A hybrid polynomial chaos expansion – Gaussian process regression method for Bayesian uncertainty quantification and sensitivity analysis

Computer Methods in Applied Mechanics and Engineering 436 (2025) 117693

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

A hybrid polynomial chaos expansion – Gaussian process regression method for Bayesian uncertainty quantification and sensitivity analysis

Paolo Manfredi *

Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi 24, Turin 10129, Italy

ARTICLE INFO

Keywords:
Gaussian process
Kernel method
Kriging
Machine learning
Polynomial chaos expansion
Sensitivity analysis
Surrogate modeling
Uncertainty quantification

ABSTRACT

This paper introduces a novel hybrid method for uncertainty quantification (UQ) combining the benefits of polynomial chaos expansion (PCE) and Gaussian process regression (GPR). The proposed method features a GPR formulation that leverages special implicit kernels involving an infinite sequence of some of the orthogonal polynomials from the Wiener-Askey scheme. These kernels enable the closed-form calculation of PCE coefficients by analytical integration of the GPR posterior, thereby leading to a Bayesian estimation in terms of both expected value and covariance matrix. Notably, the Bayesian definition allows associating confidence information to the computed coefficients, which is then propagated to the classical closed-form estimates associated to PCEs, i.e., first- and second-order moments and Sobol' sensitivity indices. The advocated method helps mitigate some long-standing shortcomings of PCEs in terms of training efficiency and scalability to higher dimensions, while providing an accurate quantification of the intrinsic model uncertainty. Moreover, it allows for a nonparametric computation of PCE coefficients, since the basis functions do not need to be selected a priori. A simple and effective multi-output formulation, involving the tuning of a single set of hyperparameters, is also discussed. The hybrid PCE-GPR method is extensively illustrated and validated based on both synthetic examples and high-dimensional application test cases in the field of electrical engineering, for which it is shown to substantially outperform state-of-the-art PCE methods such as least-angle regression, orthogonal matching pursuit, subspace pursuit, and Bayesian compressive sensing.

1. Introduction

Uncertainty quantification (UQ) and, specifically, uncertainty propagation,[1] is attracting an ever-growing interest in many field of science. This is particularly true in the design of modern mass-production electronics, since the increasing miniaturization is stressing the impact of process variations on the electrical performance [1]. Manufacturing tolerances are currently one of the main bottlenecks to further scaling towards sub-nm technologies [2]. Safe design margins are usually set via extensive Monte Carlo (MC) simulations [3], which significantly slows down the design phase. In this framework, polynomial chaos expansion (PCE) emerged as a primary alternative tool for UQ owing to its efficiency, optimal convergence rate, and model interpretability [4].

* Correspondence to: Department of Electronics and Telecommunications, Politecnico di Torino, Turin 10129, Italy.
E-mail address: paolo.manfredi@polito.it.

[1] The term "uncertainty quantification" is somewhat ambiguous in the scientific community. The meaning in the context of this work will be clarified later on.

https://doi.org/10.1016/j.cma.2024.117693
Received 18 September 2024; Received in revised form 17 December 2024; Accepted 18 December 2024
Available online 1 January 2025

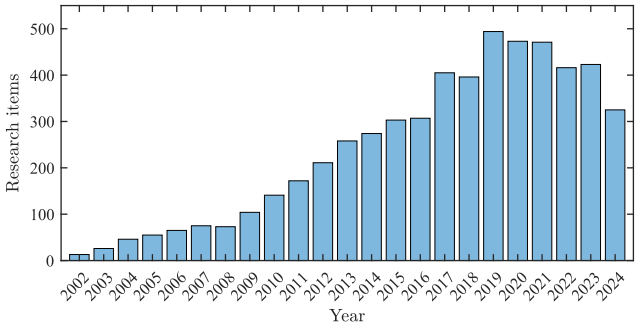

Fig. 1. Number of research items per year indexed by Scopus and containing the keyword "polynomial chaos".

While several alternative approaches have been proposed for UQ and sensitivity analysis (e.g., [5–7]), PCE offers several advantages, ranging from an analytical evaluation of first- and second-order moments and sensitivity indices [8] to an exponential convergence rate for finite-variance functions [9]. The key aspect in PCE-based simulations is the calculation of the expansion coefficients. A variety of methods have been proposed to accomplish this task [10,11]. Galerkin-based approaches [12] modify the stochastic governing equation to produce an augmented deterministic equivalent in the unknown PCE coefficients. They are regarded as the most accurate, yet they are intrusive and difficult to extend to nonlinear systems. Spectral or projection-based approaches approximate the exact integral calculation of the coefficients using suitable quadrature rules, usually relying on sparse grids [13]. As such, they only require to simulate the system at the quadrature nodes. However, the number of quadrature nodes increases superlinearly with the number of unknown coefficients, even for sparse schemes, thereby leading to an excessive amount of simulations in high-dimensional settings. Regression-based methods fit the PCE model using either standard least-square approaches or sparse schemes [14–16]. For instance, least-angle regression (LAR) [17] identifies a subset of important coefficients and the number of data samples can be smaller than the (total) number of unknown coefficients, which however may have a detrimental impact on the accuracy. Furthermore, an important limitation is that the uncertainty involved in the estimation of PCE coefficients is not quantified [18].

Since its modern conception in Xiu and Karniadakis's work of 2002, the PCE has been increasingly adopted in many domains of science including, but not limited to, electronics [19], fluid dynamics [20], and structural reliability [21]. Fig. 1 shows the number of research items per year containing the keyword "polynomial chaos" (source: Scopus). Despite the burgeoning of machine learning methods, polynomial chaos still represents a very active field of research, particularly targeting training efficiency and high dimensionality (e.g., [22–33]).

In general, most state-of-the-art PCE-based methods share one common limitation that reduces their efficacy for high-dimensional problems: the PCE is in essence a parametric method, whose model form is determined a priori by the expansion order and the input dimensionality. This leads to a number of regressors that increases exponentially with these two parameters, which in turn causes an increase in the amount of data required. This is a limitation for virtually any engineering system that is expensive to simulate. Moreover, the regressors must be built a priori, even for sparse methods, which may lead to memory issues for large expansions.

Along with PCE, kernel machine learning methods such as Gaussian process regression (GPR) and support-vector machines were recently investigated as alternative surrogates for expensive simulators in UQ and optimization tasks [34–40]. Since they are nonparametric, i.e., they build on data rather than on a predefined model form, they typically achieve a good accuracy with a limited amount of training data, even for high-dimensional problems. However, they typically lack any special properties in UQ scenarios, and as such they are usually employed as mere surrogates in a MC-based simulation.

Among the various kernel methods, GPR, also known as Kriging, leverages a Bayesian inference framework [41] and exhibits the attractive feature of embedding an estimate of the model uncertainty due to the limited availability of data [42]. The uncertainty information is typically used to assign confidence intervals to model predictions and/or to drive the acquisition of additional training samples in an active learning scheme [43–48]. GPR is usually more flexible and parsimonious in terms of training data [18]. However, propagation of uncertainty from inputs to outputs usually needs to be carried out numerically.

GPR has been applied to UQ, e.g., in [49–58]. In [49–51], semi-analytical estimates were provided for statistical moments and sensitivity, which require to compute several bulky integrals in the input probability space. Therefore, the calculations are only viable for low-dimensional problems. In [52], a treed multi-output GPR model was put forward that is able to identify discontinuities, local features and unimportant dimensions in the solution of stochastic differential equations. The approach was later extended to an efficient and fully Bayesian framework in which correlation between distinct outputs as well as space and/or time are explicitly modeled [53]. An approach that uses an infinite mixture of Gaussian processes was introduced in [54] to simultaneously address the difficulties arising from a limited availability of data, the scaling to higher dimensions, as well as multivariate and correlated responses that exhibit localized features and/or discontinuities. High-dimensional UQ was further addressed in [55] by means of a GPR formulation with built-in dimensionality reduction based on active subspaces and in [57] via the projection of training points onto a Grassmannian manifold. In [58], the convergence was improved by incorporating physical constraints. The case in which high dimensionality originates from a truncated Karhunen-Loève expansion was tackled in [59]. In [60], the prediction confidence

is propagated to UQ measures when the GPR model is used as a surrogate in a MC analysis, which allows extending the method to higher dimensions and assigning confidence bounds to estimated statistical moments and probability distributions. However, the uncertainty propagation requires to explicitly compute the covariance matrix of all MC samples, which limits the calculation efficiency if their number is large and poses a trade-off with the accuracy.

A new paradigm that combines the advantages of PCE and kernel machine learning methods was recently introduced in [61]. The approach reformulates the calculation of PCE coefficients as a least-square support-vector machine regression problem by leveraging special kernels built from an infinite sequence of Hermite and Legendre polynomials. In this paper, we utilize similar kernels to put forward a novel formulation that hybridizes the PCE and GPR methods. The proposed hybrid PCE-GPR technique first builds a GPR model using suitable kernels for Gaussian and uniform variables (possibly mixed). The GPR model is then converted to a PCE by the analytical integration of its posterior, which is made possible thanks to the special kernel definitions. This results in a compact set of equations, similar to posterior calculations in standard GPR settings and providing the expected value and covariance matrix of the estimated PCE coefficients. Confidence information on the PCE coefficients, which was not available in [61], is readily obtained and propagated to statistical moments and sensitivity indices (analytically) and to distribution functions (numerically). Therefore, the model borrows the training efficiency and confidence information from the GPR component and the statistical interpretability from the PCE one. While a similar idea was proposed in [62], the formulation relied on a covariance function explicitly constructed from the inner product of the orthogonal basis functions. The limitations in terms of accuracy and scalability of this approach in the kernel construction were already discussed and illustrated in [61].

A multi-output model is also introduced, which readily generalizes the formulation to multiple quantities of interest and involves the optimization of a single set of hyperparameters for all output components. While providing a more efficient training, this approach is shown to provide comparable results against alternative compression techniques based on principal component analysis (PCA) or similar strategies like proper orthogonal decomposition [63–66], sufficient dimension reduction [67], or polynomial dimensional decomposition [68].

The advocated technique is first illustrated based on two simple synthetic functions. It is then applied to real-life application test cases in the field of electrical engineering, involving high-dimensional and/or multi-output simulations in time or frequency domain. The examples demonstrates that the PCE-GPR method achieves accurate results with a number of training samples that is lower than the state-of-the-art LAR, orthogonal matching pursuit (OMP), subspace pursuit (SP), and Bayesian compressive sensing (BCS), as well as comparable performance against standard GPR, while also providing accurate and closed-form statistical and confidence information. The impact of the hyperparameters is also investigated.

The remainder of the paper is organized as follows. Section 2 briefly states the problem. Sections 3 and 4 provide the necessary background of PCE and GPR methods, respectively. The proposed hybrid method is outlined in Section 5 and extended to multiple outputs in Section 6. Section 7 illustrates the method based on two analytical examples, whereas Section 8 discusses real-life application test cases. The accuracy and training efficiency are further investigated in Section 9. Section 10 discuss an additional test case with 54 uncertain parameters. Finally, conclusions are drawn in Section 11.

2. Problem statement

Let us introduce a target quantity of interest y that depends on a set of d uncertain design parameters $\boldsymbol{x} = (x_1, \dots, x_d) \in \mathcal{X} \subseteq \mathbb{R}^d$ with known probability distribution $\omega : \mathcal{X} \to \mathbb{R}^+$. We express the generic input–output relationship as

$$y = \mathcal{M}(\boldsymbol{x}), \tag{1}$$

where $\mathcal{M} : \mathcal{X} \to \mathbb{R}$ denotes the computational model or code that is run in order to obtain y for a given configuration of the parameters $\boldsymbol{x}$.

We aim at performing UQ of y by leveraging a surrogate representation of $\mathcal{M}$. In engineering, UQ is usually considered as the process of estimating statistics and distribution of the output y given those on the inputs $\boldsymbol{x}$. This is sometimes more properly referred to as (forward) uncertainty propagation and is the primary focus in this work. However, we also address the intrinsic uncertainty introduced by the approximation of $\mathcal{M}$ with the surrogate, which is part of the more general scope of UQ.

We first assume the output of $\mathcal{M}$ to be scalar and real-valued. In Section 6, we discuss an extension to multiple and possibly complex-valued outputs. We further assume the uncertain input parameters to be independent. This is a typical assumption in PCE-based methods. Extensions to correlated inputs are available, and they are trivial in the Gaussian case [10]. In the next sections, we introduce the baseline PCE and GPR frameworks as needed to develop the advocated hybrid method.

3. PCE

The PCE approximates the functional relationship (1) using an expansion of special orthogonal polynomials based on the distribution of the input parameters $\boldsymbol{x}$ [9], i.e.,

$$y \approx \hat{y} = \mathcal{M}_{\mathrm{PCE}}(\boldsymbol{x}) = \sum_{\boldsymbol{\kappa} \in \mathcal{K}} c_{\boldsymbol{\kappa}} \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}). \tag{2}$$

The multivariate orthogonal polynomials $\Psi_{\boldsymbol{\kappa}}$ are built as the product combination of standard univariate polynomials [10], i.e.,

$$\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) = \prod_{i=1}^{d} \psi_{\kappa_i}^{(i)}(x_i). \tag{3}$$

The univariate polynomials $\psi_{\kappa_i}^{(i)}$ are orthogonal based on the inner product

$$\langle f, g \rangle = \int_{\mathcal{X}_i} f(x_i) g(x_i) \omega_i(x_i) dx_i, \tag{4}$$

where $\omega_i(x_i)$ is the probability density function (PDF) of x_i, whereas the multi-indices $\boldsymbol{\kappa} = (\kappa_1, \dots, \kappa_d) \in \mathcal{K} \subseteq \mathbb{N}^d$ describe the degree of the polynomial in each input dimension, since $\deg\left(\psi_k^{(i)}\right) = k$. For example, in a three-dimensional expansion, $\boldsymbol{\kappa} = (1, 0, 2)$ indicates the product of a linear, constant, and quadratic polynomial in the first, second, and third variable, respectively.

The orthogonality of multivariate polynomials is readily inherited from the univariate ones thanks to their product definition (3), which make $\Psi_{\boldsymbol{\kappa}}$ satisfy the orthogonality condition

$$\langle \Psi_{\boldsymbol{\kappa}}, \Psi_{\boldsymbol{\upsilon}} \rangle = \int_{\mathcal{X}} \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) \Psi_{\boldsymbol{\upsilon}}(\boldsymbol{x}) \omega(\boldsymbol{x}) d\boldsymbol{x} = \delta_{\boldsymbol{\kappa}\boldsymbol{\upsilon}}, \tag{5}$$

where $\delta_{\boldsymbol{\kappa}\boldsymbol{\upsilon}}$ denotes the Kronecker's delta and $\omega(\boldsymbol{x}) = \prod_{i=1}^{d} \omega_i(x_i)$ is the joint probability density of the input variables $\boldsymbol{x}$.

In the following, we will restrict the discussion to Hermite and Legendre polynomials, which are orthogonal based on the Gaussian and uniform distribution, respectively [9]. These are the two most common distributions that cover the vast majority of PCE applications. We denote the univariate (probabilists') Hermite polynomials as $H_k(x)$ and the Legendre polynomials as $P_k(x)$. We further assume the polynomials to be orthonormal, so that $\left\|\Psi_{\boldsymbol{\kappa}}\right\| = 1$, $\forall \boldsymbol{\kappa}$. This is readily achieved by a proper renormalization of the univariate polynomials, i.e., $\tilde{H}_k(x) = H_k(x)/\sqrt{k!}$ for the Hermite polynomials and $\tilde{P}_k(x) = \sqrt{2k+1} P_k(x)$ for the Legendre polynomials.

3.1. Truncation

If $\mathcal{K} \equiv \mathbb{N}^d$, the PCE representation (2) is exact for the functions with finite variance, since the orthogonal polynomials form a complete basis on L^2. However, for the practical implementation, the PCE is typically truncated by defining a multi-index set $\mathcal{K}$ in which a suitable norm of the multi-indices is bounded by a maximum expansion order p, i.e.,

$$\mathcal{K} = \{\boldsymbol{\kappa} : \|\boldsymbol{\kappa}\|_u \leq p\}. \tag{6}$$

The most popular strategy uses $u = 1$ and retains all polynomials whose total degree is at most p, leading to $|\mathcal{K}| = \binom{p+d}{d} = (p+d)!/(p!d!)$ expansion terms. Alternative strategies use $u = \infty$, retaining all terms up to order p in each variable (tensor product truncation), or $u < 1$, which neglects higher-order interactions thereby leading to a sparser expansion (hyperbolic truncation).

3.2. Statistical information

A peculiar property of the PCE is that statistical moments and sensitivity indices are analytically derived from the model coefficients, thanks to the orthogonality of the basis functions. Indeed, the expectation and the variance of y are readily found as

$$\mu_y \approx \mathrm{E}(\hat{y}) = c_{\mathbf{0}} \tag{7}$$

(i.e., the coefficient of the constant term) and

$$\sigma_y^2 \approx \mathrm{Var}(\hat{y}) = \sum_{\boldsymbol{\kappa} \in \mathcal{K} \setminus \mathbf{0}} c_{\boldsymbol{\kappa}}^2, \tag{8}$$

respectively [10]. The total Sobol' sensitivity indices, describing the individual impact of each random input on the output variance, are instead obtained as [69]

$$S_{Ti} \approx \frac{\sum_{\boldsymbol{\kappa} \in \mathcal{K}_i} c_{\boldsymbol{\kappa}}^2}{\mathrm{Var}(\hat{y})} = \frac{\sum_{\boldsymbol{\kappa} \in \mathcal{K}_i} c_{\boldsymbol{\kappa}}^2}{\sum_{\boldsymbol{\kappa} \in \mathcal{K} \setminus \mathbf{0}} c_{\boldsymbol{\kappa}}^2}, \tag{9}$$

where $\mathcal{K}_i = \{\boldsymbol{\kappa} \in \mathcal{K} : \kappa_i \neq 0\} \subset \mathcal{K}$ denotes the subset of multi-indices with non-zero ith component. Hence, the total index S_{Ti} for the ith variable is proportional to the sum of the squared coefficients of the basis functions that are non-constant in the ith variable.

3.3. Calculation of the coefficients

Since (2) is an orthogonal expansion, the rigorous calculation of its coefficients is by projection onto the basis functions [10], i.e.,

$$c_{\boldsymbol{\kappa}} = \langle \mathcal{M}, \Psi_{\boldsymbol{\kappa}} \rangle = \int_{\mathcal{X}} \mathcal{M}(\boldsymbol{x}) \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) \omega(\boldsymbol{x}) d\boldsymbol{x}. \tag{10}$$

In practice, the above calculation is often unfeasible because the dimensionality is high and/or (1) is expensive to evaluate, which is the case of interest here. Therefore, several alternative strategies were devised in the literature.

The stochastic Galerkin method (SGM) is an intrusive approach by which stochastic system variables are expressed by their respective PCEs (2) and the projections (10) are performed directly on the governing equations [12]. The resulting integrals only involve the basis functions and are resolved analytically. The result is a $|\mathcal{K}|$-times augmented and coupled deterministic system in the unknown PCE coefficients. This numerical approach is considered to be the most accurate [10], yet it is only applicable to relatively

simple systems for which the governing equations are accessible (and, typically, linear). The SGM has been extensively applied to physics and engineering problems (e.g., [70–76].) In the context of circuit simulations, a modular element-wise approach was proposed [19], in which an augmented equivalent companion was created for stochastic elements and used to assemble the overall SGM circuit, whose simulation provided the sought-for expansion coefficients. This technique will serve as one of the reference methods for the application examples.

