Thus, how to efficiently construct a PCE for models with dependent inputs without using distribution information about the inputs still requires more investigations.

Consequently, the main contribution of this paper is to propose a sparse PCE algorithm which can efficiently construct sparse PCEs for models with both independent or dependent inputs regardless the input distribution. To the best of our knowledge, the proposed method is also the first data-driven method that constructs a sparse PCE for models with dependent inputs without requiring a large number of observations. We validate our proposed method empirically using four simulation examples in the sense of estimating the variance of the output. The simulation examples show the computational efficiency and numerical stability

of using the proposed method comparing with the state-of-the-art method on constructing a sparse PCE model regardless the input distribution or their dependency structure.

The remainder of this paper is organized as follows. Sec. 4.3 briefly reviews the technical background on the PCE and the state-of-the-art methods on constructing sparse PCEs. Sec. 4.4 proposes the algorithm of our proposed method and discuss the advantages of using the proposed method. In Sec. 4.5, the proposed algorithm is empirically evaluated using four simulation examples. Sec. 4.6 concludes the paper with a discussion on future research directions.

4.3 Background

In this section, we will first introduce PCE models and how to estimate the lower-order moments using their coefficients. Then we will particularly review how to construct GS-PCE since it is regarded as the pioneering work of data-driven PCE model regardless the distribution and dependency about the inputs. In the end, we will review the state-of-the-art routine for constructing sparse PCEs for dependent inputs which applies the least angle regression method on orthonormal polynomials constructed using the modified Gram-Schmidt algorithm.

4.3.1 PCE model

PCE models the relationship between the n random inputs in $\boldsymbol{X}$ and the output Y uses a finite number of orthonormal polynomials as follows:

$$Y = f(\boldsymbol{X}) \approx \sum_{i=0}^{P} \theta_i \psi_i(\boldsymbol{X}), \tag{4.1}$$

where θ_i, $i = 0,1,2,\ldots,P$, are called PCE coefficients and ψ_i, $i = 1,2,\ldots,P$ are orthonormal polynomials. The orthonormal polynomials can be constructed based on different PCE models. We will particularly introduce how to construct orthonormal polynomials using the

modified Gram-Schmidt algorithm in Section 4.3.2.

$$P+1=\binom{n+p}{n} \tag{4.2}$$

is the number of polynomial terms, where p is the highest polynomial degree in the PCE model. As p increases to infinity, the approximated error of estimating the output using the PCE model converges to 0.

In this chapter, the PCE coefficients are solved by solving an overdetermined linear system equations by adopting a regression-based in the least-squares sense as follows [53]:

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^{P+1}}{\operatorname{argmin}}\sum_{j=1}^{m}\left(Y_j-\sum_{i=0}^{P}\theta_i\psi_i\left(\boldsymbol{X}_j\right)\right)^2, \tag{4.3}$$

where $\boldsymbol{\theta}$ denotes $(\theta_0,\theta_1,\ldots,\theta_P)$. Y_j and $\boldsymbol{X}_j$ represent the output and the input vector of the j^{th} observation, $j=1,\ldots,m$, respectively.

Thanks to the orthogonality of the orthonormal polynomials, we can approximate the lower-order moments of output Y using the PCE coefficients as follows:

$$\begin{aligned}&\mathbb{E}(y)\approx\theta_0,\\&Var(y)\approx\sum_{i=1}^{P}\theta_i^2.\end{aligned} \tag{4.4}$$

The accuracy of estimating the lower-order moments improves as P in Eq. (4.4) increases.

4.3.2 GS-PCE

The GS-PCE constructs orthonormal polynomials based on P initial polynomials $(e_i)_{i\in\{1,2,\ldots,P\}}$, where $(e_i)_{i\in\{1,2,\ldots,P\}}$ are assumed to be linearly independent. Then the orthonromal polynomials $(\psi_i(\boldsymbol{X}))_{i\in\{1,2,\ldots,P\}}$ are obtained using the modified Gram-Schmidt algorithm described as follows [39]:

Algorithm 4 Modified Gram-Schmidt Algorithm

Input: P linearly independent initial polynomials $(e_i)_{i\in\{1,2,\ldots,P\}}$.

Output: P orthonormal polynomials $(\psi_i(\boldsymbol{X}))_{i\in\{1,2,\ldots,P\}}$.

1: **for** $i = 1, 2, \ldots, P$ **do**
2: $\quad\phi_i(\boldsymbol{X}) \leftarrow e_i(\boldsymbol{X})$
3: $\quad$**for** $k = 1, 2, \ldots, i-1$ **do**
4: $\quad\quad\phi_i(\boldsymbol{X}) \leftarrow \phi_i(\boldsymbol{X}) - \langle\phi_i(\boldsymbol{X}), \psi_k(\boldsymbol{X})\rangle\psi_k(\boldsymbol{X})$
5: $\quad$**end for**
6: $\quad\psi_i(\boldsymbol{X}) \leftarrow \frac{\phi_i(\boldsymbol{X})}{||\phi_i(\boldsymbol{X})||_2}$
7: **end for**

The inner-product in the algorithm is defined with respect to the empirical measure in this chapter.

Even though GS-PCE provides feasibility on constructing orthonormal polynomials for dependent inputs following arbitrary distributions, it is computational demanding as the number of input or polynomial order increases [67]. In addition, the GS-PCE might be inaccurate for models with highly correlated inputs since the constructed orthonormal polynomials might lose their orthogonality due to the rounding error [21].

4.3.3 Sparse PCE

A variety sparse PCEs have been explored in the literature. [7] proposes a greedy forward-backward selection algorithm, is regarded as a pioneering work for constructing sparse PCEs in the literature. Based on this work, many other techniques have been introduced to construct sparse PCEs using the least angle regression (LAR) and the diffeomorphic modulation under observable response preserving homotopy (D-MORPH) regression [14] as well as solving a sparse regression with a regularization term [22, 31]. Those techniques can also be applied to construct sparse PCEs with m orthonormal polynomials using the algorithm summarized in Algorithm 5.

Algorithm 5 Sparse PCE algorithm for models with dependent inputs

Input: At least $P+1$ random observations of output Y and inputs $\boldsymbol{X}$.

Output: A sparse PCE representation of Y with respect to $\{\boldsymbol{\psi}_{i'}(\boldsymbol{X})\}_{i'=1}^{m}$.

1: Construct P initial linearly independent polynomials $(e_i)_{i\in\{1,2,\ldots,P\}}$.

2: Construct orthonormal polynomial basis $\{\psi_i(\boldsymbol{X})\}_{i=1}^{P}$ using the modified Gram-Schmidt polynomials described in Algorithm 4.

3: Construct a sparse PCE model of Y by selecting $\{\boldsymbol{\psi}_{i'}(\boldsymbol{X})\}_{i'=1}^{m} \subseteq \{\psi_i(\boldsymbol{X})\}_{i=1}^{P}$ based on sparse algorithm.

As the procedure of the modified Gram-Schmidt algorithm is embedded in Algorithm 5, it inherits the computational inefficiency and inaccuracy from Algorithm 4. To this regard, we propose a algorithm in the following section that improve the efficiency and accuracy of constructing a sparse PCE for a model with dependent inputs. Step 3 in Algorithm 5 can be placed with varied methods that construct sparse PCEs. In this chapter, we use the LAR-based method to build sparse PCEs based on the orthonormal polynomials constructed using the modified Gram-Schmidt algorithm as the benchmark method.

4.4 Methodology

As we described in Section 4.3.2, constructing orthonormal polynomials becomes more computationally demanding as the number of inputs or polynomial order increases. Therefore, we propose a new algorithm which builds a sparse PCE for inputs regardless of the distributions and dependency structure of the inputs. The proposed algorithm not only relaxes the need for large number of random observations but also improves the numerical accuracy and computational efficiency.

Unlike the benchmark method described in Algorithm 5, which constructs P orthonoraml polynomials based on P initial linearly independent polynomials before applying an operator to construct a sparse PCE, the proposed algorithm only constructs a limited number of orthonormal polynomials that truly explain the output Y from P initial polynomials.