Non-intrusive approaches are more flexible as they only require to know some input–output data, thereby applying to black-box and/or nonlinear systems as well. In quadrature-based pseudo-spectral methods, the projection integral (10) is approximated by means of a suitable quadrature rule, e.g., a full tensor Gauss quadrature or a Smolyak' sparse grid quadrature [13]. In both cases, (10) is computed as a weighted sum of the model observations evaluated at the Q nodes specified by the quadrature rule, i.e.,

$$c_{\boldsymbol{\kappa}} \approx \sum_{q=1}^{Q} \mathcal{M}(\boldsymbol{x}_q)\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}_q)w_q \tag{11}$$

Typically, however, $Q \gg |\mathcal{K}|$, even for sparse schemes. Hence, the number of simulations is larger than the augmentation factor in the Galerkin case, although each problem can be solved independently.

While collocation methods require to carry out simulations at predefined points, regression-based approaches are purely data-driven and fit the coefficients by minimizing the quadratic model error based on a given dataset of observations $D = \{(\boldsymbol{x}_l, y_l)\}_{l=1}^{L}$, with $y_l = \mathcal{M}(\boldsymbol{x}_l)$ for $l = 1, \dots, L$, using the well-known least-square solution

$$\boldsymbol{c} = \underset{\boldsymbol{c}\in\mathbb{R}^{|\mathcal{K}|}}{\arg\min} \left(\frac{1}{L}\sum_{l=1}^{L}\left(\sum_{\boldsymbol{\kappa}\in\mathcal{K}} c_{\boldsymbol{\kappa}}\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}_l) - y_l \right)^2 \right) = \left(\boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{y}, \tag{12}$$

where $\boldsymbol{c} = (\dots, c_{\boldsymbol{\kappa}}, \dots)^{\mathsf{T}}$ is the vector of PCE coefficients, $\boldsymbol{\Psi}$ is the matrix of the basis functions evaluated at the data samples, with entries $\Psi_{\boldsymbol{\kappa}l} = \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}_l)$, for $\boldsymbol{\kappa} \in \mathcal{K}$ and $l = 1, \dots, L$, whereas $\boldsymbol{y} = (y_1, \dots, y_L)^{\mathsf{T}}$ is the vector of the observations. As opposed to quadrature-based approaches, the sampling points are not predetermined. They are usually drawn randomly according to the input distribution, so that the "loss function" that is minimized in (12) becomes a MC approximation of the L^2 error and the solution approaches the rigorous computation by projection as the number of data samples is increased.

For ordinary least-square schemes, the number of samples must be $L > |\mathcal{K}|$ for the regression system to be overdetermined (typically, $L \geq 2|\mathcal{K}|$ is recommended [4]). Sparse approaches, such as LAR [17], introduce a penalty related to the 1-norm of the coefficients to enforce low-rank solutions and identify a subset $\mathcal{A} \subseteq \mathcal{K}$ of most relevant coefficients, i.e., they solve

$$\boldsymbol{c} = \underset{\boldsymbol{c}\in\mathbb{R}^{|\mathcal{A}|}}{\arg\min} \left(\frac{1}{L}\sum_{l=1}^{L}\left(\sum_{\boldsymbol{\kappa}\in\mathcal{A}} c_{\boldsymbol{\kappa}}\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}_l) - y_l \right)^2 + \lambda \left\|\boldsymbol{c}\right\|_1 \right), \tag{13}$$

where $\boldsymbol{c}$ is now a subset of coefficients (the others are assumed to be zero). Theoretically, there is no lower limit on the number of observations L, but the number of identifiable coefficients is upper bounded by L, i.e., $|\mathcal{A}| \leq L$. Hence, as we will show in the numerical examples, using a low amount of data samples may lead LAR to miss relevant coefficients, thereby causing a detrimental loss of accuracy.

Alternative approaches include OMP [77], SP [78], and BCS [79,80]. OMP aims at minimizing the approximation residual using a greedy iterative strategy, in which the active set of regressors is updated by including the polynomial basis that is most correlated to the current approximation residual. SP is sparse regression algorithm that identifies a subset of non-zero elements in the coefficient vector by means of iterative ordinary least-square solutions. BCS recast the regression in a Bayesian framework in which likelihood and priors are selected to promote sparsity in the regression coefficients, which are calculated via a maximum-a-posteriori estimate. We refer to [16] and [81] for additional details on these methods.

3.4. Limitations

As discussed in the previous section, the number of PCE terms $|\mathcal{K}|$ increases with the expansion order p and/or the problem dimensionality d. This in turn determines the size of the problem to be solved for the calculation of the coefficients, thereby leading to the so-called "curse of dimensionality". Even with sparse methods like LAR, the complete set of basis functions must still be constructed based on a predefined truncation scheme, which may lead to memory issues and – as we will show – has an impact on the prediction accuracy. Furthermore, the uncertainty in the estimation is usually not quantified, unless a Bayesian approach is used [6,28,41,82,83]. In this regard, machine learning methods are more advantageous as they typically benefit from a model-free and data-driven nature. These include GPR, which inherently provides a Bayesian framework and is outlined next.

4. GPR

In this section, we introduce the second key ingredient for the proposed hybrid method. GPR is a kernel method that assumes the target function to be a specific realization of a prior Gaussian process [42], i.e.,

$$y = \mathcal{M}(\boldsymbol{x}) \sim \mathcal{GP}\left(\mu(\boldsymbol{x}), \sigma^2 r(\boldsymbol{x}, \boldsymbol{x}')\right). \tag{14}$$

The kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 r(\boldsymbol{x}, \boldsymbol{x}')$ plays the role of prior covariance function, whereas $\mu(\boldsymbol{x})$ is the prior mean function or *trend*. We outline here the general regression framework, in which each observation is possibly affected by a Gaussian homoschedastic noise [84],

i.e., $y_l = \mathcal{M}(\boldsymbol{x}_l) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. Assuming noise on the training data helps avoid overfitting when the number of samples is large, whereas setting $\sigma_n = 0$ provides an exact interpolation of the data. Moreover, for our purposes, we assume $\mu(\boldsymbol{x}) = 0$, which is a rather common assumption also in general settings [42].

The posterior model is obtained by means of Bayesian inference, conditioning the prior to the available observations. This leads to a stochastic posterior function

$$\mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), \varsigma^2(\boldsymbol{x}, \boldsymbol{x}')\right) \tag{15}$$

that also follows a Gaussian process distribution. The model predictions are provided by the posterior mean, which reads [84]

$$y \approx \hat{y} = m(\boldsymbol{x}) = \mathrm{E}\left(\mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x})\right) = \sum_{l,m=1}^{L} y_m \left[\tilde{\boldsymbol{R}}^{-1}\right]_{lm} \tilde{r}(\boldsymbol{x}, \boldsymbol{x}_l), \tag{16}$$

where

$$\tilde{r}(\boldsymbol{x}, \boldsymbol{x}_l) = (1-\tau) r(\boldsymbol{x}, \boldsymbol{x}'), \tag{17}$$

$$\tilde{\boldsymbol{R}} = (1-\tau)\boldsymbol{R} + \tau \boldsymbol{I}, \tag{18}$$

$$\tau = \frac{\sigma_n^2}{\sigma^2 + \sigma_n^2} \tag{19}$$

is the "noise ratio", and $\boldsymbol{R}$ is the correlation matrix of the training samples, with entries $R_{lm} = r(\boldsymbol{x}_l, \boldsymbol{x}_m)$, $l, m = 1, \dots, L$. Notably, the posterior covariance

$$\varsigma^2(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{cov}\left(\mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x}), \mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x}')\right) = \sigma_{\mathrm{tot}}^2 c(\boldsymbol{x}, \boldsymbol{x}'), \tag{20}$$

where $\sigma_{\mathrm{tot}}^2 = \sigma^2 + \sigma_n^2$ and

$$c(\boldsymbol{x}, \boldsymbol{x}') = \tilde{r}(\boldsymbol{x}, \boldsymbol{x}') - \sum_{l,m=1}^{L} \tilde{r}(\boldsymbol{x}, \boldsymbol{x}_l) \left[\tilde{\boldsymbol{R}}^{-1}\right]_{lm} \tilde{r}(\boldsymbol{x}_m, \boldsymbol{x}'), \tag{21}$$

is an expression of the model uncertainty due to the limited amount of data used for the inference. In particular, the probabilistic prediction at a given point $\boldsymbol{x}^*$ is Gaussian random variable with expected value $m(\boldsymbol{x}^*)$, given by (16), and standard deviation $\varsigma(\boldsymbol{x}^*, \boldsymbol{x}^*)$, computed from (20).

One fundamental difference that renders the GPR model (16) more attractive than the PCE model (2) is that the complexity is dictated by the amount of available training data L rather than by the size $|\mathcal{K}|$ of the polynomial basis, potentially with $L \ll |\mathcal{K}|$. This is one aspect of the so-called "kernel trick" in machine learning methods.

The prior correlation function $r(\cdot, \cdot)$ often depends on a set of independent hyperparameters $\boldsymbol{\rho}$. Popular choices for generic applications are the squared-exponential and the family of Matérn correlation functions [42,84]. The hyperparameters and the noise ratio $\tau \in [0, 1)$ are usually estimated using either maximum likelihood (ML) or cross-validation (CV). In the latter case, several models are trained using mutually exclusive subsets of the training samples $\{\mathcal{D}_k\}_{k=1}^K \subset \mathcal{D}$, with $\mathcal{D}_k \cap \mathcal{D}_j = \emptyset$, $\forall k, j = 1, \dots, K$. The hyperparameters and the noise ratio are then obtained by minimizing the cumulative CV error between the subsets, defined as

$$\epsilon_{\mathrm{CV}} = \sum_{k=1}^{K} \sum_{(\boldsymbol{x}_l, y_l) \in \mathcal{D}_k} \left(y_l - m_{-k}(\boldsymbol{x}_l)\right)^2, \tag{22}$$

where m_{-k} denotes the posterior trend (prediction) of the model trained without the samples in subset $\mathcal{D}_k$. The total variance is estimated as [84]

$$\sigma_{\mathrm{tot}}^2 = \frac{1}{L} \sum_{k=1}^{K} \sum_{(\boldsymbol{x}_l, y_l) \in \mathcal{D}_k} \frac{\left(y_l - m_{-k}(\boldsymbol{x}_l)\right)^2}{c_{-k}^2(\boldsymbol{x}_l, \boldsymbol{x}_l)}, \tag{23}$$

where c_{-k} denotes the posterior correlation (21) of the model trained without the samples in subset $\mathcal{D}_k$. The kernel variance σ^2 and the noise variance σ_n^2 are readily derived from (19) as $\sigma_n^2 = \tau \sigma_{\mathrm{tot}}^2$ and $\sigma^2 = (1-\tau)\sigma_{\mathrm{tot}}^2$, respectively. Alternatively, the total variance can be obtained using the ML estimate [84]

$$\sigma_{\mathrm{tot}}^2 = \frac{1}{L} \boldsymbol{y}^{\mathsf{T}} \tilde{\boldsymbol{R}}^{-1} \boldsymbol{y}. \tag{24}$$

Special cases are hold-out CV, which considers a single subset with only a fraction of the available training samples, and leave-one-out CV (LOO-CV), which operates over $K = L$ subsets (folds), each with one sample out. This is perhaps the most popular and robust method, whose efficiency however reduces for larger datasets. For LOO-CV, (22) and (23) simplify to

$$\epsilon_{\mathrm{CV}} = \sum_{k=1}^{L} \sum_{l \neq k} \left(y_l - m_{-k}(\boldsymbol{x}_l)\right)^2, \tag{25}$$

and

$$\sigma_{\mathrm{tot}}^2 = \frac{1}{L} \sum_{k=1}^{L} \sum_{l \neq k} \frac{\left(y_l - m_{-k}(\boldsymbol{x}_l)\right)^2}{c_{-k}^2(\boldsymbol{x}_l, \boldsymbol{x}_l)}, \tag{26}$$

respectively, where subset $\mathcal{D}_k$ is understood to contain only the kth training sample.

Standard GPR has been applied to forward UQ in several works. In [49], statistical information was extracted via numerical integration of the posterior over the input probability density. Albeit rigorous, this approach is only feasible for a moderate number of uncertain parameters. In [60], GPR was applied to surrogate the actual system (1) in a MC analysis. Closed-form results were derived for the expected value, the variance, and the distribution of the MC predictions of the first two statistical moments, i.e., the mean and the variance of the output. Confidence bounds for the PDF of the output were obtained numerically based on posterior realizations. However, the accuracy and the efficiency of the calculations was limited by the amount of MC samples considered. Specifically, considering a large number of MC samples ensures a good accuracy but also leads to a prohibitive size of the covariance matrix to be handled. The situation worsens for the MC-based calculation of sensitivity indices, which requires an even larger amount of samples [85].

5. Hybrid PCE-GPR method

In this section, we outline the proposed hybrid PCE-GPR method. The approach consists of three main steps: (1) the training of a GPR model with specifically designed kernels; (2) the post-processing and closed-form calculation of PCE coefficients; (3) the extraction of statistical information from the PCE coefficients.

5.1. GPR settings

The hybrid method features a GPR formulation with a zero trend $\mu(\boldsymbol{x}) = 0$ and special correlation functions that are constructed from the orthogonal polynomials of the Wiener-Askey scheme. The special kernels for Hermite and Legendre polynomials were originally introduced in the framework of least-square support-vector machine regression [61]. While this paper focuses on Gaussian and uniform variability, it is very likely that similar kernels can be found also for other orthogonal polynomials.

The univariate Hermite kernel is the probabilists' version of Mehler kernel [86], which reads [87]

$$r(x, x'; \rho) = \sum_{k=0}^{\infty} \frac{\rho^k}{k!} H_k(x) H_k(x') = \sum_{k=0}^{\infty} \rho^k \tilde{H}_k(x) \tilde{H}_k(x') = \frac{1}{\sqrt{1-\rho^2}} \exp\left(\frac{-\rho^2 \left(x^2 + x'^2\right) - 2\rho x x'}{2(1-\rho^2)} \right), \tag{27}$$

where $\rho \in (0, 1)$ is a hyperparameter. Similarly, the univariate Legendre kernel reads [88]

$$r(x, x'; \rho) = \sum_{k=0}^{\infty} \rho^k P_k(x) P_k(x') = \sum_{k=0}^{\infty} \frac{\rho^k}{2k+1} \tilde{P}_k(x) \tilde{P}_k(x') = \frac{K(u)}{\pi \sqrt{a-b}}, \tag{28}$$

where

$$u = \sqrt{\frac{2b}{b-1}}, \tag{29}$$

$$a = 1 - 2xx'\rho + \rho^2, \tag{30}$$

$$b = -2\rho \sqrt{1-x^2} \sqrt{1-x'^2}, \tag{31}$$

and $K(u)$ is the complete elliptic integral of the first kind, i.e.,

$$K(u) = \int_0^1 \frac{dt}{\sqrt{(1-t^2)(1-u^2t^2)}}. \tag{32}$$

Notably, both kernels are constructed from an infinite sequence of orthogonal polynomials, i.e., from the complete L^2 basis.

The multivariate kernel is built as a separable function [89], leading to

$$r(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\rho}) = \prod_{i=1}^{d} r_i(x_i, x_i'; \rho_i) = \prod_{i=1}^{d} \left(\sum_{k=0}^{\infty} \lambda_k^{(i)} \psi_k^{(i)}(x_i) \psi_k^{(i)}(x_i') \right), \tag{33}$$

where r_i denote the kernel for the ith variable (Hermite or Legendre, depending on the distribution of x_i) and $\psi_k^{(i)}$ are the corresponding orthonormal basis polynomials ($\tilde{H}_k$ or $\tilde{P}_k$). The coefficients $\lambda_k^{(i)}$ are

$$\lambda_k^{(i)} = \rho_i^k \tag{34}$$

for the Hermite kernel and

$$\lambda_k^{(i)} = \frac{\rho_i^k}{2k+1} \tag{35}$$

for the Legendre kernel. As in classical GPR, the kernel is said to be isotropic if the same hyperparameter ρ is used for each input dimension. It should be noted that, in its original formulation [61], the kernel variance was simply assumed to be $\sigma^2 = 1$. However, while this has little effect on the prediction, it does affect the posterior covariance (20) and is therefore fundamental to correctly gauge the prediction uncertainty.

The comparison with (3) allows observing that the multivariate kernel (33) contains all possible product combinations of the polynomials, up to order infinity. Therefore, it can be expressed in terms of the multivariate basis (3) as

$$r(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\rho}) = \sum_{\boldsymbol{\kappa} \in \mathbb{N}^d} \lambda_{\boldsymbol{\kappa}} \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}'), \tag{36}$$

where

$$\lambda_{\boldsymbol{\kappa}} = \prod_{i=1}^{d} \lambda^{(i)}_{\kappa_i}. \tag{37}$$

It should be noted that L^2 functions implicitly satisfy the fundamental GPR assumption, i.e., they are well represented by a linear combination of the above-defined kernel functions.

With the above definitions, the GPR model is readily trained as outlined in Section 4. We recast the model prediction (16) as

$$y \approx \hat{y} = m(\boldsymbol{x}) = \sum_{l=1}^{L} \alpha_l r(\boldsymbol{x}, \boldsymbol{x}_l), \tag{38}$$

where the kernel coefficients

$$\alpha_l = (1-\tau) \sum_{m=1}^{L} \left[\tilde{\boldsymbol{R}}^{-1}\right]_{lm} y_m \tag{39}$$

have been introduced.

Overall, the model prediction is a linear combination of kernel functions centered at the training samples. At this stage, predictions can be generated with (38) and the model be used to surrogate a MC analysis. However, in the next section, we show that the model can also be inexpensively converted into a PCE of arbitrary order by means of a simple post-processing step. This readily allows obtaining PCE coefficients as well as related statistical information with the inclusion of confidence bounds.

5.2. Post-Processing Calculation of the PCE coefficients

We now want to derive the PCE coefficients probabilistically, by fully exploiting the Bayesian definition of the GPR model. According to (10), the GPR-based PCE coefficient with multi-index $\boldsymbol{\kappa}$ is obtained by projecting the model onto the corresponding basis function $\Psi_{\boldsymbol{\kappa}}$, i.e.,

$$c_{\boldsymbol{\kappa}} = \int_{\mathcal{X}} \mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x}) \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) \omega(\boldsymbol{x}) d\boldsymbol{x}. \tag{40}$$

Since $\mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x})$ is a stochastic function, we are interested in computing the expectation of (40) as well as the covariance between the various PCE coefficients. The calculation is carried out analytically thanks to the special structure of the kernel function introduced in Section 5.1.

5.2.1. Expected value of the PCE coefficients

The expected value of (40) is readily computed as

$$\begin{aligned}
\mathrm{E}\left(c_{\boldsymbol{\kappa}}\right) &= \mathrm{E}\left(\int_{\mathcal{X}} \mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x}) \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) \omega(\boldsymbol{x}) d\boldsymbol{x}\right) = \int_{\mathcal{X}} \underbrace{\mathrm{E}\left(\mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x})\right)}_{=m(\boldsymbol{x})} \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) \omega(\boldsymbol{x}) d\boldsymbol{x} \\
&= \sum_{l=1}^{L} \alpha_l \int_{\mathcal{X}} r(\boldsymbol{x}, \boldsymbol{x}_l) \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) \omega(\boldsymbol{x}) d\boldsymbol{x} = \sum_{l=1}^{L} \alpha_l \sum_{\boldsymbol{\upsilon} \in \mathbb{N}^d} \lambda_{\boldsymbol{\upsilon}} \Psi_{\boldsymbol{\upsilon}}(\boldsymbol{x}_l) \underbrace{\int_{\mathcal{X}} \Psi_{\boldsymbol{\upsilon}}(\boldsymbol{x}) \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}) \omega(\boldsymbol{x}) d\boldsymbol{x}}_{=\delta_{\boldsymbol{\upsilon}\boldsymbol{\kappa}}} \\
&= \lambda_{\boldsymbol{\kappa}} \sum_{l=1}^{L} \alpha_l \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}_l)
\end{aligned} \tag{41}$$

where (5) and (38) have been used. An important feature to point out is that (41) allows retrieving any PCE coefficient with an arbitrary multi-index $\boldsymbol{\kappa}$ from the GPR coefficients α_l, regardless of its polynomial degree, by merely evaluating the corresponding basis function at the training samples. This is possible because the PCE-GPR kernel (36) embeds all basis functions of any order, which therefore does not need to be preset.