The proposed sparse PCE algorithm is a recursive algorithm which requires a pre-defined threshold value $\epsilon \in (0,1)$ and a set of initial polynomials $(e_i^{(1=0)})_{i\in\{1,2,\ldots,P^{(1=0)}\}}$ at its initial step, where l represents the iteration number. As it is defined in Eq. (4.2), P depends on the number of inputs n and the polynomial order p. In this chapter, the P initial polynomials are constructed by the tensor product of the univariate polynomial of each input $X_i \in \boldsymbol{X}, i = 1,2,\ldots,n$ as follows:

$$(e_i^{(1=0)})_{i\in\{1,2,\ldots,P^{(1=0)}\}} = \left\{ \prod_{k=1}^{n} X_k^{j_k} : j_k \in \{0,1,\ldots p\}, \sum_{k=1}^{n} j_k \leq p \right\}. \tag{4.5}$$

In the l^{th} iteration, we calculate $\forall i \in \{1,2,\ldots,P^{(l-1)}\}, \rho(e_i^{(l-1)}(\boldsymbol{X}), Y)$, where ρ is an operator that calculates the empirical Pearson correlation coefficient between two variables. The algorithm stops with the following condition:

$$\forall i \in \{1,2,\ldots,P^{(l-1)}\}, \rho(e_i^{(l-1)}(\boldsymbol{X}), Y) < \epsilon. \tag{4.6}$$

Otherwise, we update $(e_i^{(l)})_{i\in\{1,2,\ldots,P^{(l)}\}}$ by selecting $e_i^{(l-1)}(\boldsymbol{X}) \in (e_i^{l-1})_{i\in\{1,2,\ldots,P^{(l-1)}\}}$ as follows:

$$\forall i \in \{1,2,\ldots,P^{(l-1)}\}, \rho((e_i^{(l-1)})(\boldsymbol{X}), Y) \geq \epsilon. \tag{4.7}$$

This procedure prevents selecting polynomials that are linearly dependent or highly correlated with the constructed orthonormal polynomials in the previous iterations and reduces the number of constructed orthonormal polynomials compared with the modified Gram-Schmidt algorithm. In the next step, we select one polynomial $(e_i^{(l)}(\boldsymbol{X}) \in (e_i^l)_{i\in\{1,2,\ldots,P^{(l)}\}})$, which satisfies the equation as follows:

$$\forall j \in \{1,2,\ldots,P^{(l)}\}, \rho(e_i^{(l)}(\boldsymbol{X}), Y) \geq \rho(e_j^{(l)}(\boldsymbol{X}), Y). \tag{4.8}$$

After we get $e_i^{(l)}(\boldsymbol{X})$, we transform it into an orthonormal polynomial regarding $\{\boldsymbol{\psi}_i(\boldsymbol{X})\}_{i=1}^{l-1}$ using Steps 3–6 in Algorithm 4. Unlike the modified Gram-Schmidt algorithm, which constructs orthonormal polynomials based on the initial polynomials whose ordering is defined, the proposed algorithm constructs the orthonormal polynomials based on their correlations

with Y. To be more specific, the polynomials which have high correlations with Y are selected first to construct the orthonormal polynomials. It is shown that such ordering improves the numerical stability of constructing orthonormal polynomials in Section 4.5. The proposed algorithm is summarized in Algorithm 6.

Algorithm 6 Step-forward sparse PCE

Input: Random observations of output Y and inputs $\boldsymbol{X}$; Threshold value ϵ; Iteration counter $l = 0$.

Output: A sparse PCE representation of Y with respect to $\{\boldsymbol{\psi}_i(\boldsymbol{X})\}_{l=0}^{m}$.

1: Construct P initial polynomials $(e_i^{(l)})_{i\in\{1,2,\ldots,P^{(l)}\}}$ using Eq. (4.5).

2: Check Eq. (4.6) and if it is satisfied, go to Step 8.

3: Select $e_l(\boldsymbol{X}) \in (e_i^{(1-1)})_{i\in\{1,2,\ldots,P^{(1-1)}\}}$ based on Eq. (4.8).

4: Update $(e_i^{(l)}(\boldsymbol{X}))_{i\in\{1,2,\ldots,P^{(l+1)}\}}$ based on Eq. (4.7).

5: Transform $e_l(\boldsymbol{X})$ into $\psi_l(\boldsymbol{X})$ using Steps 4–6 in Algorithm 4, where $e_l(\boldsymbol{X})$ and $\{\boldsymbol{\psi}_i(\boldsymbol{X})\}_{i=0}^{l-1}$ replace $\phi_i(\boldsymbol{X})$ and $\{\psi_k(\boldsymbol{X})\}_{k=1}^{i-1}$, respectively.

6: Increase l by 1 and go to Step 2.

7: Model the output Y using a sparse PCE model with respect to $\{\boldsymbol{\psi}_i(\boldsymbol{X})\}_{i=1}^{l}$ using Eq. (4.1).

The pre-specified ϵ decides the goodness of fitting and the sparseness of the PCE model. It is chosen by conducting an n-fold cross-validation. To conduct the n-fold cross-validation, we first randomly split all random observations into n subsets where each subset contains the same number of observations. Then we treat each subset as a testing set and the rest of the subsets as a training set. After that, we construct a sparse PCE using Algorithm 6 and estimate PCE coefficients of a sparse PCE based on the training set and predict Y on the testing set using its inputs. By following this procedure for n times, where each time we use

a different subset as the testing set, we choose ϵ based on the optimization as follows:

$$\underset{\epsilon\in(0,1)}{\operatorname{argmin}} \sum_{j=1}^{n} \left(\boldsymbol{Y}_j - \sum_{i=0}^{P^{(j)}} \theta_{i,\epsilon}^{(j)} \psi_{i,\epsilon}^{(j)} \left(\boldsymbol{X}_j\right) \right)^2, \tag{4.9}$$

where $\boldsymbol{Y}_j$ represents the outputs in the j^{th} fold. $\{\psi_i^{(j)}\left(\boldsymbol{X}_j\right)\}_{i=0}^{P^{(j)}}$ and $\boldsymbol{\theta}^{(j)}$ are the orthonormal polynomials and the estimated PCE coefficents estimated based on the rest of the folds.

4.5 Empirical validation

In this section, we present four numerical examples to empirically validate the proposed method. The first and second examples consider the inputs are independent and dependent, respectively, in synthetic settings. The third and forth examples consider modeling the output using sparse PCE for dependent inputs in real-world problems.

In this chapter, we use the relative error (RE) to compare the numerical stability on estimating the standard deviation of Y for both the benchmark method and the proposed method. The relative error is defined as follows:

$$\epsilon_{re} = \frac{|\sigma_Y - \hat{\sigma}_Y|}{\sigma_Y}, \tag{4.10}$$

where σ_Y is the theoretical standard deviation of Y or a estimation using the Monte Carlo method based on a large number of random observations. $\hat{\sigma}_Y$ is the estimated standard deviation of Y using either the benchmark method or the proposed method using Eq. (4.4). In each simulation example, a smaller ϵ_{re} represents a more accurate estimation. Note that the accurate estimation of σ_Y using all the PCE coefficients in Eq. (4.4) indicates that the PCE represents an accurate spectral decomposition of the uncertainty in Y with respect to $\boldsymbol{X}$. Thus, the PCE is useful for uncertainty quantification such as the variance-based sensitivity analysis, which aims to quantify the influence of each input on the output variance [39].

4.5.1 Ishigami function approximation

We use the Ishigami function [29] in Eq. (4.11) as our first simulation example to validate the proposed method for a model with *independent* inputs.

$$Y = \sin(X_1) + 7\sin^2(X_2) + 0.1X_3^4\sin(X_1), \tag{4.11}$$

where $X_i \sim \mathcal{U}(-\pi, \pi), i = 1, 2, 3$. This function is widely used as a test function to benchmark PCE methods due to its strong non-linearity and non-monotonicity [67].

In this example, we first show how to choose the threshold value for the proposed method. In addition, we compare the proposed method with the benchmark method in Section 4.3.3 in terms of the estimation accuracy and computational efficiency.

The 5-fold cross-validation is used to find the optimal threshold value for the proposed method considering different polynomial orders using 200 random observations. The blue (resp. red) points of the left subfigure and right subfigure in Figure 4.1 represent the threshold values that correspond to the minimal cross-validation errors as defined in Eq. (4.9) for the PCE with the polynomial order of 3 (resp. 4) and 8 (resp. 9), respectively. In addition, as it is presented in Figure 4.1, the proposed model with a higher polynomial order achieves a smaller cross-validation error than the model with a lower polynomial order. It reflects the fact that a more complex PCE model tends to better approximate a target function.