While the PCE coefficients can be retrieved individually, the calculation is more conveniently performed for a set of basis functions $\{\Psi_{\boldsymbol{\kappa}}\}_{\boldsymbol{\kappa} \in \mathcal{K}}$ and expressed in matrix form as

$$\boldsymbol{\mu}_{\boldsymbol{c}} = \mathrm{E}(\boldsymbol{c}) = \boldsymbol{\Lambda} \boldsymbol{\Psi} \boldsymbol{\alpha}, \tag{42}$$

where

$$\boldsymbol{\alpha} = (1-\tau) \tilde{\boldsymbol{R}}^{-1} \boldsymbol{y} \tag{43}$$

is the vector of kernel coefficients, $\boldsymbol{\Psi}$ is again the matrix of the basis functions evaluated at the training samples as defined in Section 3, whereas $\boldsymbol{\Lambda}$ is a diagonal matrix with entries $\Lambda_{\boldsymbol{\kappa}\boldsymbol{\kappa}} = \lambda_{\boldsymbol{\kappa}}$. The above equation provides a very simple formula to retrieve the most likely prediction of the PCE coefficients (in Bayesian terms) in post-processing.

5.2.2. Covariance between PCE coefficients

The GPR model not only provides the most likely prediction, but also the estimate of the prediction uncertainty, which is given by (20). This is reflected into the covariance between two arbitrary PCE coefficients $c_{\boldsymbol{\kappa}}$ and $c_{\boldsymbol{\nu}}$. In order to derive it, let us first recall that the following relation holds for the covariance between two random variables X and Y:

$$\operatorname{cov}(X,Y) = \mathrm{E}(XY) - \mathrm{E}(X)\,\mathrm{E}(Y)\,. \tag{44}$$

Therefore,

$$\begin{aligned}\operatorname{cov}\left(c_{\boldsymbol{\kappa}}, c_{\boldsymbol{\nu}}\right) &= \mathrm{E}\left(\left(\int_{\mathcal{X}} \mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x})\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})\omega(\boldsymbol{x})d\boldsymbol{x}\right)\left(\int_{\mathcal{X}} \mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x}')\Psi_{\boldsymbol{\nu}}(\boldsymbol{x}')\omega(\boldsymbol{x}')d\boldsymbol{x}'\right)\right) - \mathrm{E}\left(c_{\boldsymbol{\kappa}}\right)\mathrm{E}\left(c_{\boldsymbol{\nu}}\right) \\ &= \iint_{\mathcal{X}\times\mathcal{X}} \underbrace{\mathrm{E}\left(\mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x})\mathcal{M}_{\mathrm{GPR}}(\boldsymbol{x}')\right)}_{=\varsigma^2(\boldsymbol{x},\boldsymbol{x}')+m(\boldsymbol{x})m(\boldsymbol{x}')} \Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})\Psi_{\boldsymbol{\nu}}(\boldsymbol{x}')\omega(\boldsymbol{x})\omega(\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}' - \mathrm{E}\left(c_{\boldsymbol{\kappa}}\right)\mathrm{E}\left(c_{\boldsymbol{\nu}}\right),\end{aligned} \tag{45}$$

where we further combined (44) with (20). Since

$$\iint_{\mathcal{X}\times\mathcal{X}} m(\boldsymbol{x})m(\boldsymbol{x}')\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})\Psi_{\boldsymbol{\nu}}(\boldsymbol{x}')\omega(\boldsymbol{x})\omega(\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}' = \mathrm{E}\left(c_{\boldsymbol{\kappa}}\right)\mathrm{E}\left(c_{\boldsymbol{\nu}}\right), \tag{46}$$

the term cancels out in (45) and the calculation of the covariance reduces to

$$\operatorname{cov}\left(c_{\boldsymbol{\kappa}}, c_{\boldsymbol{\nu}}\right) = \iint_{\mathcal{X}\times\mathcal{X}} \varsigma^2(\boldsymbol{x},\boldsymbol{x}')\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})\Psi_{\boldsymbol{\nu}}(\boldsymbol{x}')\omega(\boldsymbol{x})\omega(\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}' \tag{47}$$

By combining (21) and (36), we obtain

$$\begin{aligned}\operatorname{cov}\left(c_{\boldsymbol{\kappa}}, c_{\boldsymbol{\nu}}\right) =& \sigma_{\mathrm{tot}}^2(1-\tau)\iint_{\mathcal{X}\times\mathcal{X}} r(\boldsymbol{x},\boldsymbol{x}')\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})\Psi_{\boldsymbol{\nu}}(\boldsymbol{x}')\omega(\boldsymbol{x})\omega(\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}' \\ &- \sigma_{\mathrm{tot}}^2(1-\tau)^2\sum_{l,m=1}^{L}\left[\tilde{\boldsymbol{R}}^{-1}\right]_{lm}\iint_{\mathcal{X}\times\mathcal{X}} r(\boldsymbol{x},\boldsymbol{x}_l)r(\boldsymbol{x}_m,\boldsymbol{x}')\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})\Psi_{\boldsymbol{\nu}}(\boldsymbol{x}')\omega(\boldsymbol{x})\omega(\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}' \\ =& \sigma_{\mathrm{tot}}^2(1-\tau)\sum_{\boldsymbol{\upsilon}\in\mathbb{N}^d}\lambda_{\boldsymbol{\upsilon}}\underbrace{\iint_{\mathcal{X}\times\mathcal{X}}\Psi_{\boldsymbol{\upsilon}}(\boldsymbol{x})\Psi_{\boldsymbol{\upsilon}}(\boldsymbol{x}')\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})\Psi_{\boldsymbol{\nu}}(\boldsymbol{x}')\omega(\boldsymbol{x})\omega(\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}'}_{\delta_{\boldsymbol{\kappa}\boldsymbol{\upsilon}}\delta_{\boldsymbol{\nu}\boldsymbol{\upsilon}}} \\ &- \sigma_{\mathrm{tot}}^2(1-\tau)^2\sum_{l,m=1}^{L}\left[\tilde{\boldsymbol{R}}^{-1}\right]_{lm}\sum_{\boldsymbol{\upsilon},\boldsymbol{\eta}\in\mathbb{N}^d}\lambda_{\boldsymbol{\upsilon}}\lambda_{\boldsymbol{\eta}}\Psi_{\boldsymbol{\upsilon}}(\boldsymbol{x}_l)\Psi_{\boldsymbol{\eta}}(\boldsymbol{x}_m)\underbrace{\iint_{\mathcal{X}\times\mathcal{X}}\Psi_{\boldsymbol{\upsilon}}(\boldsymbol{x})\Psi_{\boldsymbol{\eta}}(\boldsymbol{x}')\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x})\Psi_{\boldsymbol{\nu}}(\boldsymbol{x}')\omega(\boldsymbol{x})\omega(\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}'}_{\delta_{\boldsymbol{\kappa}\boldsymbol{\upsilon}}\delta_{\boldsymbol{\nu}\boldsymbol{\eta}}} \\ =& \sigma_{\mathrm{tot}}^2(1-\tau)\left(\lambda_{\boldsymbol{\kappa}}\delta_{\boldsymbol{\kappa}\boldsymbol{\nu}} - (1-\tau)\sum_{l,m=1}^{L}\left[\tilde{\boldsymbol{R}}^{-1}\right]_{lm}\lambda_{\boldsymbol{\kappa}}\lambda_{\boldsymbol{\nu}}\Psi_{\boldsymbol{\kappa}}(\boldsymbol{x}_l)\Psi_{\boldsymbol{\nu}}(\boldsymbol{x}_m)\right).\end{aligned} \tag{48}$$

The covariance matrix for the coefficients of a set of basis functions is conveniently computed as

$$\boldsymbol{\Sigma}_{\boldsymbol{c}} = \operatorname{cov}(\boldsymbol{c},\boldsymbol{c}) = \sigma_{\mathrm{tot}}^2(1-\tau)\left(\boldsymbol{\Lambda} - (1-\tau)\boldsymbol{\Lambda}\boldsymbol{\Psi}\tilde{\boldsymbol{R}}^{-1}\boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{\Lambda}\right). \tag{49}$$

Notably, the result is similar to the posterior covariance matrix for an ensemble of predictions [42], with the kernel matrix evaluated at the training samples replaced by the matrix of the basis functions evaluated at the same points.

Owing to the linearity of (40), the PCE coefficients predicted by the hybrid PCE-GPR method follow a multivariate Gaussian distribution with mean vector (42) and covariance matrix (49). The (co)variance describes the estimation uncertainty related to the limited amount of training data used for the calculation.

5.3. Propagation to UQ measures

The probabilistic information on the PCE coefficients derived in the previous section is in turn propagated to UQ measures on the output quantity. According to (7), the mean μ_y predicted by the PCE coincides with the zero-order coefficient $c_{\boldsymbol{0}}$. As such, its distribution is Gaussian with expected value and variance obtained from (41) and (45) as

$$\mathrm{E}\left(\mu_y\right) = \mathrm{E}\left(c_{\boldsymbol{0}}\right) = \sum_{l=1}^{L}\alpha_l, \tag{50}$$

and

$$\operatorname{Var}\left(\mu_y\right) = \operatorname{cov}\left(c_{\boldsymbol{0}}, c_{\boldsymbol{0}}\right) = \sigma_{\mathrm{tot}}^2(1-\tau)\left(1 - (1-\tau)\sum_{l,m=1}^{L}\left[\tilde{\boldsymbol{R}}^{-1}\right]_{lm}\right), \tag{51}$$

respectively, since $\lambda_{\boldsymbol{0}} = 1$ and $\Psi_{\boldsymbol{0}} = 1$.

The variance σ_y^2 is instead given by (8), i.e., the sum over the remaining coefficients squared. Hence, it can be trivially expressed as the quadratic form

$$\sigma_y^2 = \boldsymbol{c}_{-\boldsymbol{0}}^{\mathsf{T}}\boldsymbol{c}_{-\boldsymbol{0}} \tag{52}$$

where $\boldsymbol{c}_{-\mathbf{0}}$ denotes the vector of PCE coefficients excluding the zero-order one. Following the properties of quadratic forms [90], the expected value and the variance of (52) are computed as

$$\mathrm{E}\left(\sigma_y^2\right) = \mathrm{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{c}_{-\mathbf{0}}}\right) + \boldsymbol{\mu}_{\boldsymbol{c}_{-\mathbf{0}}}^{\mathsf{T}}\boldsymbol{\mu}_{\boldsymbol{c}_{-\mathbf{0}}} \tag{53}$$

and

$$\mathrm{Var}\left(\sigma_y^2\right) = \mathrm{tr}\left(2\boldsymbol{\Sigma}_{\boldsymbol{c}_{-\mathbf{0}}}^2\right) + 4\boldsymbol{\mu}_{\boldsymbol{c}_{-\mathbf{0}}}^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{c}_{-\mathbf{0}}}\boldsymbol{\mu}_{\boldsymbol{c}_{-\mathbf{0}}}, \tag{54}$$

respectively, where $\boldsymbol{\mu}_{\boldsymbol{c}_{-\mathbf{0}}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{c}_{-\mathbf{0}}}^2$ denote the mean vector (42) and covariance matrix (49) of the PCE coefficients with the zero-order coefficient removed.

It should be noted that the results for the variance applies because the posterior distribution of the estimated PCE coefficients is Gaussian. Moreover, the expected value of the variance is not just the sum of the squares of the expected values of the coefficients, but there is an additional contribution due to the trace of the covariance matrix. Finally, it is relevant to mention that the distribution of a quadratic form such as (52) is a generalized chi-square [91] and can be derived analytically [60]. Nevertheless, we will show that the variance already provides a good indication of the confidence level, especially when the number of training samples increases.

A similar reasoning is used to obtain the expected value and variance of the Sobol' indices, by considering in the quadratic form only the subset of coefficients indexed by the pertinent subset $\mathcal{K}_i$. For the sake of simplicity, we calculate the result for the non-normalized indices, corresponding to the numerator of (9). Accounting for the denominator, which is also stochastic, would be highly non-trivial, as no closed-form properties are available for the ratio of two random variables. However, this is not a limitation since one is typically interested in ranking the indices rather than assessing their absolute value. This information is not affected as the denominator is common to all indices. The result is

$$\mathrm{E}\left(S_{Ti}\right) = \mathrm{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{c}_i}\right) + \boldsymbol{\mu}_{\boldsymbol{c}_i}^{\mathsf{T}}\boldsymbol{\mu}_{\boldsymbol{c}_i} \tag{55}$$

and

$$\mathrm{Var}\left(S_{Ti}\right) = \mathrm{tr}\left(2\boldsymbol{\Sigma}_{\boldsymbol{c}_i}^2\right) + 4\boldsymbol{\mu}_{\boldsymbol{c}_i}^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{c}_i}\boldsymbol{\mu}_{\boldsymbol{c}_i}, \tag{56}$$

where $\boldsymbol{\mu}_{\boldsymbol{c}_i}$ and $\boldsymbol{\Sigma}_{\boldsymbol{c}_i}$ denote the mean vector and covariance matrix of the PCE coefficients indexed by the multi-indices in subset $\mathcal{K}_i$, which are a suitable selections of rows and columns of $\boldsymbol{\mu}_{\boldsymbol{c}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{c}}$.

At this point, it is important to highlight a fundamental difference with the approach in [60], which is based on MC sampling and whose accuracy and efficiency are therefore affected by the MC sample size (which is, however, not accounted for by the predictive variance). Here, the calculation is analytical and the only source of uncertainty is in the training sample size (besides the estimation of the hyperparameters). In this regard, it is important to recall that the expansion order (which also affects the accuracy of the prediction) can be increased at will, e.g., until the magnitude of the coefficients drops below some tolerance. Therefore, the predicted statistics are directly comparable to their exact value rather than to their MC estimates, as we will show in the application examples.

Finally, it is worth mentioning that the probabilistic model also allows for a convenient numerical calculation of confidence bounds for the PDF of y. As we will shown in the application examples, one can generate random realizations of posterior GPR trajectories for an ensemble of MC predictions or, equivalently, of PCE coefficients and evaluate repeatedly (2). In either case, each realization produces a different estimate of the PDF, and confidence bounds or statistical moments are readily obtained from the ensemble of predictions.

6. Multi-output model and implementation

So far, we assumed the output of (1) to be a scalar. We now extend the analysis to a multi-output system. Therefore, we assume a training dataset for the multiple outputs be available in the form of a matrix $\boldsymbol{Y} \in \mathbb{R}^{L\times P}$, where P is the number of output components. For example, the dataset $\boldsymbol{Y}$ could collect different system variables and/or a transient response evaluated at several time points for each training configuration of the uncertain system parameters.

The first observation is that (39) is readily extended to a matrix dataset by replacing the training vector $\boldsymbol{y}$ with $\boldsymbol{Y}$, thereby producing a matrix of kernel coefficients

$$\boldsymbol{A} = (1-\tau)\tilde{\boldsymbol{R}}^{-1}\boldsymbol{Y} \tag{57}$$

of size $L \times P$, where each column represents the vector of GPR coefficients for the corresponding output component. At this point, the main issue becomes the tuning of the hyperparameters. We propose a simple approach in which a single set of hyperparameters is trained for all outputs by means of a simple modification of the LOO-CV error (25), i.e.,

$$\epsilon_{\mathrm{CV}} = \sum_{p=1}^{P}\sum_{k=1}^{L}\sum_{l\neq k}\left(Y_{lp} - m_{-k,p}(\boldsymbol{x}_l)\right)^2, \tag{58}$$

where $m_{-k,p}$ denotes the prediction of the pth output provided by the GPR model trained without the kth training sample. In fact, the error is cumulated over all output components. The hyperparameters $\boldsymbol{\rho}$ and the noise ratio τ are computed by minimizing the above error, and the total variance is obtained for each output in analogy with (26):

$$\sigma^2_{\text{tot},p} = \frac{1}{L}\sum_{k=1}^{L}\sum_{l\neq k}\frac{\left(Y_{lp} - m_{-k,p}(\boldsymbol{x}_l)\right)^2}{c^2_{-k}(\boldsymbol{x}_l,\boldsymbol{x}_l)}. \tag{59}$$

It should be noted that the denominator c_{-k} in (59) is the same for each output component, as (21) depends only on the location of the training samples, but not on the corresponding output observation. Moreover, it is important to point out that, even though the same noise ratio τ is considered for all outputs, the kernel and noise variances will differ as they are proportional to (59). The alternative ML estimation (24) is also readily extended to multiple output data. Once the model is trained, the expected values of the output-dependent PCE coefficients are obtained by replacing $\boldsymbol{\alpha}$ with $\boldsymbol{A}$ in (42). A common correlation matrix is obtained from (49) by neglecting the total variance, which is then included to individually retrieve the covariance matrix for each output.

The outlined approach offers a viable solution with a fast training, even for a large number of outputs. Possible alternatives include training a separate model for each output, which is however rather inefficient, or using PCA to compress the dataset $\boldsymbol{Y}$ and training a separate model for each principal component [61]. A comparison between the aforementioned strategies will be provided for one of the application examples.

The PCE-GPR method is implemented in MATLAB via user-defined codes. In this case, the hyperparameters are estimated by minimizing the LOO-CV error using Bayesian optimization (BO). Alternatively, any GPR toolbox that allows the definition of a custom kernel function can be used to train the model. For instance, the Kriging module in UQLab [84] is an effective option that provides tools for estimating the hyperparameters using either ML or CV with any number of folds, as well as several optimization methods. However, it does not support BO or the training of a multi-output model as described above. In the following, we will use the custom implementation to illustrate the simpler test cases, while we will take advantage of the well-consolidated UQLab toolbox for the high-dimensional examples.

7. Illustrative examples

We first illustrate the method based on two simple analytical functions, which allow carrying out extensive analyses and comparisons.

7.1. One-dimensional function

We define the synthetic function

$$y = \mathcal{M}_1(x) = e^{-10/x}\left(2\cos(x) + \sin(x)\right), \tag{60}$$

with $x \sim \mathcal{U}(1,9)$. The function is representative of the response of an underdamped second-order dynamical system. The exact expected value and variance of (60) are computed with an accurate numerical integration to be $\mu_y = 0.0703$ and $\sigma^2_y = 0.0572$, respectively.

We aim to compute the same statistical information using a Legendre PCE. For the numerical methods, we consider three training datasets with $L = 5$, $L = 7$, and $L = 10$ samples, linearly spaced within the uncertainty interval. We shall compare the results of the proposed PCE-GPR method against the state-of-the-art LAR, OMP, SP, and BCS methods, which are all implemented via the PCE module in UQLab [92]. The SP is implemented in conjunction with a LOO-CV, since it was shown in [16] to provide better results compared to using a lower number of folds. To simplify the investigations, in this first analysis we neglect noise on the training data (i.e., we set $\sigma_n = 0$). The only hyperparameter ρ is estimated via LOO-CV error minimization and BO. The total variance is computed with (26).

7.1.1. PCE coefficients, statistical moments, and distribution

Fig. 2 illustrates the prediction of the target function (60) (solid blue line) based on the different training samples (red dots). The top panels compare the kernel-based prediction provided by (38) (green dash-dotted lines), corresponding to a PCE of infinite order, against the predictions with the expansion coefficients retrieved in post processing up to order $p = 3$ ($\triangle$), $p = 5$ ($\circ$), and $p = 10$ ($\square$). It is interesting to note that, for $L = 7$ and $L = 10$, the model of order 10 is indistinguishable from the full kernel model of order infinity, which suggests that the former provides virtually the same accuracy as the latter. This is a possible criterion to determine, a posteriori, the maximum expansion order up to which to calculate the coefficients. For the case with $L = 5$ instead, the kernel model turns out to be less smooth in order to interpolate the few training data (as implied by setting $\sigma_n = 0$), and therefore it is not very well represented even by a high-order expansion.

The bottom panels show the 95% confidence interval of the predictions instead. The green area with solid bounds indicate the classical confidence interval of the kernel model obtained from the posterior covariance (20). The red area with dashed bounds is instead obtained numerically from the tenth-order expansion model (2) after drawing a large number of random posterior realizations of the coefficients according to their expected value (42) and covariance matrix (49). It is observed that the two confidence interval match except for the case with $L = 5$, again to due the limited accuracy of the tenth-order approximation in that case. This comparison provides an indirect validation for the posterior model of the PCE coefficients. Overall, the prediction is already very good for $L = 7$, and the confidence interval shrinks significantly around the target function for $L = 10$.

Using a similar random sampling of the PCE coefficients, we assess the dispersion of the PDF resulting from the uncertainty in the estimate of such coefficients. To this end, for each random realization of the coefficients, the PCE model (2) is evaluated for a set of 1000 MC samples, and a kernel density estimate of the distribution is produced. Confidence bounds are then obtained from

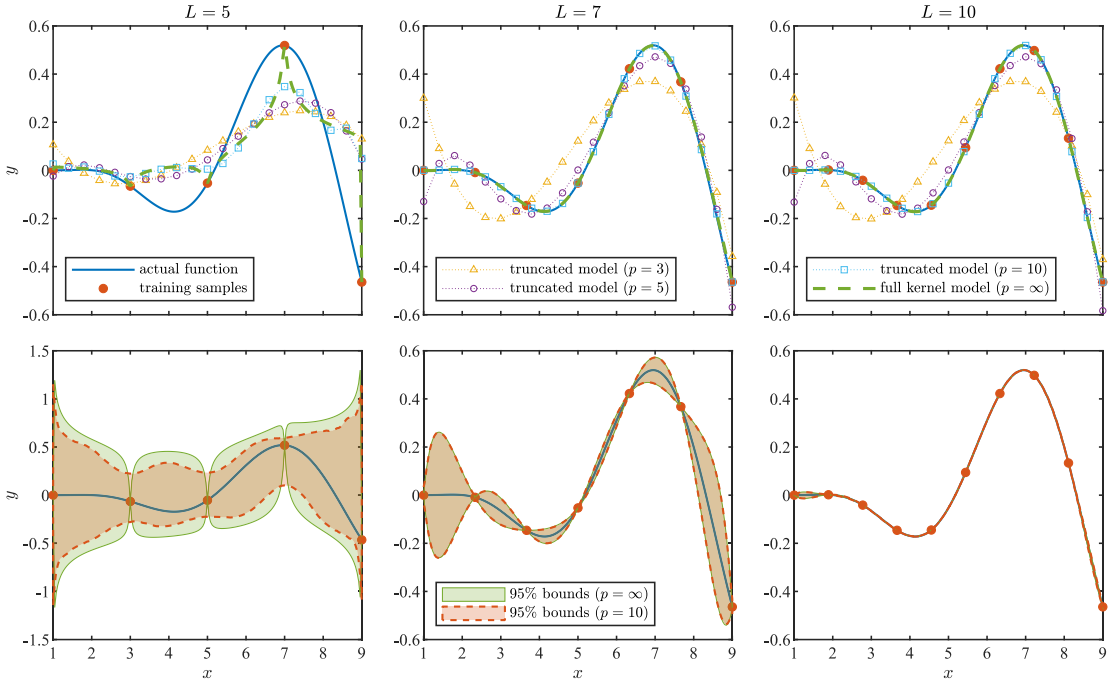

Fig. 2. PCE-GPR prediction of function $\mathcal{M}_1$ in (60) with different training set sizes. The actual function is shown by the solid blue lines, whereas the red dots indicate the training samples. The top panels show the prediction of the full kernel model (dash-dotted green lines) and of the model (dotted lines) truncated to order three (△), five (○), or ten (□). The bottom panels show the 95% confidence interval of the full kernel model (shaded green area with solid edges) and of the tenth-order expansion model (shaded red area with dashed edges). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

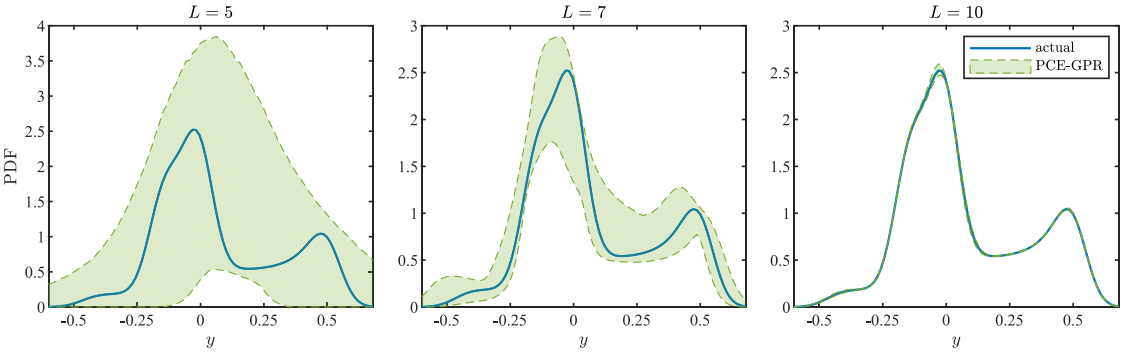

Fig. 3. PDF of function $\mathcal{M}_1$. The 95% confidence interval of the PCE-GPR prediction for different training datasets (shaded green area) is compared against the reference MC distribution (solid blue line).

the ensemble of predictions. The results are shown in Fig. 3, where the solid blue line indicates the reference distribution obtained by sampling the actual function with the same MC samples. The green shaded area represents instead the 95% confidence interval of the tenth-order PCE-GPR prediction. Also for the distribution, it is found that the confidence interval shrinks around the correct distribution as the number of training samples is increased. This is the case also for $L = 5$, although the confidence interval is rather wide to reflect the large model uncertainty.

Table 1 summarizes the main results regarding the prediction of the mean and variance of (60) as well as the root-mean-square error (RMSE) in comparison with the state-of-the art LAR, OMP, SP, and BCS techniques. The RMSE between the actual function

Table 1
Mean, variance, and RMSE obtained by LAR, OMP, SP, BCS, standard GPR, and PCE-GPR in comparison with the MC and exact results.

Observations	Method	μ_y	σ_y^2	RMSE
–	Exact	0.0703	0.0572	–
$N = 1000$	MC	0.0703	0.0573	–
$L = 5$	LAR	0.1152	**0.0550**	$\mathbf{9.2632 \times 10^{-2}}$
	OMP	0.1152	**0.0550**	$\mathbf{9.2632 \times 10^{-2}}$
	SP	0.0000	0.0074	2.6403×10^{-1}
	BCS	**0.0886**	0.0782	1.6386×10^{-1}
	Standard GPR	0.0048	0.0922	1.7395×10^{-1}
	PCE-GPR	0.0953	0.0490	1.6568×10^{-1}
	95% bounds	$[-0.0914, 0.2820]$	$[0.0152, 0.1107]$	
	2-sigma bounds		$[0.0003, 0.0977]$	
$L = 7$	LAR	0.0168	0.0000	2.4509×10^{-1}
	OMP	0.0841	0.0467	7.8563×10^{-2}
	SP	0.0000	0.0677	1.6496×10^{-1}
	BCS	0.0974	0.0725	1.4303×10^{-1}
	Standard GPR	**0.0711**	**0.0573**	4.2356×10^{-3}
	PCE-GPR	0.0714	0.0592	$\mathbf{3.9184 \times 10^{-3}}$
	95% bounds	$[0.0416, 0.1013]$	$[0.0499, 0.0747]$	
	2-sigma bounds		$[0.0465, 0.0720]$	
$L = 10$	LAR	0.0959	0.0507	7.2526×10^{-2}
	OMP	0.0678	0.0586	9.8370×10^{-3}
	SP	0.0759	0.0736	1.4942×10^{-1}
	BCS	0.0709	0.0570	1.6007×10^{-3}
	Standard GPR	**0.0703**	**0.0572**	5.1019×10^{-4}
	PCE-GPR	0.0702	0.0573	$\mathbf{3.5426 \times 10^{-4}}$
	95% bounds	$[0.0691, 0.0714]$	$[0.0567, 0.0579]$	
	2-sigma bounds		$[0.0567, 0.0578]$	

values y_i and the model predictions $\hat{y}_i$ is computed as

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{61}$$

based on the set of $N = 1000$ MC samples. Boldface is used to highlight the most accurate result. The PCE-GPR predictions are given in terms of expected value as well as both the 95% and 2-sigma confidence intervals, computed from the tenth-order PCE. The calculation avoids the need for sampling the GPR posterior for a large number of MC samples, as was instead required by the approach in [60]. While the two confidence intervals coincide for the prediction of the mean μ_y, owing to the Gaussian posterior distribution, this is not the case for the prediction of the variance σ_y^2, which follows a generalized chi-square distribution. The "Generalized chi-square distribution" toolbox [93] is used to handle the predictive distribution of the variance and, later on, of the Sobol' indices.

It is found that the 95% and 2-sigma bounds approach when the number of training samples is increased, a circumstance that was already observed in [60]. As far as the accuracy is concerned, the PCE-GPR method substantially outperforms the alternative PCE-based methods for the second and third dataset, for which it achieves an RMSE that is one or two orders of magnitude lower. The confidence intervals are also well representative of the prediction accuracy, since they always include the reference result and narrow as the number of training samples is increased. For further comparison, the results obtained using a standard GPR formulation with a constant trend and a squared-exponential kernel are also reported. It is found that the two methods achieve similar accuracy, with PCE-GPR yielding a slightly lower RMSE.

Fig. 4 shows the individual PCE coefficients instead. The exact coefficients, obtained by computing the projection (10) via an accurate numerical integration, are indicated by the blue circles. The red crosses are the coefficients estimated with the best alternative method according to Table 1, namely LAR for $L = 5$, OMP for $L = 7$, and BCS for $L = 10$. The green asterisks denote the coefficients retrieved analytically from the PCE-GPR, with the green bars indicating their 95% confidence interval. The proposed PCE-GPR method achieves a much better result especially for the dataset with $L = 7$ samples, with OMP failing to estimate most coefficients. Moreover, the confidence interval provides a good indication of the prediction uncertainty, with the correct result falling always inside it, even for the case with $L = 5$ samples.

7.1.2. Repeated-run analysis

Finally, we perform a repeated-run analysis to assess the robustness of the model predictions against different training datasets. We consider 100 independent runs. For each run, the training samples are now drawn randomly according to the input distribution. The aim of this analysis is twofold, since it allows assessing both the dispersion of the prediction error across different datasets and the actual confidence level of the PCE-GPR method.

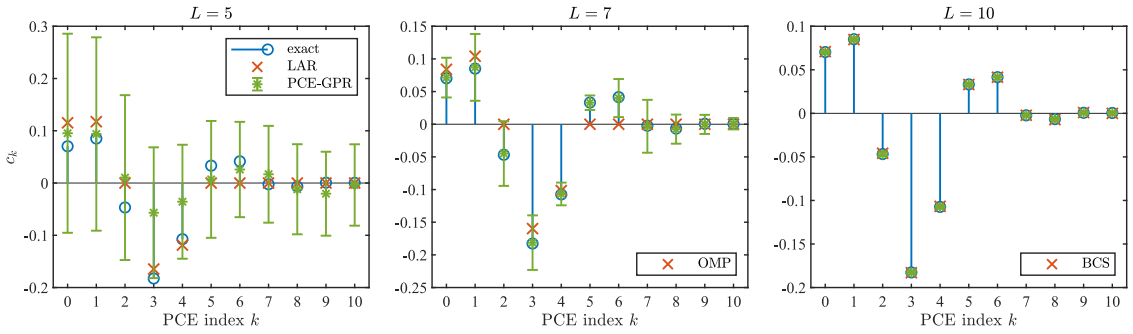

Fig. 4. Coefficients of the tenth-order PCE estimated with PCE-GPR (green asterisks with 95% confidence interval) and LAR, OMP, or BCS (red crosses) using training sets of different sizes, compared to the exact values (blue circles).

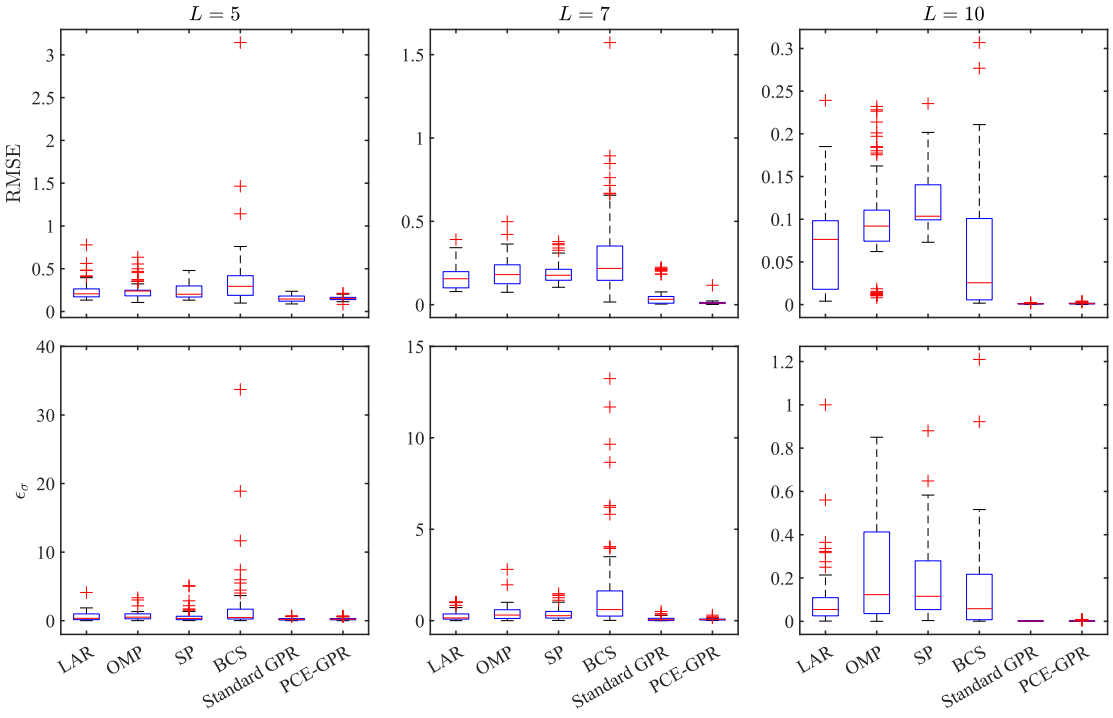

Fig. 5. Boxplot illustrating the dispersion of the RMSE (top panels) and ϵ_σ (bottom panels) across 100 training datasets for the various methods and sample sizes.

For the error analysis, we define two error metrics: (1) the RMSE, computed as in (61); (2) the relative error on the predicted variance, defined as

$$\epsilon_\sigma = \frac{|\sigma_y^2 - \hat{\sigma}_y^2|}{\sigma_y^2}, \tag{62}$$

where $\hat{\sigma}_y^2$ and σ_y^2 denote the predicted variance and the corresponding reference value, respectively. Besides LAR and PCE-GPR with a tenth-order expansion, we also consider a standard GPR model with a constant trend and a squared-exponential kernel. While for the first two methods the predicted variance can be compared directly to the exact value, the prediction of standard GPR is computed based on the MC samples as in [60], and therefore its reference value is the MC variance. However, as shown in Table 1, the difference between MC estimations and actual values is minimal.

The boxplot in Fig. 5 illustrates the dispersion of the two error metrics across the various training datasets of increasing size and for all the methods considered in the analysis. The red line indicates the median, the bottom and top edges are the 25th and 75th percentiles, respectively, whereas the whiskers indicate the minimum and maximum values. The red crosses are outliers. It should be noted that, for $L = 5$ and $L = 7$, the BCS results for ϵ_σ contain further outliers beyond the upper limit of the plots, which are omitted for the sake of graph readability. It is observed that PCE-GPR achieves a significantly lower median error compared to the

Table 2
Mean, variance, Sobol' indices, and RMSE obtained by the various methods in comparison with the MC and exact results.

Observations	Method	μ_y	σ_y^2	$S_{T_1} \cdot \sigma_y^2$	$S_{T_2} \cdot \sigma_y^2$	RMSE
–	Exact	−0.0337	0.0680	–	–	–
$N = 10^4$	MC	−0.0349	0.0685	0.0251	0.0673	–
$L = 25$	LAR	−0.0350	0.0671	0.0263	0.0657	5.1122×10^{-2}
	OMP	−0.0339	0.0813	0.0359	0.0798	9.4963×10^{-2}
	SP	−0.0382	0.0690	0.0268	0.0684	8.1326×10^{-2}
	BCS	−0.0365	0.0724	0.0278	0.0714	6.5236×10^{-2}
	Standard GPR	−0.0341	0.0701	0.0258	0.0687	$\mathbf{1.0108 \times 10^{-2}}$
	PCE-GPR	**−0.0338**	**0.0687**	**0.0253**	**0.0680**	1.5615×10^{-2}
	95% bounds	[−0.0373, −0.0304]	[0.0666, 0.0711]	[0.0238, 0.0269]	[0.0659, 0.0704]	
	2-sigma bounds		[0.0665, 0.0710]	[0.0237, 0.0268]	[0.0658, 0.0703]	
$L = 50$	LAR	−0.0388	0.0726	0.0273	0.0717	5.9159×10^{-2}
	OMP	−0.0390	0.0785	0.0348	0.0776	7.4573×10^{-2}
	SP	−0.0413	0.0736	0.0311	0.0726	8.8357×10^{-2}
	BCS	−0.0383	0.0801	0.0355	0.0792	7.6400×10^{-2}
	Standard GPR	−0.0349	0.0684	0.0250	0.0672	$\mathbf{2.8734 \times 10^{-3}}$
	PCE-GPR	**−0.0338**	**0.0680**	**0.0251**	**0.0674**	3.3716×10^{-3}
	95% bounds	[−0.0341, −0.0336]	[0.0679, 0.0681]	[0.0251, 0.0252]	[0.0673, 0.0674]	
	2-sigma bounds		[0.0679, 0.0681]	[0.0251, 0.0252]	[0.0673, 0.0674]	
$L = 100$	LAR	−0.0347	0.0684	0.0256	0.0678	1.8528×10^{-2}
	OMP	−0.0325	0.0693	0.0255	0.0686	3.1104×10^{-2}
	SP	−0.0316	0.0706	0.0256	0.0696	6.4845×10^{-2}
	BCS	−0.0331	0.0686	0.0253	0.0679	1.5642×10^{-2}
	Standard GPR	−0.0348	0.0685	0.0251	0.0673	8.9708×10^{-4}
	PCE-GPR	**−0.0337**	**0.0680**	**0.0251**	**0.0674**	$\mathbf{4.7480 \times 10^{-4}}$
	95% bounds	[−0.0337, −0.0337]	[0.0680, 0.0680]	[0.0251, 0.0252]	[0.0674, 0.0674]	
	2-sigma bounds		[0.0680, 0.0680]	[0.0251, 0.0252]	[0.0674, 0.0674]	

alternative PCE-based techniques. Moreover, the dispersion is substantially lower for all training datasets. Compared to the standard GPR, the median error is similar, but a smaller dispersion, without the many outliers, is observed for the case $L = 7$. Overall, the performance of PCE-GPR is found to be significantly better than the state-of-the-art PCE-based techniques and comparable to or better than standard GPR.