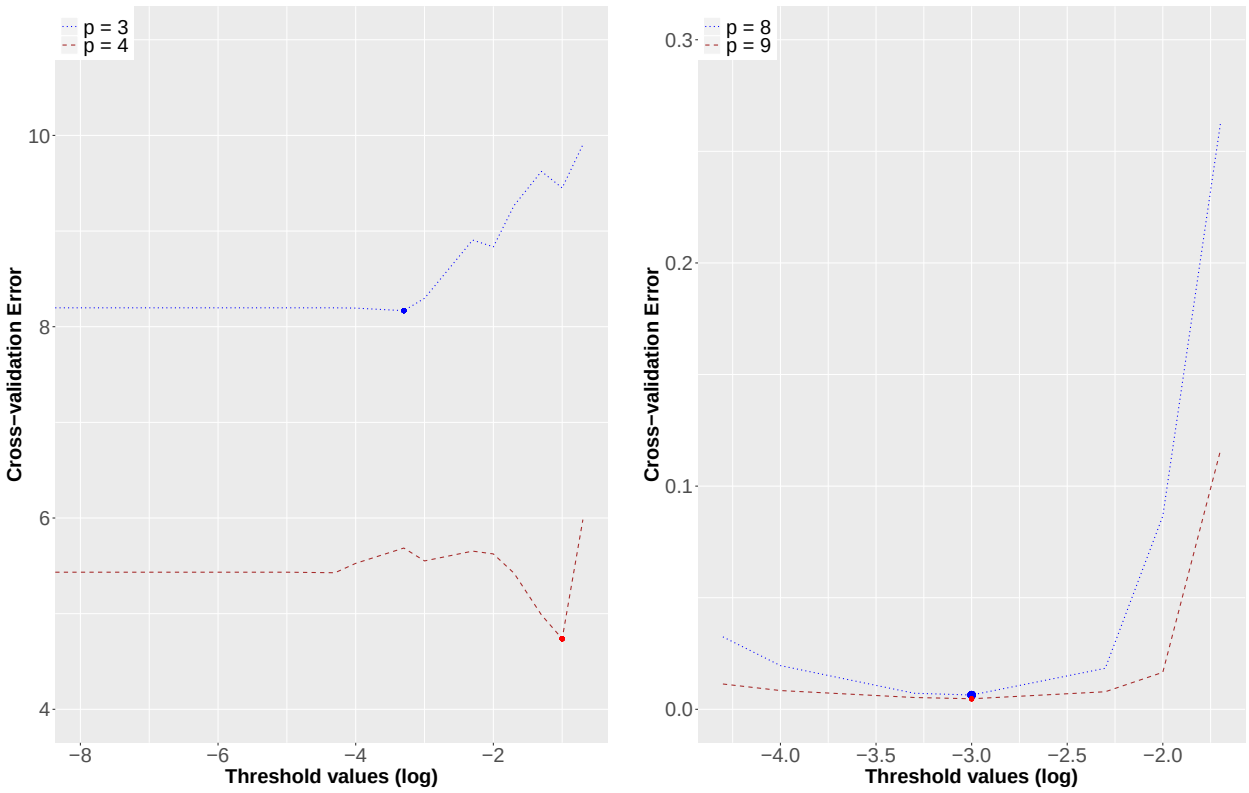


Figure 4.1: The left subgraph shows the 5-fold cross-validation errors of PCEs with $p = 3$ and $p = 4$ across different threshold values for the proposed method. The right subgraph shows the 5-fold cross-validation errors of PCEs with $p = 8$ or $p = 9$ across different threshold values. The blue (resp. red) points represent the threshold values that correspond to the threshold values in Eq. (4.9) for PCEs with $p = 3$ (resp. 4) or $p = 8$ (resp. 9).

We first compare how the polynomial order p and the sample size m interactively affect the performance for both the benchmark method and the proposed method. The left subfigure in Figure 4.2 shows that the proposed method achieves a better accuracy by increasing the polynomial order when the sample size is small. On the other hand, the performance of the benchmark method does not improve as the polynomial order increases when the polynomial order is greater than 6. This is due to the numerical instability by the over-parametrization of using the GS-PCE based on an insufficient number of random observations. The right

subfigure in Figure 4.2 shows the trend that increasing the polynomial order improves the accuracy for both methods given a large sample size. Therefore, we conclude that the proposed method achieves the similar or a better accuracy than the benchmark method given the same number of observations for models with independnet inputs.

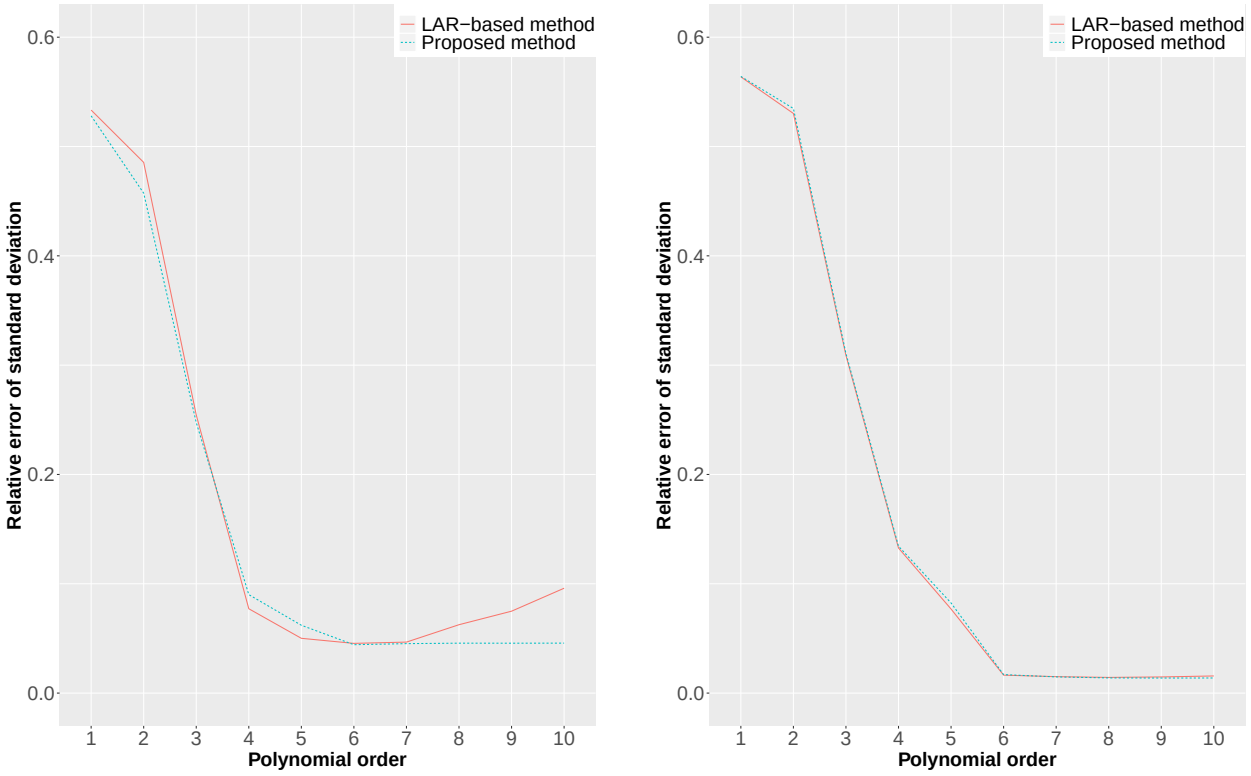


Figure 4.2: The plots show the relative errors of estimating the standard deviation of Y using $m = 100$ (left) and 1000 (right) random observations for both methods. The relative error is averaged across 50 simulation runs for each polynomial order p.

Furthermore, we study how the sample size m affects the estimation accuracy of the proposed method compared with the benchmark method based on a fixed polynomial order $p = 8$. As shown in Figure 4.3, the proposed method achieves a much better accuracy than the benchmark method when the sample size is small. When the sample size is large, both methods perform similarly as expected.

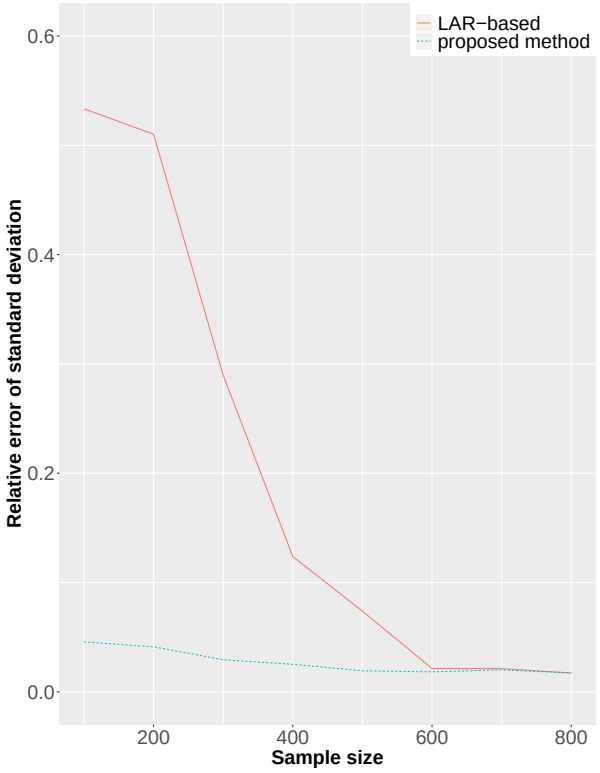


Figure 4.3: Relative errors of estimating the output standard deviation with $p = 8$ v.s. the number of random observations. The relative errors are averaged across 50 simulation runs for each sample size.

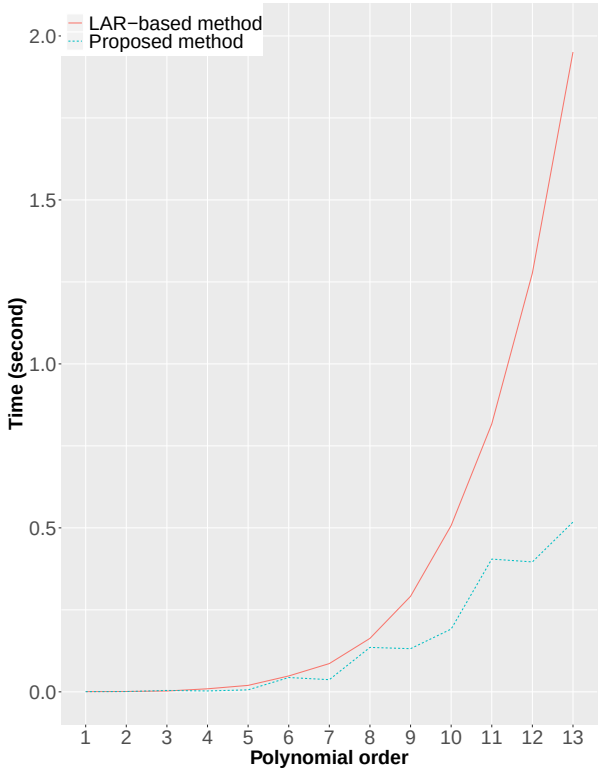


Figure 4.4: Computational time (seconds) v.s. the polynomial order p for both methods. For each polynomial order, the computational time is averaged across 50 simulation runs, where each simulation run uses $1,000$ random observations.