This analysis also offers the opportunity to assess the actual confidence level of the PCE-GPR predictions. Results in this regard are provided in Appendix, where we also assess the posterior distribution of the predicted mean and variance, and the impact of the hyperparameter ρ, and the actual confidence level of the PCE-GPR predictions.

7.2. Two-dimensional function

We define the following modification of (60):

$$y = \mathcal{M}_2(x_1, x_2) = e^{-10/x_1} \left(2\cos(x_2) + \sin(x_2)\right), \tag{63}$$

with two independent parameters, $x_1 \sim \mathcal{U}(1, 9)$ and $x_2 \sim \mathcal{N}(5, 1)$. The exact mean and variance are computed to be $\mu_y = -0.0337$ and $\sigma_y^2 = 0.0680$, respectively.

For this second test function, we compute a mixed Legendre-Hermite PCE up to order $p = 8$. The expansion features $|\mathcal{K}| = 45$ when a total degree truncation is used. We consider again three training datasets, with $L = 25$, $L = 50$, and $L = 100$ samples randomly drawn from the input distribution using a Latin hypercube strategy. For PCE-GPR, we use again $\sigma_n = 0$ and an isotropic kernel with its hyperparameter ρ estimated via LOO-CV. For standard GPR, we assume a constant trend and an isotropic squared-exponential kernel.

Table 2 collects the results regarding the prediction of the mean and the variance. The results of a MC analysis based on $N = 10^4$ samples are also provided for reference. Although not so relevant for a two-dimensional scenario, the non-normalized total Sobol' indices are also included in the table. In this case, the reference result is provided by their MC estimate, computed as in [85]. Several observations can be made. First of all, the PCE-GPR method always achieves a lower RMSE compared to the alternative PCE-based techniques, with the former being two orders of magnitude more accurate for the largest dataset. The confidence intervals of PCE-GPR steadily shrink by increasing the sample size and always include the reference result. The difference between the 95% and 2-sigma confidence intervals for the variance and Sobol' indices reduces again as the number of training samples is increased. An excellent accuracy is obtained also for the estimate of the Sobol' indices. This is particularly relevant, as their MC estimation requires $4 \cdot 10^4$ function evaluations, whereas it is inexpensive with PCE-based methods.

Compared to standard GPR, the proposed PCE-GPR yields a better accuracy in the prediction of statistical information. In this regard, it is noted that the mean and the variance predicted by PCE-GPR converge to the exact values, as expected. In contrast, the prediction of standard GPR relies on a MC sampling of the posterior following to the numerical approach in [60], rather than on analytical calculations. Therefore, it converges to the MC estimates, which exhibit some difference w.r.t. the exact results due to the

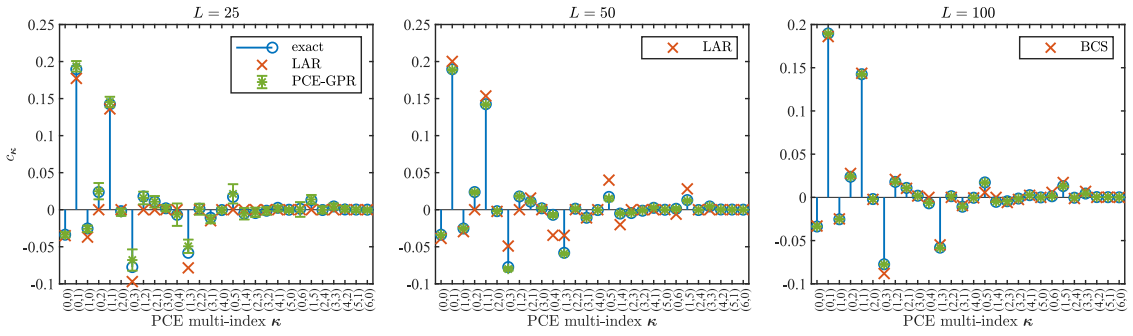

Fig. 6. Coefficients of the eight-order PCE of function $\mathcal{M}_2$ in (63), estimated with LAR or BCS (red crosses) and PCE-GPR (green asterisks with 95% confidence interval) using training sets of different sizes, and compared to the exact values (blue circles).

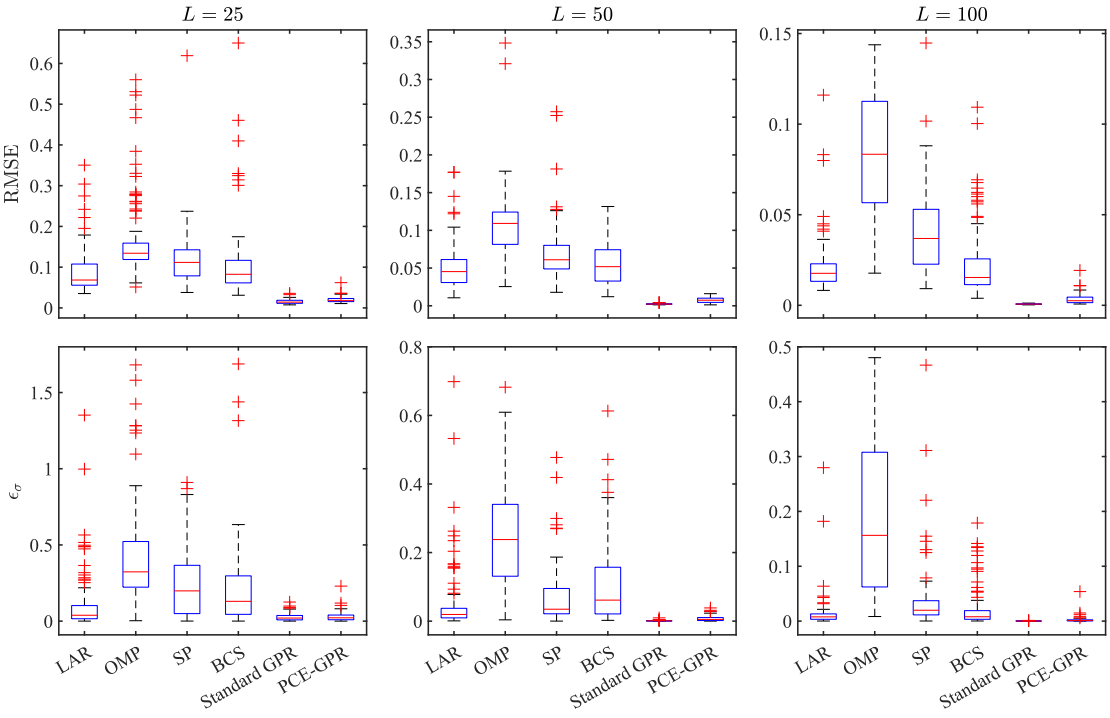

Fig. 7. Boxplot illustrating the dispersion of the RMSE (top panels) and ϵ_σ (bottom panels) for the two-dimensional function (60) across 100 training datasets for the various methods and sample sizes.

very large variability of this test function. The same consideration applies also to Sobol' indices. Standard GPR appears to compare better with the reference results, which however are based on MC sampling. If the same indices are computed numerically by sampling the PCE-GPR posterior instead of considering their analytical calculation, a lower error is obtained compared to standard GPR.

In Fig. 6, the PCE coefficients estimated with the most accurate state-of-the-art technique (red crosses) and PCE-GPR (green asterisks with 95% error bars) are compared to the exact result obtained by numerical integration (blue circles). For the ease of visualization, only the coefficients up to order 6 are displayed. The comparison highlights the remarkable accuracy achieved by the proposed PCE-GPR method, which produces a very good estimate already with the smallest training dataset. The difference with the exact result becomes indistinguishable for $L = 50$, whereas the LAR and BCS results exhibit some discrepancy even for $L = 100$ samples.

Finally, we assess the dispersion of the RMSE and the variance prediction error ϵ_σ across different training datasets. To this end, we perform the simulation for 100 independent runs. As in the previous example, we also consider a standard GPR formulation with a constant trend and a squared-exponential kernel. The results are summarized by the boxplot in Fig. 7. It is again established that the proposed PCE-GPR method achieves both lower error and dispersion compared to all the alternative PCE-based techniques,

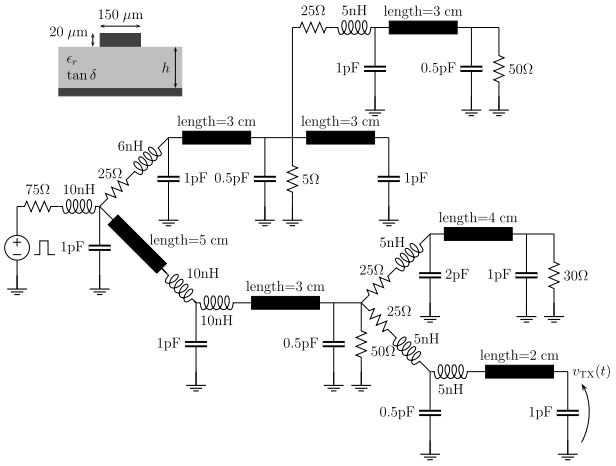

Fig. 8. Schematic of the network with single transmission lines. The cross-section of the microstrip lines is depicted on the top left.

i.e., LAR, OMP, SP, and BCS. The performance is in this case comparable with the standard GPR which, however, does not provide a closed-form estimation of any statistical information.

8. Application examples

This section discusses relevant application examples in the field of electrical engineering and related specifically to the UQ of electronic circuits affected by manufacturing or design tolerances. The simulations are performed on a workstation equipped with an Intel® Core™ i7-9700F processor, CPU running at 3 GHz, and 32 GB of RAM.

8.1. Network with single transmission lines

The first application test case deals with the circuit depicted in Fig. 8. The network is a combination of lumped components and microstrip transmission line segments. The uncertainty is provided by the substrate of the microstrip, which is common to all line sections, and specifically by the substrate thickness h, the relative permittivity ϵ_r, and the loss tangent $\tan\delta$. These parameters exhibit a Gaussian distribution with nominal values $\bar{h} = 100$ μm, $\bar{\epsilon}_r = 4.1$, and $\overline{\tan\delta} = 0.02$, and a relative standard deviation of 10%. The number of uncertain parameters is thus $d = 3$. The voltage source has a pulse waveform with an amplitude of 1 V, rise/fall times of 0.2 ns, and a width of 2.8 ns at half amplitude.

The network is described by a system of ordinary differential equations in the form of

$$-\boldsymbol{C}\frac{d}{dt}\boldsymbol{y}(t) = \boldsymbol{G}\boldsymbol{y}(t) + \boldsymbol{u}(t) \tag{64}$$

and describing the transient evolution of node voltages and some of the element currents, stored in vector $\boldsymbol{y}$. The system matrices $\boldsymbol{C}$ and $\boldsymbol{G}$ depend on the circuit element values, and hence also on the uncertain parameters. The forcing term $\boldsymbol{u}$ collects the independent sources, in this case the voltage source to the left, and is considered to be deterministic. We focus the transient analysis on the voltage $v_{\mathrm{TX}}(t)$ transmitted to the bottom right of the circuit and indicated in Fig. 8. The simulations are carried out using the well-known Simulation Program with Integrated Circuit Emphasis (SPICE), specifically a commercial version called HSPICE [94]. The software allows running simulations for a user-defined sweep of selected variables and it is therefore used as a black-box solver.

We aim at computing a second-order Hermite PCE ($p = 2$) with total degree truncation, which features $|\mathcal{K}| = 10$ terms. For the PCE-GPR method, we consider two training datasets with $L = 5$ and $L = 10$ training samples drawn according to the input distribution using a Latin hypercube strategy. We use the multi-output implementation discussed in Section 6 with an isotropic kernel and $\sigma_n \neq 0$. The kernel hyperparameter ρ and the noise ratio τ are estimated using LOO-CV. For this circuit, reference results are computed using the SGM-based method in [19], an element-wise approach that consists in deriving an augmented equivalent circuit formed by circuit interpretation of the SGM equations of each stochastic element. The method readily provides a very accurate calculation of the transient PCE coefficients of the circuit responses by means of a single deterministic simulation.

Fig. 9 shows in the left panel a subset of 100 responses (solid gray lines) from a MC simulation with 10^4 samples. The mean of the MC samples (thick blue line) is compared against the estimates provided by the zero-order PCE coefficients computed with the SGM simulation (dashed red line) and with the proposed PCE-GPR method trained with $L = 5$ samples only (shaded green area, representing the 95% confidence interval). The inset allows appreciating the accuracy of the PCE-GPR prediction, with a narrow confidence interval encompassing the reference MC and SGM results.

The right panels of Fig. 9 show instead the transient behavior of the three most relevant PCE coefficients, i.e., $c_{(1,0)}$, $c_{(0,1)}$, and $c_{(2,1)}$. The results of the SGM simulation (solid, dashed, and dotted red lines) are compared to the PCE-GPR predictions obtained

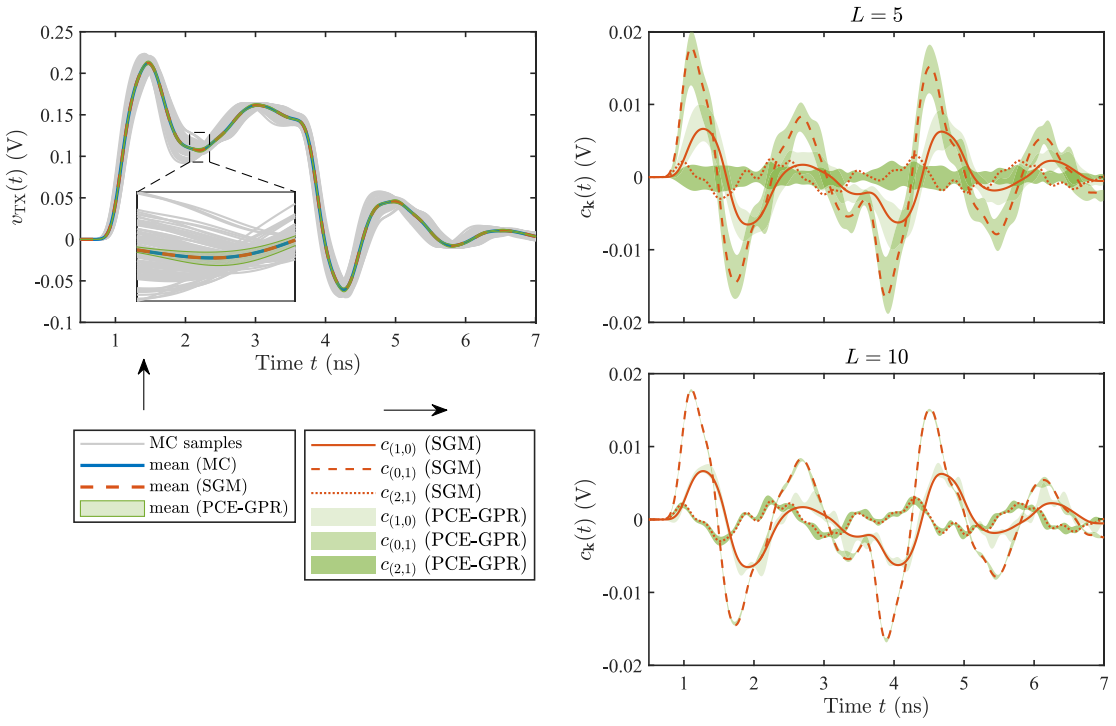

Fig. 9. Transient transmitted voltage in the network with single transmission lines of Fig. 8. Left panel: subset of MC responses (gray lines) and average response obtained from the MC samples (thick blue line), with the SGM (dashed red line), and with PCE-GPR (shaded green area). Right panels: transient PCE coefficients $c_{(10)}$, $c_{(0,1)}$, and $c_{(2,1)}$ obtained with the SGM simulation (red lines) compared against the PCE-GPR calculation (shaded green areas) with $L = 5$ (top) and $L = 10$ training samples (bottom). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with $L = 5$ (top) and $L = 10$ training samples (bottom), provided in terms of 95% confidence interval (green areas of different shades). It is interesting to note that the confidence bounds accurately enclose the reference SGM result for both training dataset. The calculation becomes more accurate (the confidence interval narrows) for the largest dataset.

A similar comparison is provided in Fig. 10 for the variance (top panels) and the Sobol' indices (bottom panels). In the former case, the SGM prediction (dashed red lines) and 2-sigma confidence intervals of PCE-GPR (shaded green areas) are compared against the MC variance (solid blue line). As expected, the SGM provides a very accurate result for this circuit, and the PCE-GPR result compares very well. As to the Sobol' indices, the reference is provided by the SGM-based calculation. The non-normalized Sobol' indices S_{T_1} (solid line), S_{T_2} (dashed line), and S_{T_3} (dotted line) refer to the substrate thickness h, relative permittivity ϵ_r, and loss tangent $\tan\delta$, respectively. The relative permittivity and the loss tangent are found to provide the largest and the least contribution to the output variance, respectively, a result that is well expected. Indeed, the relative permittivity mainly affects the propagation velocity along the microstrip sections, and hence the propagation delay. On the contrary, the loss tangent plays a marginal role except for very high frequency signals. The 2-sigma confidence bounds of the PCE-GPR prediction (shaded green areas) compare once again very well with the reference result and provide a good indication of the accuracy of the estimate.

Next, we investigate the impact of various multi-output modeling strategies. Fig. 11 shows the RMSE obtained for the largest training dataset with the proposed multi-output model outlined in Section 6 (dotted green line), with PCA compression (dashed red line), and by separately training a single model for each time point (solid blue line). Some interesting conclusions can be drawn. In principle, training a single model for each output is the least efficient but also, potentially, the most accurate approach, since a separate set of hyperparameters is estimated for each component. Nevertheless, the behavior of the RMSE over time turns out to be extremely irregular and spiky, occasionally exceeding the error achieved by the other two approaches. Conversely, both PCA and the proposed approaches exhibit a much more regular behavior over time. Hence, estimating a common set of hyperparameters for all outputs (or for a reduced set, as in the case with PCA) seems to smooth out the error across the components. As summarized in Table 3, the PCA overall achieves the best performance in terms of both average and maximum RMSE. On the other hand, the multi-output model is the most efficient in terms of computational time, especially when compared to the naive approach that trains a separate model for each component, since it simultaneously trains a model for all the 1001 time points. In contrast, PCA requires to train 5 single-output models (one for each "principal component"). Therefore, the multi-output PCE-GPR formulation is established as a viable and efficient method that avoids the truncation involved in PCA, whose effect on the accuracy is in general difficult to quantify and is not accounted for by the prediction uncertainty.