In addition, we compare the computational efficiency of the proposed method with the benchmark method in terms of the computational time. The computational times are recorded using a 1.4 GHz Intel Core i5 machine. The average computation times for both methods are calculated based on 50 simulation runs for the polynomial order of $p = 1$ through $p = 13$ using $1,000$ random observations in each simulation run. As shown in Figure 4.4, the computational time for the benchmark method increases exponentially as the polynomial

order increases. It can be explained by Eq. (4.2) since the number of constructed orthonormal polynomials increases exponentially as the polynomial order p increases. However, the computation time for the proposed method grows much more slowly. This can be intuitively explained by Step 3 in Algorithm 6 since the number of polynomials is reduced in each iteration.

4.5.2 Numerical example with dependent inputs

We use a numerical example in [39] as our second example to validate the use of the proposed method for model with dependent inputs. This example involves multiple types of probability distributions of inputs as follows:

$$\begin{aligned}
\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} \quad &\sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.3 \\ 0 & 0 & 0.3 & 1 \end{pmatrix} \right], \\
X \quad &\sim \mathcal{U}(0,1), \\
X_5 \quad &= \theta_1 X + \mathcal{U}(0,1), \\
X_6 \quad &= \theta_2 X + \theta_3 X^2 + \mathcal{U}(0,1), \\
Y \quad &= X_1 X_2 + X_3 X_4 + X_5 X_6.
\end{aligned} \tag{4.12}$$

Here, we set $(\theta_1, \theta_2, \theta_3) = (0.4, 0.6, 1)$ which is the same as it is used in [39]. We also compare the estimation accuracy of the standard deviation of Y using the two methods. Unlike the first example, where it requires a PCE model with a large p to model the response function, this example uses a PCE model with $p = 2$ for both methods. To this regard, we measure the performance for each method of using $m = 20$ or $m = 100$. The standard deviation of the output is estimated based on the average across 50 simulation runs for both methods using different m. As it is shown in Table 4.1, the proposed method achieves the same accuracy as the benchmark method when $m = 100$. However, the proposed method provides a much better accuracy than the benchmark method when $m = 20$. It is also shown that

the proposed method achieves the similar or a better accuracy than the benchmark method given the same number of observations for models with dependent inputs.

Table 4.1: Estimations of the output standard deviation using the benchmark method and the proposed method across 50 simulation runs. Each simulation run uses a $m = 20$ or $m = 100$. The relative errors are calculated based on the theoretical value $\sigma_Y = 1.655$ provided in [39]. The estimation accuracy of using the proposed method is better than using the benchmark method when $m = 20$.

Sample size	Method	Estimation	Relative error
20	Benchmark method	0.914 ± 0.112	44.77%
	Proposed method	1.618 ± 0.051	2.23%
100	Benchmark method	1.643 ± 0.027	0.73%
	Proposed method	1.643 ± 0.027	0.73%

4.5.3 23-bar horizontal truss

We consider the 23-bar horizontal truss example in [67] as our third example. The downward vertical displacement at the mid span of the structure Y is considered as the output of interest. As depicted in Figure 4.5, the uncertainty of Y is affected by Young modulus $E_i, i = 1, 2$, cross-sectional area $A_i, i = 1, 2$ for both horizontal and diagonal bars, and the random loads $P_i, i = 1, 2, \cdots, 6$. All inputs considered in this example have the same distributions as they are described in [67]. $E_i, i = 1, 2$ and $A_j, j = 1, 2$ are assumed to be mutually independent inputs and following the lognormal distribution with mean μ and standard deviation σ as follows:

$$\begin{aligned} E_1, E_2 &\sim \mathcal{LN}(2.1 \times 10^{11}, 2.1 \times 10^{10}) \text{ [Pa]}, \\ A_1 &\sim \mathcal{LN}(2.0 \times 10^{-3}, 2.0 \times 10^{-4}) \text{ [m}^2\text{]}, \\ A_2 &\sim \mathcal{LN}(1.0 \times 10^{-3}, 1.0 \times 10^{-4}) \text{ [m}^2\text{]}. \end{aligned} \tag{4.13}$$

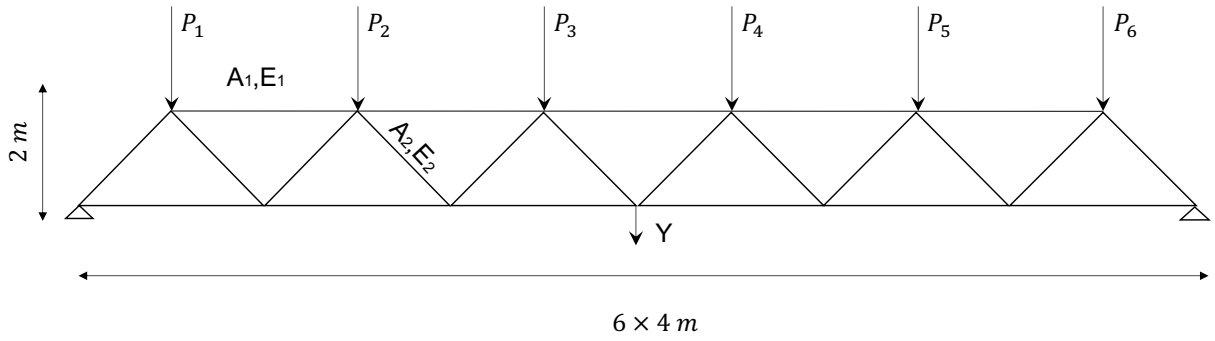


Figure 4.5: Scheme of the horizontal truss model modified from [39]. Young modulus $E_i, i = 1, 2$, cross-sectional area $A_i, i = 1, 2$ for both horizontal and diagonal bars, and the random loads $P_i, i = 1, 2, \cdots, 6$ are the inputs which affect the downward vertical displacement at the mid span of the structure Y.

Unlike $E_i, i = 1, 2$ and $A_i, i = 1, 2$ which are mutually independent, $P_i, i = 1, 2, \cdots, 6$ are mutually dependent on each other. In addition, P_i marginally follows a Gumbel distribution with mean $\mu = 5 \times 10^4$ [N] and standard deviation $\sigma = 7.5 \times 10^3$ [N] as follows:

$$F_i(x; \alpha, \beta) = e^{-e^{-(x-\alpha)}/\beta}, i = 1, 2, \ldots, 6, \tag{4.14}$$

where $\beta = \sqrt{6}\sigma/\pi, \alpha = \mu - \gamma\beta$, and $\gamma \approx 0.5772$ is the Euler-Mascharoni constant. The dependency among $P_i, i = 1, 2, \ldots, 6$ is encoded using the C-vine copula with the density as follows:

$$c_{\boldsymbol{X}}^{(\mathcal{G})}(u_1, \ldots, u_6) = \prod_{j=2}^{6} c_{1j;\theta=1.1}^{(\mathcal{GH})}(u_1, u_j), \tag{4.15}$$

where $c_{1j;\theta=1.1}^{(\mathcal{GH})}$ is the density of the pair-copula between P_1 and P_j, $j = 2, \ldots, d$. $\mathcal{GH}$ represents the Gumbel-Hougaard family whose bivariate copula can be represented as follows:

$$C_\theta^{(\mathcal{GH})}(u, v) = \exp\left(-\left((-\log u)^\theta + (-\log v)^\theta\right)^{1/\theta}\right), \quad \theta \in [1, \infty),$$

here $\boldsymbol{\theta}$ decides the correlations among the loads. Based on Eq. (4.15), we can infer that P_1 is equally correlated with the rest loads. Y is simulated based on a regression of the standardized inputs with coefficients provided in [37] as follows:

$$\begin{aligned} Y = {} & 2.8070 + 1.2598E_1' + 0.2147E_2' + 1.2559A_1' + 0.2133A_2' - 0.1510P_1' - 0.4238P_2' - \\ & 0.6100P_3' - 0.6100P_4' - 0.4238P_5' - 0.1510P_6' - 0.1978E_1'^2 - 0.0362E_2'^2 - 0.2016A_1'^2 - \\ & 0.0346A_2'^2 + 0.0023P_1'^2 + 0.0008P_2'^2 + 0.0036P_3'^2 + 0.0036P_4'^2 + 0.0008P_5'^2 + 0.0023P_6'^2 - \\ & 0.0042E_1'E_2' - 0.3022E_1'A_1' - 0.0110E_1'A_2' + 0.0381E_1'P_1' + 0.0871E_1'P_2' + 0.1232E_1'P_3' + \\ & 0.1232E_1'P_4' + 0.0871E_1'P_5' + 0.0346E_1'P_6' + 0.0041E_2'A_1' + 0.0110A_1'A_2' + 0.0261A_1'P_1' + \\ & 0.0831A_1'P_2' + 0.1172A_1'P_3' + 0.1172A_1'P_4' + 0.0832A_1'P_5' + 0.0296A_1'P_6', \end{aligned} \tag{4.16}$$

where $E_i', i = 1,2$, $A_i', i = 1,2$, and $P_i', i = 1,2,3,4,5,6$ are the standardized inputs. For example, $E_1' = \frac{E-\mu_{E_1}}{\sigma_{E_1}}$, where μ_{E_1} is the mean of E_1 and σ_{E_1} is the standard deviation of E_1.

We estimate the standard deviation of Y for the benchmark method and the proposed method by averaging their estimations over 50 simulation runs, where each simulation run uses $m = 50$ or $m = 500$. As it is shown in Table 4.2, the proposed obtains a better estimation comparing with the benchmark method when the $m = 50$.

4.5.4 HIV model

The HIV model used in [75] is considered as our fourth example to validate the proposed method. The output of interest is the basic reproduction number (R_0), which is arguably regarded as the most important quantity that measures the effectiveness of an infectious disease can spread through a population [20, 24]. R_0 is modeled using a deterministic equation as follows:

$$R_0 = \frac{\beta_0(1-\gamma)\theta_d^2 + \beta_1 n_1 Q_0(n_d - \kappa) + \beta_2 n_2 \alpha Q_0 + (1-\gamma)(\kappa+\alpha)\beta_0\theta_d}{\theta_d(\theta_d+\kappa)(\theta_d+\alpha)}, \tag{4.17}$$

where the inputs following uniform distributions with parameters listed in Table 4.3. In addition, we also assume there exists correlation between β_1 and n_1 as well as β_2 and n_2,

Table 4.2: Estimations of output standard deviation using the benchmark method and the proposed method based on 50 simulation runs. Each simulation run uses $m = 50$ or $m = 100$. The relative errors are calculated based on $\sigma_y = 2.169$, where it is estimated based on a Monte Carlo estimator with 100 simulation runs. Each simulation run uses 10^5 random observations. The proposed method shows a better estimation accuracy than the benchmark method when $m = 50$.

Sample size	Method	Estimation	Relative error
50	Benchmark method	2.101 ± 0.034	3.14%
	Proposed method	2.106 ± 0.034	2.90%
100	Benchmark method	2.156 ± 0.029	0.61%
	Proposed method	2.156 ± 0.029	0.61%

where $\rho_{\beta_1,n_1} = 0.3$ and $\rho_{\beta_2,n_2} = 0.5$.

In this example, we set $p = 4$ for both the benchmark method and the proposed method. The benchmark method requires at least $2,000$ random observations for 10 random inputs to keep the orthogonality of the constructed orthonormal polynomials using the modified Gram-Schmidt algorithm. The lack of orthogonality of the constructed orthonormal polynomials causes inaccurate estimation of the standard deviation of the output. As it is shown in Table 4.4, the benchmark method does not provide a reasonable estimation on the approximating the output standard deviation. However, our proposed method still obtains a accurate estimation on the output standard deviation based on the small sample of random observations (m=200). This also reflects the fact that our proposed method reduces the number of random observations to construct a sparse PCE for models with dependent inputs.

Table 4.3: Input descriptions and distributions of the HIV model.

Input	Descriptions	Distribution
Q_0	Recruitment rate	$U(0.0261, 0.0319)$
β_0	Birth rate of infective	$U(0.027, 0.033)$
γ	Fraction of susceptible newborn from infective class	$U(0.36, 0.44)$
β_1	Contact rate of susceptible with asymptomatic infective	$U(0.18, 0.22)$
β_2	Contact rate of susceptible with symptomatic infective	$U(0.072, 0.088)$
n_1	Number of sexual partners of susceptible with asymptomatic infective	$U(1.8, 2.2)$
n_2	Number of sexual partners of susceptible with symptomatic infective	$U(1.8, 2.2)$
θ_d	Natural death rate	$U(0.018, 0.022)$
α	Removal rate of symptomatic class	$U(0.54, 0.66)$
κ	Rate of development to AIDs	$U(0.09, 0.11)$

Table 4.4: Estimations of the output standard deviation using the benchmark method and the proposed method across 50 simulation runs. Each simulation run uses a $m = 200$. The relative errors are calculated based on the theoretical value $\sigma_Y = 0.252$ provided in [75]. The proposed method provides a more accurate estimation than the benchmark method.

Method	Sample size	Estimation	Relative error
Benchmark method	200	1034.934 ± 585.089	$> 100\%$
Proposed method		0.258 ± 0.002	2.33%

4.6 Conclusion

In this chapter, we propose a data-driven sparse PCE for models with dependent inputs without requiring any distribution information about the inputs and large number of random observations. Four simulations are used to validate the use of the proposed algorithm. It has shown that the proposed method provides a accurate estimation on the output standard

deviation using a small sample size of random observation and improves the computational efficiency comparing with the benchmark method on constructing a sparse PCE.

Sensitivity study using PCE for models with dependent inputs has been explored in the recent literature [39]. It is regarded as the pioneering work on providing interpretable sensitivity indices for models with dependent inputs without imposing distribution information or dependence structure about the inputs. Even though the proposed method improves the numerical accuracy and computational efficiency of using PCE to model the relationship between the inputs and the output, the PCE coefficients estimated using the proposed method might not be directly used to estimate the sensitivity indices for the dependent inputs. This suggests a future research direction on estimating the sensitivity indices proposed in [39] using the proposed sparse PCE.

Chapter 5

CONCLUSION AND FUTURE RESEARCH

This dissertation proposes three data-driven PCEs to address uncertainty quantification problems in real-world applications. The three proposed methods improve the efficiency on identifying influential inputs and modeling uncertainty propagation for models with dependent inputs by reducing the number of required random observations without imposing strong assumptions on the inputs. Specifically, Chapter 2 and Chapter 4 aim to improve the efficiency of variance-based sensitivity analysis from two different aspects. On the one hand, Chapter 2 proposes the SN-PCE to model uncertainty propagation in a broad class of processes represented as DAGs, where the inputs are assumed to be independent. Unlike the existing method, the proposed SN-PCE model exploits the relationship structure of a networked process by recursively relating a network node to its direct predecessors to trace the output variance back to the inputs. It, thereby, estimates the Sobol indices, which measure the influence of each input on the output variance, accurately and efficiently. Two manufacturing processes and a flooding process are used to empirically validate the SN-PCE model. In addition, Chapter 2 also provides the theoretical validation of the SN-PCE model in the sense that its prediction of the output converges in probability to the true output under reasonable assumptions. On the other hand, Chapter 4 proposes a data-driven sparse PCE to model uncertainty propagation for models with dependent inputs. The existing sparse PCE models apply sparse operator directly on the constructed orthonormal polynomials. However, this procedure is inefficient when the inputs are dependent and the constructed orthonormal polynomials are built from the Gram-Schmidt algorithm. The proposed sparse PCE model improves the efficiency of such procedure by iteratively constructing the most correlated orthonormal polynomials with the output and removing the uncorrelated or redun-

dant polynomials. Two numerical examples and two real-world applications including a truss structure model and a HIV model have shown that the proposed method can significantly improve upon the efficiency of the existing sparse PCE methods.

Besides addressing the challenge of improving the efficiency of propagating uncertainty with a limited number of random observations, Chapter 3 proposes interpretable data-driven sensitivity indices for models with dependent inputs without imposing strong assumptions on the dependence structure of the inputs. Such sensitivity indices can help better understand how the variance of the output in a real-world application is propagated from dependent inputs.

This dissertation research has identified several future research directions to improve variance-based sensitivity analysis for a broader class of real-world applications. For example, Chapter 2 only considers a system that can be represented as a DAG, where inputs are independent. One direction could be to relax the assumption of mutual independence of the inputs. While their dependencies would complicate the estimation and interpretation of the sensitivity indices, the indices proposed in Chapter 3 might help decode how the dependent inputs in a network influence the output. In addition, Chapter 2 builds a SN-PCE model to improve the efficiency on propagating the uncertainty of the output. However, the efficiency of the SN-PCE model may be further improved by using the sparse PCE model proposed in Chapter 4. Another research direction would be to allow cycles (or, feedback loops) in the network in Chapter 2, thereby, extending this work beyond directed acyclic networks. Lastly, estimating sensitivity indices for dependent inputs in Chapter 3 using the proposed sparse PCE model in Chapter 4 warrants further investigations.

Appendix A

APPENDIX FOR CHAPTER 2

A.1 Nomenclature

E collection of all the directed edges in G.

G directed acyclic graph representing the network-structured process of interest.

L $\inf\{l \in \mathbb{N} : \mathscr{P}^l(y) = \mathscr{P}^{l+1}(y)\}$ for the output y.

$S(G)$ source nodes in G.

ST_{x_j} total Sobol index of the output y with respect to the input x_j.

S_{x_j} first-order Sobol index of the output y with respect to the input x_j.

V collection of all the nodes in G.

Y_j output of the j^{th} observation, $j = 1, \ldots, m$.

$\boldsymbol{X}_j$ input vector of the j^{th} observation, $j = 1, \ldots, m$.

$\boldsymbol{\xi}$ network inputs that influence the output y.

$\boldsymbol{x}_{S(G)}$ network inputs.

$\boldsymbol{x_v}$ $(x_{v_i})_{v_i \in \boldsymbol{v}}$ for $\boldsymbol{v} \subseteq V$.

$\mathscr{P}(\boldsymbol{x_v})$ $\boldsymbol{x}_{\left\{v_i \in V : v_j \in \boldsymbol{v}, \langle v_i, v_j \rangle \in E\right\} \cup (\boldsymbol{v} \cap S(G))}$.

$\mathscr{P}^l(\boldsymbol{x_v})$ $\mathscr{P}(\cdot)$ applied on $\boldsymbol{x_v}$ $l \in \mathbb{N}$ times.

x_{v_i} random variable represented by the node $v_i \in V$.