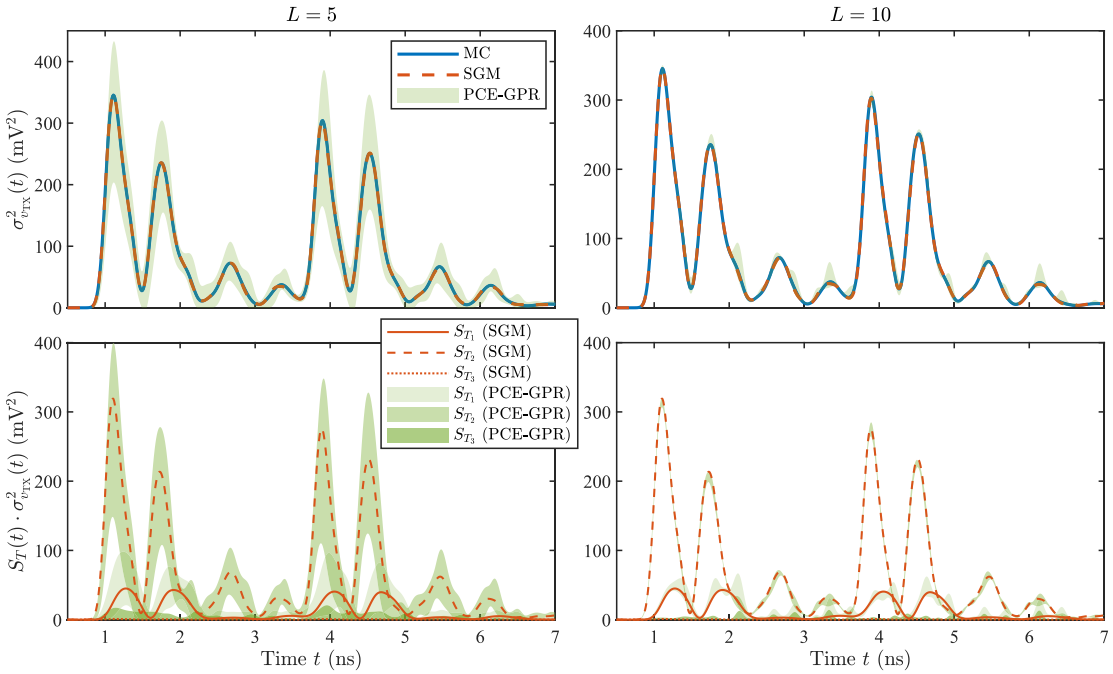

Fig. 10. Variance (top panels) and Sobol' sensitivity indices (bottom panels) of the transient transmitted voltage v_{TX}. The 2-sigma interval of the PCE-GPR prediction (shaded green areas) are compared with the reference MC and SGM results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

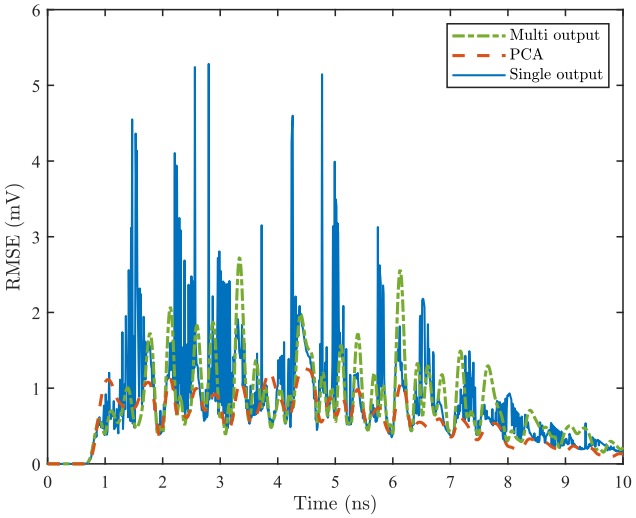

Fig. 11. RMSE over time for three different approaches to handle multiple data: multi-output model of Section 6 (dash-dotted green line), PCA compression (dashed red line), and single-output model for each component (solid blue line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Average and maximum RMSE and training time for the various multi-output modeling approaches.

Method	Average RMSE (V)	Maximum RMSE (V)	CPU Time
Multi-output model of Section 6	7.7007×10^{-4}	2.7227×10^{-3}	6.7 s
PCA compression	5.6701×10^{-4}	1.2636×10^{-3}	23.2 s
Single model for each output	8.4223×10^{-4}	5.2830×10^{-3}	4245.9 s

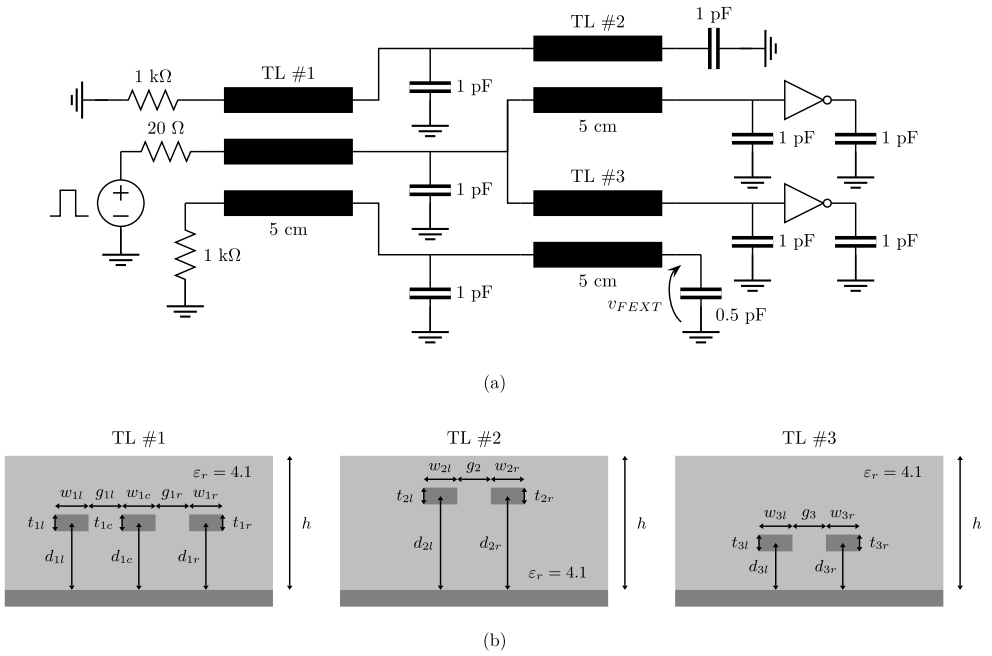

Fig. 12. Schematic of the network with coupled transmission lines (a) and cross-sections of the embedded microstrip lines (b).

Table 4
Nominal geometrical parameters for the coupled transmission line sections in the network of Fig. 12.

Parameter	w_1	w_2	w_3	d_1	d_2	d_3	g	h	t_1	t_2	t_3
Nominal value	150 μm	130 μm	170 μm	100 μm	140 μm	70 μm	150 μm	200 μm	30 μm	20 μm	40 μm

8.2. Network with coupled transmission lines

This second application example deals with the network of Fig. 12(a), consisting of three coupled transmission line sections interconnected by lumped elements [61]. The cross-sectional view of the embedded microstrip line sections is depicted in Fig. 12(b). In this high-dimensional test case, the uncertainty is provided by all the $d = 26$ geometrical parameters therein indicated, assumed to follow a Gaussian distribution with the nominal values in Table 4 and a relative standard deviation of 10%. The nominal values are the same for the geometry and position of each trace within a section, but the variation is assumed to be independent for each conductor. The voltage source on the left produces a 5-V pulse with a duration at half width of 1 ns and rise/fall times of 0.1 ns. The circuit is implemented in HSPICE. The target quantity of interest is the far-end crosstalk voltage denoted as v_{FEXT}, i.e., the coupling occurring between the inner transmission line traces, which are directly connected to the voltage source, and the adjacent line at the bottom of Fig. 12(a). References results are obtained based on a MC simulation with 5000 runs, which takes 1 h 21 min, whereas the SGM-based method used for the previous example is hardly applicable in the large dimensional setting of this example.

We first focus the analysis on a scalar quantity, namely the maximum crosstalk occurring over time. This is computed as the maximum absolute value of the transient response of $v_{\text{FETX}}(t)$ for a given parameter configuration and describes the worst-case coupling that can be expected over time for the given voltage source. We use the proposed PCE-GPR method to compute a third-order PCE. We consider an anisotropic kernel and we train the model in UQLab using ML estimation and the hybrid covariance matrix adaptation-evolution strategy (HCMAES) optimizer. We compare the performance against LAR, OMP, SP, and BCS methods as well as a standard GPR formulation with a squared-exponential kernel.

First of all, the performance is assessed based on the RMSE error between the model predictions and the reference MC samples. To this end, we consider three different sample sizes for training, namely $L = 40$, $L = 60$, and $L = 80$. To assess the dispersion of the performance across different training datasets, we perform 50 independent runs for each sample size with randomly generated configurations of the uncertain parameters. An instance of the aforementioned modeling methods is trained with each dataset. For all the PCE methods, two different truncations are considered: a total degree truncation and a hyperbolic truncation with $u = 0.7$. While the former exhibits 3654 expansion terms, the latter reduces the expansion to 404 terms only.

The results are illustrated by the boxplot in Fig. 13. The analysis allows us to make two interesting considerations. First, the two PCE-GPR results are virtually identical, thereby suggesting that high-order interactions can be safely neglected, as is the case for the hyperbolic truncation. We remind that these are two different approximations (truncations) of the same full kernel model.

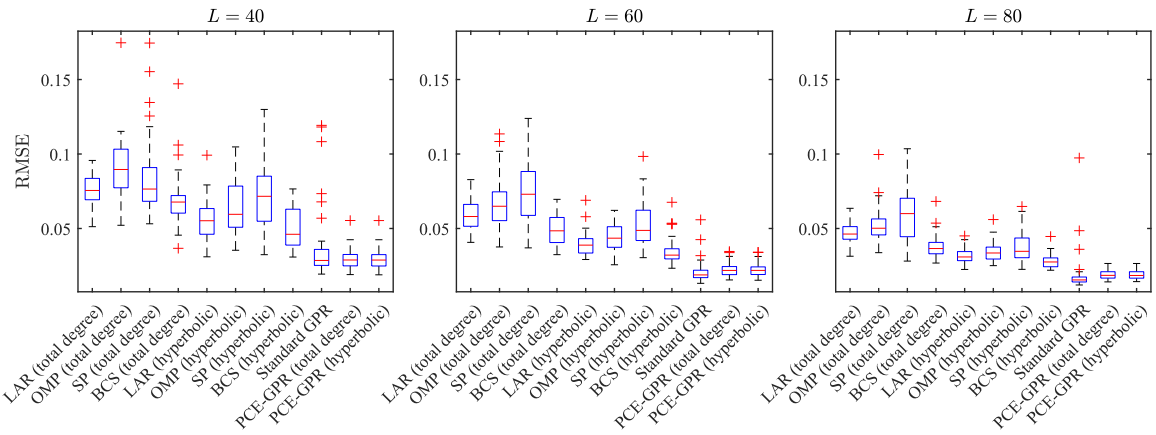

Fig. 13. Boxplot illustrating the RMSE achieved for the coupled transmission line circuit of Fig. 13 by the various methods over 50 independent runs and for three samples sizes.

Table 5
RMSE on the maximum crosstalk obtained with the various methods across 50 training datasets.

Method	Mean (V)			Standard deviation (V)			Maximum (V)		
	$L = 40$	$L = 60$	$L = 80$	$L = 40$	$L = 60$	$L = 80$	$L = 40$	$L = 60$	$L = 80$
LAR	0.0558	0.0390	0.0317	0.0139	0.0075	0.0049	0.0992	0.0689	0.0450
OMP	0.0649	0.0443	0.0344	0.0176	0.0093	0.0061	0.1048	0.0622	0.0560
SP	0.0716	0.0525	0.0379	0.0225	0.0147	0.0105	0.1299	0.0984	0.0648
BCS	0.0490	0.0343	0.0277	0.0130	0.0085	0.0044	0.0766	0.0676	0.0446
Standard GPR	0.0363	**0.0208**	**0.0184**	0.0229	0.0070	0.0128	0.1193	0.0559	0.0974
PCE-GPR	**0.0295**	0.0223	0.0190	**0.0064**	**0.0045**	**0.0031**	**0.0553**	**0.0341**	**0.0265**

On the contrary, the results of LAR, OMP, SP, and BCS for the total degree and the hyperbolic truncation differ significantly, the latter being more accurate. This is explained by the fact that, while the hyperbolic model virtually provides the same accuracy as noted above, the coefficients of a smaller basis are estimated by regression methods with a higher accuracy for a given sample size. It is important to remark, however, that the optimal basis cannot be easily determined a priori. In the case of PCE-GPR instead, the truncation can be adjusted a posteriori. Second, BCS yields the most accurate results among the state-of-the-art methods. However, PCE-GPR achieves a significantly lower median error and dispersion across all training set sizes.

The results are summarized in Table 5 in terms of mean, standard deviation, and maximum of the RMSE across the 50 training datasets. The results of the LAR, OMP, SP, and BCS methods refer to the hyperbolic truncation. It is noted that PCE-GPR provides comparable or better accuracy than the standard GPR. In particular, it exhibits a more consistent performance with a maximum RMSE that is less than half for the datasets with $L = 40$ and $L = 80$ samples and a standard deviation that is up to four times lower. This is appreciated also in Fig. 13, where PCE-GPR does not exhibit any large-error outliers as appearing instead in the standard GPR results. In this regard, we should point out that we do not expect PCE-GPR to necessarily provide a more accurate prediction compared to state-of-the-art GPR formulations based on standard choices of the prior covariance, as the main benefit of PCE-GPR lies in the analytical calculation of moments and sensitivity indices.

Next, we compute the Sobol' indices to gauge the impact of each geometrical parameter on the variability of the crosstalk. We generate reference results using a quasi-MC approach based on $N = 1000$ samples [85]. This requires overall the simulation of $N(d + 2) = 28\,000$ configurations, which took 7 h 10 min. The results are shown in Fig. 14 for a single realization of the three datasets of increasing size. The PCE-GPR predictions are compared against BCS and include the 95% confidence interval indicated by the error bars. In this regard, it is important to remark that the MC indices are an estimate based on relatively low number of samples (i.e., 1000 per dimension), and the results should be better compared to the "exact" indices, which are however unavailable. All methods compare well with the reference MC solution and identify the uncertain variables with indices $i = 8, 15, 19$ as the three most relevant parameters, which correspond to the substrate thickness h, the vertical position d_{3_r} of the right trace in transmission line #3, and the gap g_3 between the traces in the same transmission line section. This is reasonable, as crosstalk is primarily affected by the distance between the traces. Nevertheless, PCE-GPR obtains a more accurate prediction especially with the second and third training dataset, and the reference MC value always lies within its 95% confidence interval.

We now move to a full transient analysis that considers the entire time-domain response of v_{FEXT}, which is evaluated at $P = 1001$ time points with the SPICE simulator. We compare the PCE-GPR predictions based on the multi-output formulation of Section 6 to the results obtained with LAR and BCS in UQLab, which train a separate model for each column when a matrix dataset is provided. Fig. 15 shows, in the top panels, a subset of 100 samples from the MC simulation (thin gray lines). The mean of the MC samples is shown by the thick solid blue line and is compared to the LAR (dashed red line) and BCS result (dotted purple line). The shaded

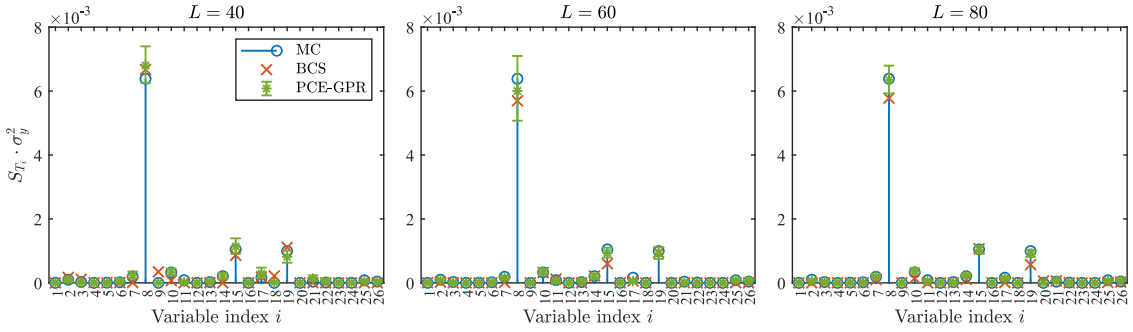

Fig. 14. Non-normalized Sobol' indices for the maximum crosstalk in the coupled transmission line network of Fig. 12, computed for three dataset sizes with MC (blue circles), BCS (red crosses), and GPR (green asterisks with 95% confidence interval).

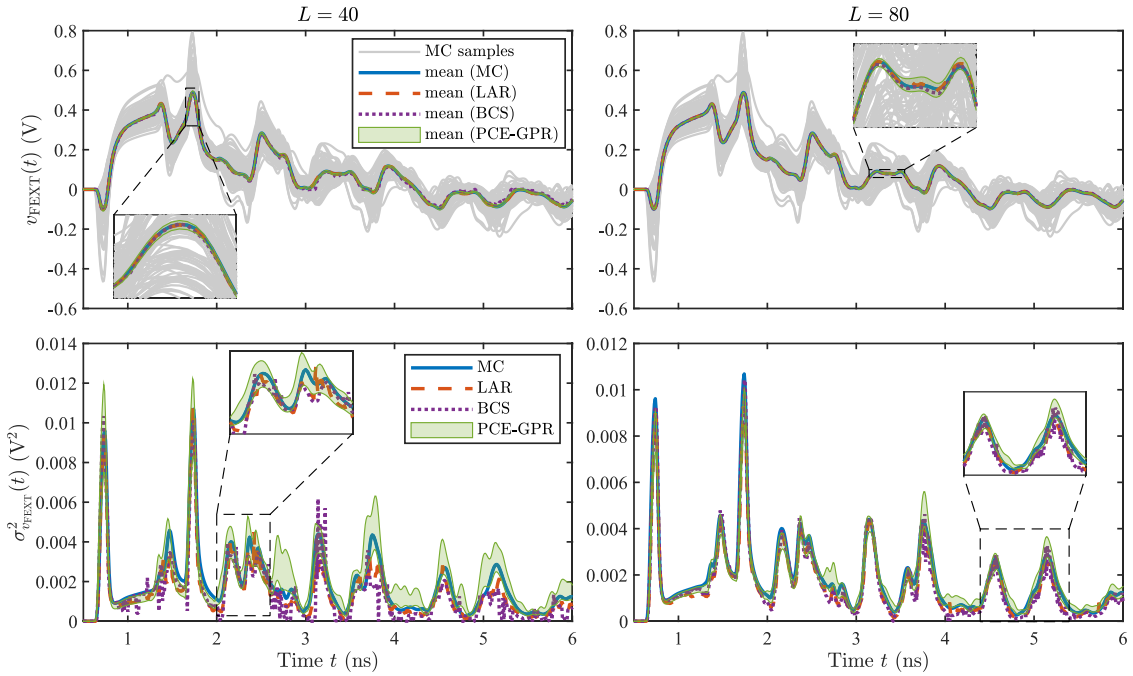

Fig. 15. UQ of the transient crosstalk in the network with coupled transmission lines. The top panels show a subset of MC samples (gray lines), the mean computed from the MC samples (thick blue line), with LAR (dashed red line) and BCS (dotted purple line), as well as the 95% confidence interval of the PCE-GPR prediction of the mean (shaded green area). The bottom panels show the variance of the MC samples (solid blue line), the variance computed with LAR (dashed red line) and BCS (dotted purple line), and the 2-sigma confidence interval of the PCE-GPR prediction of the variance (shaded green area). The left and the right panels refer to the smallest and the largest training dataset, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

green area shows the 2-sigma confidence interval of the PCE-GPR prediction of the mean. The insets allow appreciating the accuracy of PCE-GPR in predicting the mean transient crosstalk. The bottom panel of Fig. 15 shows, with the same curve identification, the prediction of the crosstalk variance. The difference between the 2-sigma and the 95% confidence intervals is found to be minimal, with the former being faster to estimate. Although the MC variance is just an estimate with a finite precision, the PCE-GPR prediction compares well, with the confidence interval providing a good indication of the prediction uncertainty. Conversely, the LAR and BCS results are less accurate and less smooth, as seen in the insets, with the latter being particularly irregular and spiky over time.