A.2 Proof of Theorem 3

Proof. The proof is based on mathematical induction.

Base case (iteration $l = 1$). Let $\gamma^{(0)}$ be an identify function and $P^{(0)} := P^{(1)}$ so that $p^{(1)} = \gamma^{(0)}\left(P^{(0)}\right) = P^{(1)}$. Based on Theorem 11 in [55],

$$\hat{y}^{(1)} = \sum_{i=0}^{P^{(1)}} \theta_i^{(1)} \psi_i^{(1)} \left(\mathscr{P}(y)\right)$$

in eq. (2.12) with $l = 1$ converges to $y\left(\boldsymbol{x}_{\boldsymbol{v}^{(1)}}\right) \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ in probability as $p^{(1)} \to \infty$, or equivalently, $P^{(1)} \to \infty$.

Inductive hypothesis. For $l \in \mathbb{N}$, suppose there exists an increasing function $\gamma^{(l-1)} : \mathbb{N} \mapsto \mathbb{N}$ such that if $p^{(l)} = \gamma^{(l-1)}\left(P^{(l-1)}\right)$, $\hat{y}^{(l)}$ in eq. (2.12) converges to y in probability as $P^{(1)} \to \infty$.

Inductive step. We want to show that there exists an increasing function $\gamma^{(l)} : \mathbb{N} \mapsto \mathbb{N}$ such that if $p^{(l+1)} = \gamma^{(l)}\left(P^{(l)}\right)$, $\hat{y}^{(l+1)}$ in eq. (2.16) converges to y in probability as $P^{(1)} \to \infty$.

To prove $\hat{y}^{(l+1)} \xrightarrow{\mathcal{P}} y$ under the aforementioned conditions, we show that for any $\epsilon > 0$ and any $\delta > 0$, there exists $P_0^{(l)}$ such that $\forall P^{(l)} \geq P_0^{(l)}$,

$$\begin{aligned} \mathcal{P}\left(\left|\hat{y}^{(l+1)} - y\right| > \epsilon\right) &= \mathcal{P}\left(\left|\sum_{i=0}^{P^{(l)}} \theta_i^{(l)} \hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right) - y\right| > \epsilon\right) \\ &< \delta. \end{aligned} \tag{A.1}$$

Note that $P^{(l)}$ is an increasing function of $p^{(l')}, l' = 1, \ldots, l$. The right-hand side of eq.(A.1) can be rewritten as

$$\begin{aligned} &\mathcal{P}\left(\left|\sum_{i=0}^{P^{(l)}} \theta_i^{(l)} \hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right) - y\right| > \epsilon\right) \\ &= \mathcal{P}\left(\left|\sum_{i=0}^{P^{(l)}} \theta_i^{(l)} \hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right) - \sum_{i=0}^{P^{(l)}} \theta_i^{(l)} \psi_i^{(l)}\left(\mathscr{P}^l(y)\right)\right.\right. \\ &\quad \left.\left. + \sum_{i=0}^{P^{(l)}} \theta_i^{(l)} \psi_i^{(l)}\left(\mathscr{P}^l(y)\right) - y\right| > \epsilon\right) \end{aligned}$$

$$\leq \mathcal{P}\left(\left|\sum_{i=0}^{P^{(l)}}\theta_i^{(l)}\hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right)-\sum_{i=0}^{P^{(l)}}\theta_i^{(l)}\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)\right|>\frac{\epsilon}{2}\right)$$
$$+\mathcal{P}\left(\left|\sum_{i=0}^{P^{(l)}}\theta_i^{(l)}\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)-y\right|>\frac{\epsilon}{2}\right) \tag{A.2}$$

by the triangular inequality. By the inductive hypothesis, there exists $P_1^{(l)}$ such that $\forall P^{(l)} \geq P_1^{(l)}$, the second term in eq. (A.2) satisfies

$$\mathcal{P}\left(\left|\sum_{i=0}^{P^{(l)}}\theta_i^{(l)}\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)-y\right|>\frac{\epsilon}{2}\right)<\delta/2. \tag{A.3}$$

The first term in eq. (A.2) can be upper-bounded using the triangular inequality as follows:

$$\mathcal{P}\left(\left|\sum_{i=0}^{P^{(l)}}\theta_i^{(l)}\hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right)-\sum_{i=0}^{P^{(l)}}\theta_i^{(l)}\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)\right|>\frac{\epsilon}{2}\right)$$
$$\leq \sum_{i=0}^{P^{(l)}}\mathcal{P}\left(\left|\hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right)-\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)\right|>\frac{\epsilon}{2}\right). \tag{A.4}$$

Note that $P_{ij}^{(l)}$ in eq. (2.14) is an increasing function of the highest polynomial order $p_{ij}^{(l)}$, which is assumed to be the constant $p^{(l+1)}$ in Sec. III.C. By Lemma 2, there exists an increasing function $\gamma^{(l)}:\mathbb{N}\mapsto\mathbb{N}$ such that $\forall p^{(l+1)} \geq \gamma^{(l)}\left(P^{(l)}\right)$, the summand in eq. (A.4) satisfies

$$\mathcal{P}\left(\left|\theta_i^{(l)}\right|\left|\hat{\psi}_i^{(l)}\left(\mathscr{P}^l(y)\right)-\psi_i^{(l)}\left(\mathscr{P}^l(y)\right)\right|>\frac{\epsilon}{2}\right)<\frac{\delta}{2\left(P^{(l)}+1\right)}$$

for $i=0,1,\ldots,P^{(l)}$, and for any $P^{(l)}\in\mathbb{N}$. Therefore, eq. (A.4) is upper-bounded by $\delta/2$, $\forall p^{(l+1)} \geq \gamma^{(l)}\left(P^{(l)}\right)$. Putting this together with eq. (A.3) proves the existence of $P_0^{(l)} := P_1^{(l)}$ such that $\forall P^{(l)} \geq P_0^{(l)}$, eq. (A.1) is upper-bounded by δ, completing the proof of $\hat{y}^{(l+1)} \xrightarrow{\mathcal{P}} y$ as $P^{(l)}\to\infty$ if $p^{(l+1)}=\gamma^{(l)}\left(P^{(l)}\right)$. By mathematical induction, the conclusion of the theorem follows. □

Appendix B

APPENDIX FOR CHAPTER 3

B.1 List of sensitivity indices

$ST^u_{X_i}$ Total *uncorrelated* sensitivity index of the output Y with respect to the input X_i.

ST_{X_i} Total Sobol index of the output Y with respect to the input X_i.

$S^u_{X_i}$ First-order *uncorrelated* sensitivity index of the output Y with respect to the input X_i.

S_{X_i} First-order Sobol index of the output Y with respect to the input X_i.

$\bar{S}_{X_i}$ First-order *full* sensitivity index of the output Y with respect to the input X_i.

$\overline{ST}_{X_i}$ Total *full* sensitivity index of the output Y with respect to the input X_i.

$\tilde{S}_{k|1,2,\ldots,k-1}$ The k^{th} order sensitivity index of the output Y with respect to the inputs $\boldsymbol{X}$ after taking account of the first order sensitivity index through the $(k-1)^{th}$ order sensitivity index.

B.2 Analytical method for calculating the sensitivity indices in Example 3 in Section 3.5

The following lemma is used for the analytical method in Example 3 in Section 3.5.

Lemma B.2.1. *Suppose*

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \right].$$

Then, $Var(X_1X_2) = \mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2 + 2\mu_1\mu_2\rho\sigma_1\sigma_2 + \rho^2\sigma_1^2\sigma_2^2$.

Proof. We can express X_1 and X_2 as follows:

$$X_1 = \mu_1 + r\sigma_1 Z + (1-r^2)^{\frac{1}{2}}\sigma_1 Y_1,$$
$$X_2 = \mu_2 + r\sigma_2 Z + (1-r^2)^{\frac{1}{2}}\sigma_2 Y_2,$$

where $r = \sqrt{\rho}$, $Z \sim \mathcal{N}(0,1)$, and $Y_i \sim \mathcal{N}(0,1), i = 1,2$. We have the covariance among Z and $Y_i, i = 1,2$ as follows:

$$Cov(Z, Y_i) = 0, \text{ for } i = 1,2,$$
$$Cov(Y_1, Y_2) = 0.$$

Therefore, we have

$$\begin{aligned} Var(X_1X_2) &= E(X_1^2X_2^2) - [E(X_1X_2)]^2 \\ &= E(X_1^2)E(X_2^2) + Cov(X_1^2, X_2^2) - [E(X_1)E(X_2) + Cov(X_1, X_2)]^2 \\ &= (\sigma_1^2 + \mu_1^2)(\sigma_2^2 + \mu_2^2) + Cov(r^2\sigma_1^2Z^2 + 2\mu_1 r\sigma_1 Z, r^2\sigma_2^2Z^2 + 2\mu_2 r\sigma_2 Z) - \\ &\quad (\mu_1\mu_2 + r^2\sigma_1\sigma_2)^2 \\ &= \mu_1^2\sigma_2^2 + \mu_2^2\sigma_1^2 + \sigma_1^2\sigma_2^2 + 2\mu_1\mu_2\rho\sigma_1\sigma_2 + \rho^2\sigma_1^2\sigma_2^2. \end{aligned}$$

□

We now present how to analytically calculate the sensitivity indices in Example 3. From (3.9), $\{X_1, X_2\}$, $\{X_3, X_4\}$, and $\{X_5, X_6\}$ are mutually independent. We have

$$\begin{aligned} Var(Y) &= Var(X_1X_2 + X_3X_4 + X_5X_6) \\ &= Var(X_1X_2) + Var(X_3X_4) + Var(X_5X_6). \end{aligned}$$

Based on Lemma B.2.1, we can easily obtain $Var(X_1X_2)$ and $Var(X_3X_4)$. As for $Var(X_5X_6)$, X_5 and X_6 can be expressed as follows:

$$X_5 = \theta_1 U_1 + U_2,$$
$$X_6 = \theta_2 U_1 + \theta_3 U_1^2 + U_3,$$

where $U_i \sim U(0,1), i = 1,2,3$ and U_i's are mutually independent for $i = 1,2,3$. Because

$$\begin{aligned} Var(X_5X_6) &= E(X_5^2X_6^2) - [E(X_5X_6)]^2 \\ &= Cov(X_5^2, X_6^2) + E(X_5^2)E(X_6^2) - [Cov(X_5, X_6) + E(X_5)E(X_6)]^2, \end{aligned}$$

using the property that U_i's are mutually independent for $i = 1,2,3$, it is straightforward to express $Var(X_5X_6)$ as a function of $(\theta_1,\theta_2,\theta_3)$ and the moments of $U_i, i = 1,2,3$.

After calculating $Var(X_1X_2)$, $Var(X_3X_4)$, and $Var(X_5X_6)$, we can calculate $ST_{\{X_1X_2\}}$, $ST_{\{X_3X_4\}}$, and $ST_{\{X_5X_6\}}$ as follows:

$$ST_{\{X_1X_2\}} = \frac{Var(X_1X_2)}{Var(X_1X_2)+Var(X_3X_4)+Var(X_5X_6)},$$
$$ST_{\{X_3X_4\}} = \frac{Var(X_3X_4)}{Var(X_1X_2)+Var(X_3X_4)+Var(X_5X_6)},$$
$$ST_{\{X_5X_6\}} = \frac{Var(X_5X_6)}{Var(X_1X_2)+Var(X_3X_4)+Var(X_5X_6)}.$$

B.3 The correlation matrix for the loads in Example 4 in Section 3.5

The correlation matrix for the loads listed below is estimated using 10^6 random observations.

Table B.1: A correlation matrix for the six loads estimated based on 10^6 random observations generated based on the C-vine copula in Eq. (3.11).

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	1.000	0.172	0.173	0.171	0.176	0.173
P_2	0.172	1.000	0.032	0.031	0.033	0.032
P_3	0.173	0.032	1.000	0.032	0.034	0.031
P_4	0.171	0.031	0.032	1.000	0.033	0.031
P_5	0.176	0.033	0.034	0.033	1.000	0.032
P_6	0.173	0.032	0.031	0.031	0.032	1.000

B.4 The response surface model used for simulating the output Y in Example 4 in Section 3.5

The response surface model used to simulate the output Y is provided in [37] as follows:

$$\begin{aligned} Y = {} & 2.8070 + 1.2598E_1' + 0.2147E_2' + 1.2559A_1' + 0.2133A_2' - 0.1510P_1' - 0.4238P_2' - \\ & 0.6100P_3' - 0.6100P_4' - 0.4238P_5' - 0.1510P_6' - 0.1978E_1'^2 - 0.0362E_2'^2 - 0.2016A_1'^2 - \\ & 0.0346A_2'^2 + 0.0023P_1'^2 + 0.0008P_2'^2 + 0.0036P_3'^2 + 0.0036P_4'^2 + 0.0008P_5'^2 + 0.0023P_6'^2 - \\ & 0.0042E_1'E_2' - 0.3022E_1'A_1' - 0.0110E_1'A_2' + 0.0381E_1'P_1' + 0.0871E_1'P_2' + 0.1232E_1'P_3' + \\ & 0.1232E_1'P_4' + 0.0871E_1'P_5' + 0.0346E_1'P_6' + 0.0041E_2'A_1' + 0.0110A_1'A_2' + 0.0261A_1'P_1' + \\ & 0.0831A_1'P_2' + 0.1172A_1'P_3' + 0.1172A_1'P_4' + 0.0832A_1'P_5' + 0.0296A_1'P_6', \end{aligned} \tag{B.1}$$

where $E_i', i = 1,2$, $A_i', i = 1,2$, and $P_i', i = 1,2,3,4,5,6$ are the standardized inputs. For example, $E_1' = \frac{E-\mu_{E_1}}{\sigma_{E_1}}$, where μ_{E_1} is the mean of E_1 and σ_{E_1} is the standard deviation of E_1.

BIBLIOGRAPHY

[1] © 2020 IEEE. Reprinted, with permission, from Z. Liu, A. G. Banerjee, and Y. Choe. Identifying the influential inputs for network output variance using sparse polynomial chaos expansion. *IEEE Transactions on Automation Science and Engineering*, pages 1–11, 2020.

[2] Pragati Agrawal and Shrisha Rao. Energy-aware scheduling of distributed systems. *IEEE Transactions on Automation Science and Engineering*, 11(4):1163–1175, 2014.

[3] Alexander Avdonin, Stefan Jaensch, Camilo F Silva, Matic Češnovar, and Wolfgang Polifke. Uncertainty quantification and sensitivity analysis of thermoacoustic stability with non-intrusive polynomial chaos expansion. *Combustion and Flame*, 189:300–310, 2018.

[4] Marc Berveiller, Bruno Sudret, and Maurice Lemaire. Stochastic finite element: a non intrusive approach by regression. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 15(1-3):81–92, 2006.

[5] Åke Björck. Numerics of Gram-Schmidt orthogonalization. *Linear Algebra and its Applications*, 197:297–316, 1994.

[6] Åke Björck and Chris C Paige. Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm. *SIAM Journal on Matrix Analysis and Applications*, 13(1):176–190, 1992.

[7] Géraud Blatman and Bruno Sudret. Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach. *Comptes Rendus Mécanique*, 336(6):518–523, 2008.

[8] Géraud Blatman and Bruno Sudret. Efficient computation of global sensitivity indices using sparse polynomial chaos expansions. *Reliability Engineering & System Safety*, 95(11):1216–1229, 2010.

[9] Géraud Blatman and Bruno Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics*, 230(6):2345–2367, 2011.

[10] Robert H Cameron and William T Martin. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Annals of Mathematics*, pages 385–392, 1947.

[11] CJ Carstens. Motifs in directed acyclic networks. In *2013 International Conference on Signal-Image Technology & Internet-Based Systems*, pages 605–611. IEEE, 2013.

[12] Gaëlle Chastaing, Fabrice Gamboa, and Clémentine Prieur. Generalized Hoeffding-Sobol decomposition for dependent variables-application to sensitivity analysis. *Electronic Journal of Statistics*, 6:2420–2448, 2012.

[13] Po-Chen Chen, Vuk Malbasa, Yimai Dong, and Mladen Kezunovic. Sensitivity analysis of voltage sag based fault location with distributed generation. *IEEE Transactions on Smart Grid*, 6(4):2098–2106, 2015.

[14] Kai Cheng and Zhenzhou Lu. Sparse polynomial chaos expansion based on d-morph regression. *Applied Mathematics and Computation*, 323:17–30, 2018.

[15] Thierry Crestaux, Olivier Le Maitre, and Jean Marc Martinez. Polynomial chaos expansion for sensitivity analysis. *Reliability Engineering & System Safety*, 94(7):1161–1172, 2009.

[16] G Deman, Katerina Konakli, Bruno Sudret, Jaouher Kerrou, Pierre Perrochet, and Hakim Benabderrahmane. Using sparse polynomial chaos expansions for the global sensitivity analysis of groundwater lifetime expectancy in a multi-layered hydrogeological model. *Reliability Engineering & System Safety*, 147:156–169, 2016.

[17] Thomas J. DiCiccio and Bradley Efron. Bootstrap confidence intervals. *Statistical Science*, 11(3):189–228, 1996.

[18] Dragan Djurdjanovic and Jun Ni. Stream-of-variation (Sov)-based measurement scheme analysis in multistation machining systems. *IEEE Transactions on Automation Science and Engineering*, 3(4):407–422, 2006.

[19] M Fesanghary, E Damangir, and I Soleimani. Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search algorithm. *Applied Thermal Engineering*, 29(5):1026–1031, 2009.

[20] Christophe Fraser, Christl A Donnelly, Simon Cauchemez, William P Hanage, Maria D Van Kerkhove, T Déirdre Hollingsworth, Jamie Griffin, Rebecca F Baggaley, Helen E Jenkins, Emily J Lyons, et al. Pandemic potential of a strain of influenza a (h1n1): early findings. *science*, 324(5934):1557–1561, 2009.

[21] Luc Giraud, Julien Langou, Miroslav Rozložník, and Jasper van den Eshof. Rounding error analysis of the classical gram-schmidt orthogonalization process. *Numerische Mathematik*, 101(1):87–100, 2005.

[22] Ling Guo, Akil Narayan, and Tao Zhou. A gradient enhanced l1-minimization for sparse approximation of polynomial chaos expansions. *Journal of Computational Physics*, 367:49–64, 2018.

[23] Kang He, Minping Jia, and Qingsong Xu. Optimal sensor deployment for manufacturing process monitoring based on quantitative cause-effect graph. *IEEE Transactions on Automation Science and Engineering*, 13(2):963–975, 2016.