Finally, Table 6 provides an estimate of the actual confidence level of the PCE-GPR predictions. For this analysis, we take the MC estimates as the reference and we count the rate at which they fall within the 2-sigma interval of the PCE-GPR prediction over 50 independent runs with randomized training datasets. Since the rate may differ over the time points, we provide the result in terms of average, median, and mode. We exclude from the analysis the response before $t = 6.2$ ns, as crosstalk is virtually zero and small numerical errors may alter the calculation. Moreover, we consider both (24) and (59) to calculate the total variance of the PCE-GPR model.

Table 6
Actual confidence level of the PCE-GPR predictions in terms of average, median, and mode over time.

Variance model		$L = 40$		$L = 60$		$L = 80$	
		$\mu_{v_{\text{FEXT}}}$	$\sigma^2_{v_{\text{FEXT}}}$	$\mu_{v_{\text{FEXT}}}$	$\sigma^2_{v_{\text{FEXT}}}$	$\mu_{v_{\text{FEXT}}}$	$\sigma^2_{v_{\text{FEXT}}}$
Eq. (24)	Average	90%	83%	93%	83%	95%	80%
	Median	90%	84%	94%	84%	96%	80%
	Mode	88%	86%	98%	90%	96%	78%
Eq. (59)	Average	94%	81%	96%	74%	95%	78%
	Median	94%	82%	96%	76%	96%	80%
	Mode	94%	76%	98%	90%	96%	74%

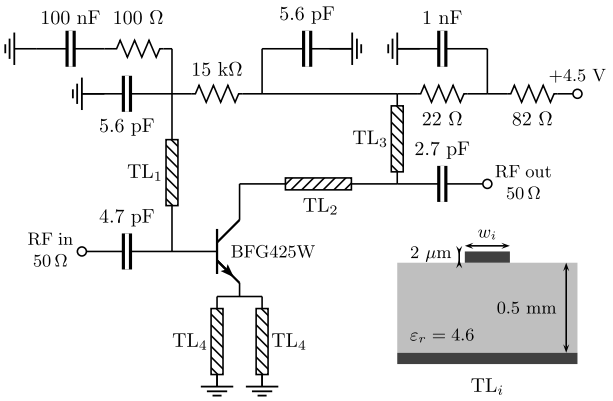

Fig. 16. Schematic of the low-noise amplifier considered in the third application example.

The figures in Table 6 show that the actual confidence level is very close to the theoretical 95% for the predicted mean, and with both estimates of the total variance. Conversely, the confidence level of the predicted variance is slightly lower, with (24) providing a more conservative result. However, we should keep in mind that the following two factors contribute in making the actual confidence deviate from the reference 95% value: (1) the calculation is based on the MC estimate of the moments rather than on their actual value, and the difference may be non-negligible especially for the variance; (2) the 2-sigma interval does not precisely correspond to a 95% confidence level for the variance due to its non-Gaussian predictive distribution, though it does provide a reasonable approximation. Overall, this analysis demonstrates with a practical example that the 2-sigma interval, which is more straightforward to compute, effectively indicates the prediction uncertainty associated with the limited quantity and specific selection of training samples. Additionally, it confirms that the model in Eq. (24) offers a valid alternative for calculating the total variance.

8.3. Low-noise amplifier

The last proposed example concerns the low-noise amplifier whose schematic is depicted in Fig. 16. This is a circuit designed to amplify an input radio-frequency signal at 2 GHz. The variability is provided by $d = 25$ uncertain parameters, including the resistors and the capacitors in Fig. 16, the width of the four microstrip sections, and 11 parameters within the BFG425 W bipolar junction transistors. These parameters are assumed to follow a uniform distribution within ±20% from the nominal value. We refer to [61] for additional details on this test case.

Just like in the previous example, we first focus the analysis on a scalar quantity, i.e., the amplifier gain at the designed operating frequency of 2 GHz. This corresponds to the magnitude of the scattering parameter S_{21}, expressed in decibels, which is obtained with HSPICE via a small-signal frequency-domain simulation around the bias point. For this purpose, we train a single model for the gain directly. For the PCE-based methods, we compute a third-order model with either a total degree or a hyperbolic truncation with $u = 0.7$. The two models feature 3276 and 376 expansion terms, respectively. The performance is also compared against a standard GPR model with an anisotropic squared-exponential kernel. We consider once again three training datasets of increasing size, with $L = 30$, $L = 60$, and $L = 90$ samples. Reference results are generated with a MC analysis based on 5000 random configurations of the uncertain parameters, which takes 1 h 3 min.

Fig. 17 illustrates the RMSE obtained by the various methods over 50 independent runs for each of the three dataset sizes. Similarly to the previous example, the PCE-GPR models with total degree and hyperbolic truncation exhibits similar performance, meaning that higher-order interactions can be safely neglected also in this example. Conversely, LAR, OMP, SP, and BCS all yield a more accurate model when the basis functions are pruned a priori to retain only the most relevant terms. Nevertheless, the error remains significantly larger compared to PCE-GPR, which achieves a median RMSE that is 2.6× to 1.4× lower than the best alternative

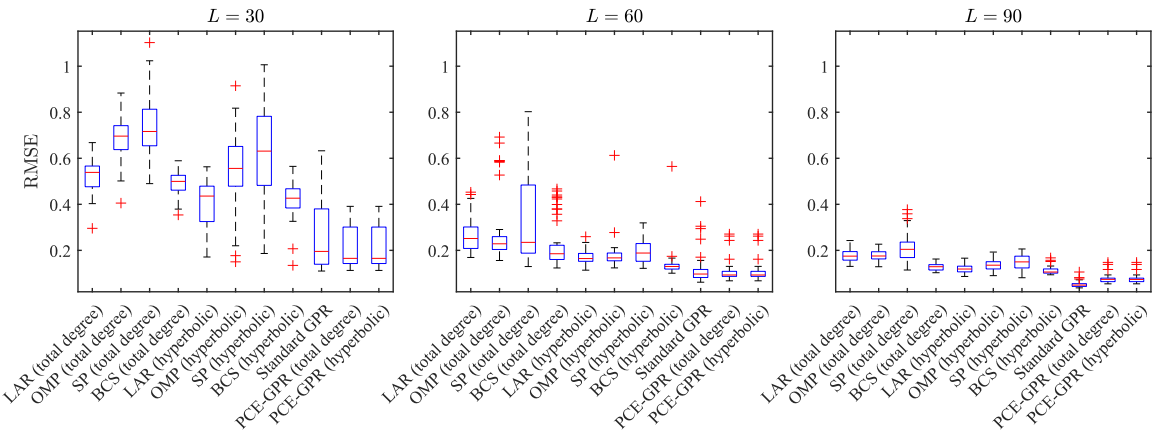

Fig. 17. Boxplot illustrating the RMSE achieved for the low-noise amplifier of Fig. 16 by the various methods over 50 independent runs and for three samples sizes.

Table 7
RMSE on the amplifier gain obtained with the various methods across 50 training datasets.

Method	Mean (dB)			Standard deviation (dB)			Maximum (dB)		
	$L = 30$	$L = 60$	$L = 90$	$L = 30$	$L = 60$	$L = 90$	$L = 30$	$L = 60$	$L = 90$
LAR	0.4002	0.1697	0.1198	0.1091	0.0298	0.0174	0.5630	0.2596	0.1661
OMP	0.5424	0.1794	0.1371	0.1690	0.0679	0.0237	0.9148	0.6127	0.1927
SP	0.6194	0.1959	0.1475	0.1955	0.0522	0.0320	1.0062	0.3194	0.2061
BCS	0.4221	0.1387	0.1122	0.0760	0.0634	0.0159	0.5649	0.5647	0.1674
Standard GPR	0.2536	0.1145	**0.0511**	0.1468	0.0662	**0.0121**	0.6326	0.4121	**0.1060**
PCE-GPR	**0.2046**	**0.1068**	0.0754	**0.0871**	**0.0421**	0.0179	**0.3912**	**0.2709**	0.1478

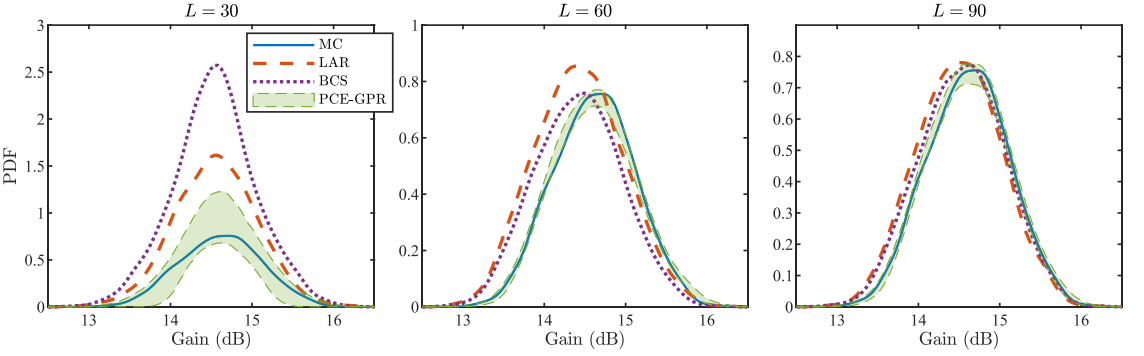

Fig. 18. PDF of the amplifier gain computed with MC (solid blue line), LAR (dashed red line), and BCS (dotted purple line) in comparison with the 95% confidence interval of the PCE-GPR prediction (shaded green area), for three training datasets of increasing size. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

PCE method. The results for the RMSE are summarized in Table 7. The metrics indicate that PCE-GPR performs similarly to standard GPR overall but outperforms it on the smaller training datasets.

Based on the previous analysis, we restrict further investigations to PCE models with hyperbolic truncation and in comparison with LAR and BCS, which turned out to provide the best performance among the state-of-the-art approaches. Fig. 18 shows the PDF of the amplifier gain. The distribution of the MC samples (solid blue line) is compared to the predictions obtained with LAR (dashed red line), BCS (dotted purple line), and PCE-GPR (shaded green area, representing the 95% confidence interval) with the three datasets of increasing size. While LAR and BCS fail in obtaining an accurate model even with 90 training samples, the PCE-GPR prediction is already in very good agreement with the reference with 30 training samples only.

Next, we perform a sensitivity analysis of the gain. Fig. 19 shows the non-normalized Sobol' indices and compares the results of BCS (red crosses) and PCE-GPR (green asterisks with 95% confidence bars) against the MC reference, computed based on 27 000 configurations of the input parameters (blue circles). The remarkable result is that PCE-GPR allows to correctly identify the most relevant parameters, with the MC reference falling almost always within its confidence interval. It should be noted that propagating

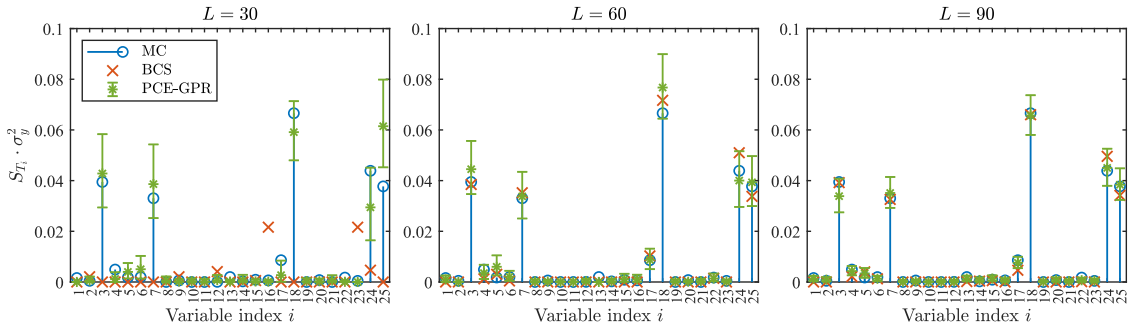

Fig. 19. Non-normalized Sobol' indices for the gain of the low-noise amplifier of Fig. 16, computed for three dataset sizes with MC (blue circles), LAR (red diamonds), and GPR (green asterisks with 95% confidence interval).

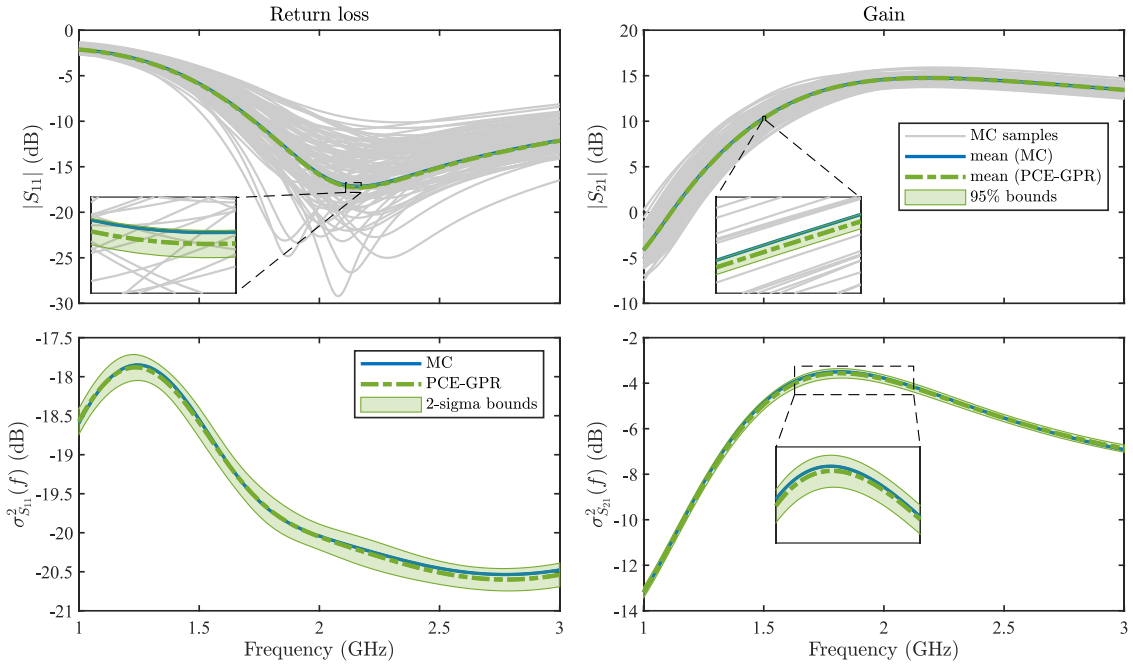

Fig. 20. Frequency-dependent scattering parameters S_{11} (left) and S_{21} (right). The top panels show the magnitude of a subset of MC responses (thin solid gray lines) as well as the magnitude of the average response obtained from the MC samples (thick solid blue line) and with the PCE-GPR model (dash-dotted green lines and shaded green area, representing the 2-sigma confidence interval). The bottom panels show, with the same color scheme, the variance of the MC samples and the corresponding PCE-GPR prediction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the prediction uncertainty to the Sobol' indices with the MC-based approach in [60] would require to handle covariance matrices of prohibitive size, whereas it is straightforward with the PCE-GPR method.

Five parameters turn out to be significantly more relevant than others, with indices $i = 18, 24, 25, 3, 7$. They correspond to one of the two 5.6-pF capacitors, to the widths of microstrip lines TL_3 and TL_4, to the collector junction capacitance, and to a parasitic inductance of the transistor package. The same parameters are identified also by PCE-GPR with the smallest dataset already. Conversely, BCS fails to identify parameters #18, #3 and #7 as relevant when using only 30 training samples.

Finally, we consider the full frequency-domain response of the amplifier in the 1–3 GHz range. To this end, we simulate both the complex-valued scattering parameters S_{11} (input reflection) and S_{21} (gain) at 201 equally-spaced frequency points. However, accounting for complex data directly is non-trivial, since the baseline GPR framework assumes real-valued observations. A rigorous approach would require to introduce a pseudo-covariance between the real and imaginary part [95], which in turn would require to suitably extend and adapt the PCE-GPR framework. On the other hand, it was shown in [96] that using a pseudo-covariance provides marginal benefits in the modeling of frequency-domain electrical responses compared to modeling the real and imaginary part separately. Hence, for the sake of mere illustration, we shall adopt this second strategy, for which the outlined framework applies almost seamlessly. To this end, we recast the complex-valued dataset $\tilde{\boldsymbol{Y}} \in \mathbb{C}^{L \times P}$ as an expanded real-valued dataset $\boldsymbol{Y} = \left(\mathfrak{Re}\left(\tilde{\boldsymbol{Y}}\right), \mathfrak{Im}\left(\tilde{\boldsymbol{Y}}\right)\right) \in \mathbb{R}^{L \times 2P}$, which is handled with the multi-output formulation of Section 6. Hence, the total size of the

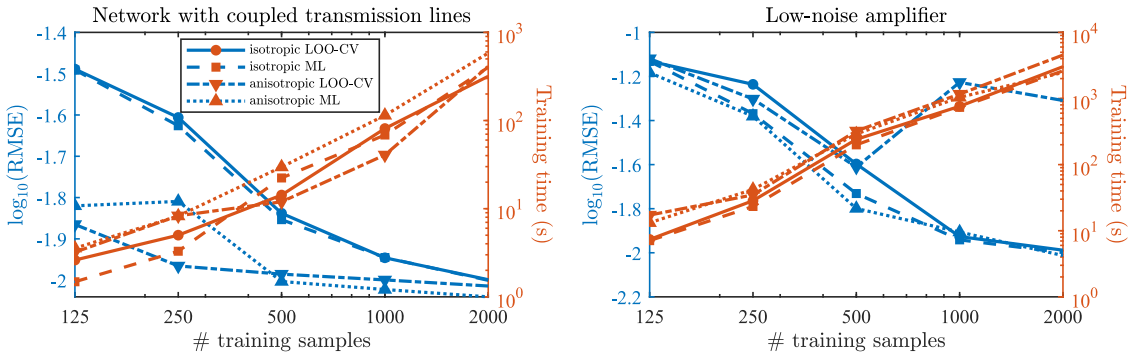

Fig. 21. RMSE (blue lines) and training time (red lines) for large training datasets in the network with coupled transmission lines and low-noise amplifier test cases. The performance of various UQLab implementations of PCE-GPR with isotropic or anisotropic kernel and LOO-CV or ML estimation of the hyperparameters are compared. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

real-valued multi-output training dataset is $P = 804$ (201 frequency points for both the real and imaginary part of each scattering parameter). The resulting PCE-GPR model will separately provide the PCE coefficients for the real and imaginary of each output component.