[24] Petter Holme and Naoki Masuda. The basic reproduction number as a predictor for epidemic outbreaks in temporal networks. *PloS one*, 10(3), 2015.

[25] Toshimitsu Homma and Andrea Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1):1–17, 1996.

[26] Shuai Huang, Jing Li, Jieping Ye, Adam Fleisher, Kewei Chen, Teresa Wu, Eric Reiman, Alzheimer's Disease Neuroimaging Initiative, et al. A sparse structure learning algorithm for Gaussian Bayesian network identification from high-dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1328–1342, 2013.

[27] Wenzhen Huang and Zhenyu Kong. Process capability sensitivity analysis for design evaluation of multistage assembly processes. *IEEE Transactions on Automation Science and Engineering*, 7(4):736–745, 2010.

[28] Bertrand Iooss and Paul Lemaître. A review on global sensitivity analysis methods. In *Uncertainty Management in Simulation-Optimization of Complex Systems*, pages 101–122. Springer, 2015.

[29] Tsutomu Ishigami and Toshimitsu Homma. An importance quantification technique in uncertainty analysis for computer models. In *[1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis*, pages 398–403. IEEE, 1990.

[30] Julien Jacques, Christian Lavergne, and Nicolas Devictor. Sensitivity analysis in presence of model uncertainty and correlated inputs. *Reliability Engineering & System Safety*, 91(10):1126–1134, 2006.

[31] John D Jakeman, Michael S Eldred, and Khachik Sargsyan. Enhancing $\ell 1$-minimization estimates of polynomial chaos expansions using basis selection. *Journal of Computational Physics*, 289:18–34, 2015.

[32] Z Kala. Sensitivity and reliability analyses of lateral-torsional buckling resistance of steel beams. *Archives of Civil and Mechanical Engineering*, 15(4):1098–1107, 2015.

[33] Zdeněk Kala and Libor Puklickỳ. Variance-based methods for sensitivity analysis in civil engineering. In *Computational Structural Engineering*, pages 991–997. Springer, 2009.

[34] Gaurav Kewlani, Justin Crawford, and Karl Iagnemma. A polynomial chaos approach to the analysis of vehicle dynamics under uncertainty. *Vehicle System Dynamics*, 50(5):749–774, 2012.

[35] Sergei Kucherenko, Stefano Tarantola, and Paola Annoni. Estimation of global sensitivity indices for models with dependent variables. *Computer Physics Communications*, 183(4):937–946, 2012.

[36] Loïc Le Gratiet, Stefano Marelli, and Bruno Sudret. Metamodel-based sensitivity analysis: polynomial chaos expansions and Gaussian processes. In Roger Ghanem, David Higdon, and Houman Owhadi, editors, *Handbook of Uncertainty Quantification*, pages 1289–1325. Springer International Publishing, 2017.

[37] Sang Hoon Lee and Byung Man Kwak. Response surface augmented moment method for efficient reliability analysis. *Structural Safety*, 28(3):261–272, 2006.

[38] Sisi Li, Kun Deng, and MengChu Zhou. Sensitivity analysis for building energy simulation model calibration via algorithmic differentiation. *IEEE Transactions on Automation Science and Engineering*, 14(2):905–914, 2017.

[39] Zhanlin Liu and Youngjun Choe. Data-driven sensitivity indices for models with dependent inputs using the polynomial chaos expansion. *arXiv preprint arXiv:1803.10978*, 2018.

[40] Thierry A Mara and Stefano Tarantola. Variance-based sensitivity indices for models with dependent inputs. *Reliability Engineering & System Safety*, 107:115–121, 2012.

[41] Thierry A Mara, Stefano Tarantola, and Paola Annoni. Non-parametric methods for global sensitivity analysis of model output with dependent inputs. *Environmental Modelling & Software*, 72:173–183, 2015.

[42] Stefano Marelli and Bruno Sudret. An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis. *Structural Safety*, 75:67–74, 2018.

[43] Jeremy A Marvel. Performance metrics of speed and separation monitoring in shared workspaces. *IEEE Transactions on Automation Science and Engineering*, 10(2):405–414, 2013.

[44] Stijn Meganck, Philippe Leray, and Bernard Manderick. Learning causal Bayesian networks from observations and experiments: A decision theoretic approach. In *International Conference on Modeling Decisions for Artificial Intelligence*, pages 58–69. Springer, 2006.

[45] DG Mogale, Sri Krishna Kumar, and Manoj Kumar Tiwari. Two stage Indian food grain supply chain network transportation-allocation model. *IFAC-PapersOnLine*, 49(12):1767–1772, 2016.

[46] EE Myshetskaya et al. Monte carlo estimators for small sensitivity indices. *Monte Carlo Methods and Applications mcma*, 13(5-6):455–465, 2008.

[47] Saideep Nannapaneni and Sankaran Mahadevan. Uncertainty quantification in performance evaluation of manufacturing processes. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 996–1005. IEEE, 2014.

[48] Maria Navarro, Jeroen Witteveen, and Joke Blom. Polynomial chaos expansion for general multivariate distributions with correlated variables. *arXiv:1406.5483*, pages 1–24, 2014.

[49] Fei Ni, Michiel Nijhuis, Phuong H Nguyen, and Joseph FG Cobben. Variance-based global sensitivity analysis for power systems. *IEEE Transactions on Power Systems*, 33(2):1670–1682, 2018.

[50] S Oladyshkin and W Nowak. Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, 106:179–190, 2012.

[51] Art B Owen. Better estimation of small Sobol' sensitivity indices. *ACM Transactions on Modeling and Computer Simulation*, 23(2):11–17, 2013.

[52] Ji Peng, Jerrad Hampton, and Alireza Doostan. On polynomial chaos expansion via gradient-enhanced ℓ1-minimization. *Journal of Computational Physics*, 310:440–458, 2016.

[53] M Per Pettersson, Gianluca Iaccarino, and J Nordstrom. *Polynomial chaos methods for hyperbolic partial differential equations*. Springer, 2015.

[54] Robert J Prill, Pablo A Iglesias, and Andre Levchenko. Dynamic properties of network motifs contribute to biological network organization. *PLoS Biology*, 3(11):e343, 2005.

[55] Sharif Rahman. A polynomial chaos expansion in dependent random variables. *Journal of Mathematical Analysis and Applications*, 464(1):749–775, 2018.

[56] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[57] D Garcia Sanchez, Bruno Lacarrière, Marjorie Musy, and Bernard Bourges. Application of sensitivity analysis in building energy simulations: Combining first- and second-order elementary effects methods. *Energy and Buildings*, 68:741–750, 2014.

[58] Qian Shao, Anis Younes, Marwan Fahs, and Thierry A Mara. Bayesian sparse polynomial chaos expansion for global sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 318:474–496, 2017.

[59] Ilya M Sobol. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1(4):407–414, 1993.

[60] Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. In P. Spanos and G. Deodatis, editors, *Proc. 5th Int. Conf. on Comp. Stoch. Mech (CSM5)*, Rhodos, Greece, June 2006.

[61] Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008.

[62] Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008.

[63] Bruno Sudret. Meta-models for structural reliability and uncertainty quantification. *arXiv preprint arXiv:1203.2062*, 2012.

[64] Bruno Sudret. Meta-models for structural reliability and uncertainty quantification. In *"Asian-Pacific Symposium on Structural Reliability and its Applications"*, pages 1–24, Singapore, 2012.

[65] Stefano Tarantola and Thierry A Mara. Variance-based sensitivity indices of computer models with dependent inputs: The Fourier amplitude sensitivity test. *International Journal for Uncertainty Quantification*, 7(6):511–523, 2017.

[66] J-Y Tissot and Clémentine Prieur. A randomized orthogonal array-based procedure for the estimation of first-and second-order sobol'indices. *Journal of Statistical Computation and Simulation*, 85(7):1358–1381, 2015.

[67] Emiliano Torre, Stefano Marelli, Paul Embrechts, and Bruno Sudret. Data-driven polynomial chaos expansion for machine learning regression. *Journal of Computational Physics*, 388:601–623, 2019.

[68] Xiaoliang Wan and George Em Karniadakis. Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM Journal on Scientific Computing*, 28(3):901–928, 2006.

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

[70] Jeroen AS Witteveen, Sunetra Sarkar, and Hester Bijl. Modeling physical uncertainties in dynamic stall induced fluid–structure interaction of turbine blades using arbitrary polynomial chaos. *Computers & Structures*, 85(11):866–878, 2007.

[71] C. F. Jeff Wu and Michael S Hamada. *Experiments: planning, analysis, and optimization*. John Wiley & Sons, 2011.

[72] Dongbin Xiu and George Em Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[73] Jun Xu and Fan Kong. A cubature collocation based sparse polynomial chaos expansion for efficient structural reliability analysis. *Structural Safety*, 74:24–31, 2018.

[74] Kaichao Zhang, Zhenzhou Lu, Lei Cheng, and Fang Xu. A new framework of variance based global sensitivity analysis for models with correlated inputs. *Structural Safety*, 55:1–9, 2015.

[75] Yueying Zhu, Qiuping A Wang, Wei Li, and Xu Cai. Analytic uncertainty and sensitivity analysis of models with input correlations. *Physica A: Statistical Mechanics and its Applications*, 494:140–162, 2018.