At this point, it is worth recalling some properties concerning a complex-valued random variable $Y = U + jV$. The expected value and variance of Y are computed as

$$\mu_y = \mathrm{E}(Y) = \mathrm{E}(U) + j\mathrm{E}(V) = \mu_u + j\mu_v \in \mathbb{C} \tag{65}$$

and

$$\sigma_y^2 = \mathrm{Var}(Y) = \mathrm{Var}(U) + \mathrm{Var}(V) = \sigma_u^2 + \sigma_v^2 \in \mathbb{R}, \tag{66}$$

respectively [97]. The PCE-GPR model allows computing μ_u, μ_v, σ_u^2, and σ_v^2 probabilistically, in terms of both expected value and variance. Therefore, by applying again (65) and (66), the expected value and the variance of μ_y and σ_y^2 are obtained as

$$\mathrm{E}(\mu_y) = \mathrm{E}(\mu_u) + j\mathrm{E}(\mu_v) \tag{67}$$

$$\mathrm{Var}(\mu_y) = \mathrm{Var}(\mu_u) + \mathrm{Var}(\mu_v) \tag{68}$$

$$\mathrm{E}(\sigma_y^2) = \mathrm{E}(\sigma_u^2) + \mathrm{E}(\sigma_v^2) \tag{69}$$

$$\mathrm{Var}(\sigma_y^2) = \mathrm{Var}(\sigma_u^2) + \mathrm{Var}(\sigma_v^2). \tag{70}$$

The results are shown in Fig. 20 for a training dataset with 90 samples. The thin gray lines in the top panels are a subset of MC responses and show the large variability of the scattering parameters, especially the return loss in the neighborhood of the operating frequency. Nevertheless, the PCE-GPR predictions (dash-dotted green lines) and their respective 2-sigma confidence intervals (shaded green areas) compare very well with the reference MC results (thick solid blue lines), as can be further appreciated in the insets. It should be noted that the exact predictive distribution of the variance, being the sum of two generalized chi-square distributions, is difficult to derive in closed form.

9. Training accuracy and efficiency

In this section, we investigate the performance of PCE-GPR in terms of both accuracy and training cost when the number of training samples is increased. In this regard, it is important to keep in mind that, when targeting the UQ of complex designs, it is crucial to limit the number of training samples to be competitive against a classical MC simulation. Hence, the designer is usually constrained to a small-data scenario, with a number of training samples in the order of a few hundreds at most. Specifically, we focus on the single-output test cases of Sections 8.2 and 8.3 and we consider the UQLab implementation with isotropic or anisotropic kernels and using LOO-CV or ML for the estimation of the hyperparameters.

The results are provided in Fig. 21 for the network with coupled transmission lines of Section 8.2 (left panel) and the low-noise amplifier of Section 8.3 (right panel). The plots show the performance in terms of RMSE (left axes, blue lines) and training cost (right axes, red lines) for the different implementations when the number of training samples is doubled from 125 to 2000. In terms of accuracy, the implementation with an anisotropic kernel and ML estimation (dotted lines) overall achieves the best results.

As far as the training time is concerned, the computational cost increases significantly with the number of samples, a typical behavior of GPR. For both test cases, the model is trained within a few seconds up to 250 training samples, regardless of the specific implementation. For the network with coupled transmission lines, the training time remains below 1 minute up to 500 samples with all implementations, and up to 1000 samples for the anisotropic LOO-CV implementation. However, the training time is comparable

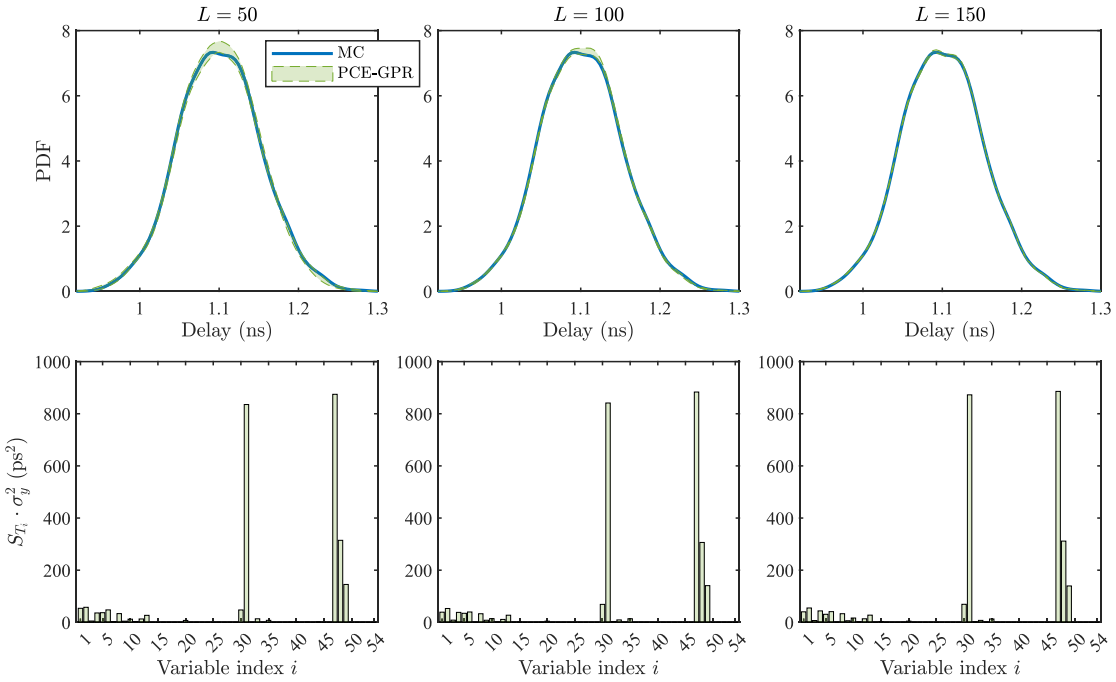

Fig. 22. UQ of the interconnect delay obtained with PCE-GPR for three training datasets of increasing size. Top panels: 95% confidence interval of the PDF prediction (shaded green area) in comparison with the distribution of the MC samples (blue lines). Bottom panels: non-normalized Sobol' indices.

for all implementations, although anisotropic kernels require to solve a minimization problem in d dimensions (or $d + 1$, if noise is considered), rather than in one (or two) dimensions. The situation differs for the low-noise amplifier test case, for which the training time remains below 1 minute only up to 250 training samples. A detailed analysis reveals that, for problems uniform variability, the calculation of the kernel matrix for a large number of samples incurs an extra cost due to the evaluation of the elliptic integral in (32). The post-processing calculation of the PCE coefficients and the prediction of the test samples for the calculation of the RMSE, which are common to all implementations, are instead negligible (within 1 s) even for the largest training dataset.

All in all, the implementation with anisotropic kernels and ML estimation exhibits a good trade-off between accuracy and training cost. However, all implementations obtain accurate results with a limited training cost when the number of training samples is below a few hundreds, which is the case of interest for the intended applications to UQ of complex designs. For larger datasets, the proposed method could benefit from established techniques for handling large-scale data (e.g., [98]).

10. High-dimensional example

Finally, we apply the PCE-GPR method to a high-dimensional variant of the coupled line network example of Section 8.2. The additional uncertainty is provided by all the 29 lumped elements (resistors, inductors, and capacitors), by the microstrip trace width, thickness, and length of all transmission lines, and by their conductivity. The total number of uncertain parameters is thus $d = 54$. A Gaussian distribution with a 10% relative standard deviation from the nominal values in Fig. 12 is assumed. For the trace conductivity, a nominal value of 58 MS/m (copper) is considered.

Given that interconnect delay is a critical aspect of integrated circuits, we focus our analysis on the delay of the voltage v_{TX}, defined as the time at which the voltage crosses 100 mV. We use the PCE-GPR method to compute a third-order expansion based on three datasets with $L = 50$, $L = 100$, and $L = 150$ samples. Following the conclusions of the previous section, we keep adopting the UQLab implementation with an anisotropic kernel and ML estimation of the hyperparameters by means of the HCMAES optimizer.

Fig. 22 shows, in the top panels, the 95% confidence bounds of the PDF of the interconnect delay obtained with PCE-GPR (shaded green area) in comparison with the distribution of 5000 MC samples (blue lines) for the three training datasets. The confidence bounds provide once again a good indication of the prediction accuracy, with the reference MC distribution always lying within them. The bottom panels provide instead the Sobol' indices describing the individual contribution of each uncertain variable on the delay. It is found that the most critical parameters are the length of the 5-cm transmission line and the relative permittivity of the substrate, followed by the length of the 3-cm and 2-cm transmission line sections on the same path, and by the height of the substrate. These parameters are already identified with the smallest training dataset.

Next, we assess the performance over 50 independent realizations of the training datasets. We also compare the results against the state-of-the-art PCE methods, for which we let the hyperbolic truncation factor to be adaptively selected between $u = 0.5$ and $u = 0.7$, and a standard GPR formulation with an anisotropic squared-exponential kernel. Table 8 provides the results in terms

Table 8
Performance comparison between PCE-GPR and the state-of-the-art methods across 50 training datasets.

Method	Median RMSE (ps)			Dispersion (ps)		
	$L = 50$	$L = 100$	$L = 150$	$L = 50$	$L = 100$	$L = 150$
LAR	22.6572	5.0996	3.5962	9.0208	1.3750	0.3887
OMP	32.5400	8.4972	3.9986	5.9191	9.5288	1.0713
SP	25.1114	4.6808	3.4860	10.7082	1.0876	0.9801
BCS	17.4715	3.9300	3.1014	5.1374	0.4340	0.5113
Standard GPR	19.8310	11.9584	8.6832	3.1943	1.7419	0.7464
PCE-GPR	**5.6863**	**2.3639**	**1.6437**	**2.8516**	**0.3196**	**0.1416**

Table 9
Predicted mean and variance of the interconnect delay and training time for PCE-GPR across various training datasets.

Method	Observations	μ_y (ns)	σ_y^2 (ns^2)	Training time (s)
MC	$N = 5000$	1.1012	2.6294×10^{-3}	–
PCE-GPR (2-sigma bounds)	$L = 50$	1.1006 [1.1000, 1.1012]	2.5800×10^{-3} $[2.4681, 2.6919] \times 10^{-3}$	3.7
	$L = 100$	1.1012 [1.1008, 1.1015]	2.5786×10^{-3} $[2.5349, 2.6222] \times 10^{-3}$	5.7
	$L = 150$	1.1010 [1.1008, 1.1011]	2.6236×10^{-3} $[2.6010, 2.6461] \times 10^{-3}$	11.0

of median RMSE and dispersion, which we define as the difference between the 75% and 25% quantiles. PCE-GPR is found to outperform all the alternative methods in both metrics. Lastly, Table 9 provides the results regarding the estimation of the mean and the variance of the delay with one instance of the training datasets, as well as the training time. The figures further confirm that PCE-GPR achieves very accurate results with a reasonable training time and that the associated confidence bounds are conservative.

11. Conclusions

This paper presented a novel UQ method that effectively combines PCE and GPR. The proposed approach leverages special implicit kernels that build upon Hermite or Legendre polynomials and inherits the advantages of both techniques, namely analytical statistical and sensitivity information from PCE and training efficiency, scalability to higher dimensions, and embedded confidence information from GPR. Therefore, PCE-GPR provides accurate statistical information with a limited amount of training data and the inclusion of confidence levels.

Closed-form formulas were derived to convert from GPR to PCE model coefficients, and for the expected value and covariance of the classical statistical information provided by the PCE. One remarkable property is that the PCE coefficients can be computed individually and up to an arbitrary order, which does not need to be defined upfront and without the need to form the entire set of basis functions at once. A simple multi-output formulation was also discussed. The standard single-output formulation was demonstrated to be compatible with the general-purpose Kriging module in UQLab, which offers several established methods for hyperparameter estimation and optimization.

The method was first illustrated based on two analytical benchmark functions. It was then applied to three test cases in electrical engineering, namely a network with single microstrip transmission lines, a network with coupled microstrip lines, and a low-noise amplifier. For all test cases, the advocated PCE-GPR was shown to achieve accurate results and outperform state-of-the-art PCE-based methods such as LAR, OMP, SP, and BCS, in terms of accuracy and training data required. Repeated-run analyses over multiple realizations of the training datasets showed that PCE-GPR consistently achieved a lower median error and a lower dispersion compared to the abovementioned methods. The performance was instead comparable to classical GPR, which however does not provide statistical information in closed form. Moreover, the analyses also confirmed that the calculated posterior covariance provides an accurate indication of the actual prediction confidence related to the limited availability of data. The excellent performance of PCE-GPR was further confirmed for a modified version of the network with single transmission lines, in which the number of uncertain parameters was increased to 54.

The multi-output formulation was shown to be accurate and viable, although its accuracy was found to be lower than an alternative and common approach based on PCA. The training efficiency decreased with larger datasets, above a few hundred samples. This is a common limitation of GPR-based methods. However, in the proposed application examples, this did not impact performance, as PCE-GPR achieved accurate results with a small sample size, aligning well with the primary goal of UQ in complex systems.

Based on the results achieved, the proposed method shows potential to become a highly effective technique for data-driven PCE. At the moment, the formulation is limited by the available kernels to Gaussian and uniform random variables only. Future works will address the extension to other distribution types, the investigation of alternative strategies for multiple and possibly complex-valued outputs, the inclusion of the PCE-GPR in optimization frameworks, and the scalability to larger datasets.

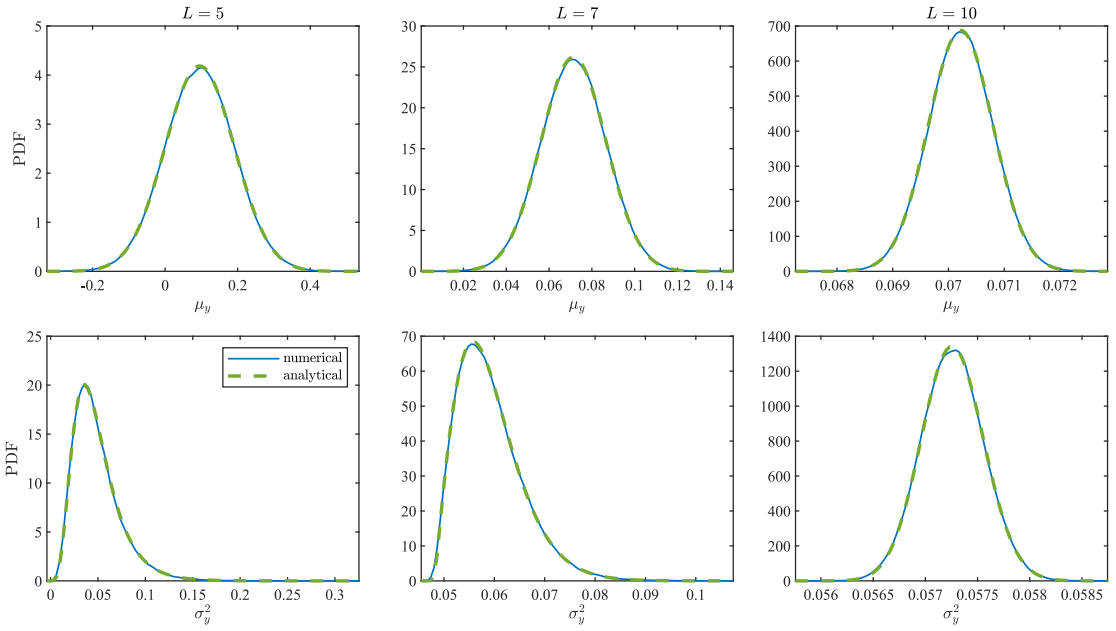

Fig. A.23. Predictive distribution of μ_y and σ_y^2. Solid blue line: numerical estimate; dashed green line: analytical model.

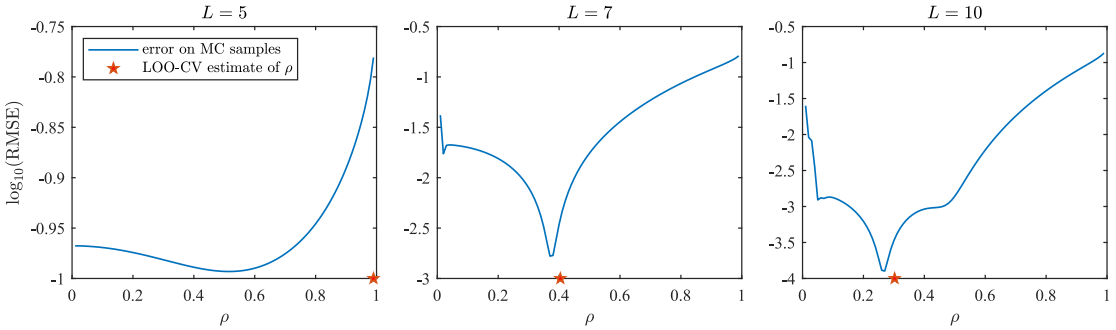

Fig. A.24. RMSE on the prediction of the MC samples for different values of the hyperparameter ρ (blue line). The red star indicates the hyperparameter value selected by the LOO-CV scheme.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Further results for the one-dimensional illustrative function

Fig. A.23 shows the posterior distribution of the predicted mean and variance of (60). The analytical model obtained from the mean vector and covariance matrix of the PCE coefficients, i.e., a Gaussian distribution for μ_y and a generalized chi-square distribution for σ_y^2, is shown by the dashed green line. The reference distribution (solid blue line) is calculated numerically based on a large number of posterior trajectories. The trajectories are then integrated according to (10) to compute the PCE coefficients, from which the corresponding samples of the predicted mean and variance are obtained. The results are in excellent agreement and further validate the posterior model of the PCE coefficients.

We now investigate the impact of the hyperparameter ρ. To this end, we sweep it in the interval $(0, 1)$ rather than optimizing its value, and we assess the resulting RMSE error on the prediction of the MC samples. The results are shown by the solid blue line in Fig. A.24. For reference, the red star indicates the location of the hyperparameter value selected through the LOO-CV scheme. This value is remarkably close to the actual optimum except for the dataset with $L = 5$ samples. However, the variation of the RMSE error in the hyperparameter range is found to be rather marginal in that case. Moreover, it is reasonable that the LOO-CV estimate become more and more accurate when the number of training samples is increased.

Table A.10
Estimate of the actual confidence level of PCE-GPR predictions.

Variance model	Training samples	μ_y	σ_y^2	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
Eq. (24)	$L = 5$	100%	93%	100%	100%	100%	60%	84%	100%	100%	100%	100%	100%	100%
	$L = 7$	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	$L = 10$	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Eq. (26)	$L = 5$	100%	96%	100%	100%	100%	70%	94%	100%	100%	100%	100%	100%	100%
	$L = 7$	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	$L = 10$	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Finally, we assess the actual confidence level of the PCE-GPR predictions based on the results of the repeated-run analysis. Since the posterior variance is a measure of the uncertainty related to the specific choice of the training samples, the actual confidence level is estimated by counting the rate at which the reference result actually falls within the 95% confidence interval of each prediction. Table A.10 reports the result over the 100 runs for all the predicted quantities of interest, i.e., the individual PCE coefficients as well as the mean and the variance (recall that the mean and the coefficient c_0 actually coincide). In this analysis, we compare the two alternative calculations of the total GPR variance σ_{tot}^2, i.e., the ML estimate (24) and the LOO-CV estimate (26). It is observed that the actual confidence often exceeds the theoretical value of 95% up to a 100% rate, thereby indicating a slight overestimation of the prediction uncertainty. There is an exception for the coefficient c_3 and the training dataset of size $L = 5$. It is also noted that (26) provides a more accurate estimate in that case. In general, it is reasonable that the estimation of the prediction uncertainty improves by increasing the number of data samples.

Data availability

Data will be made available on request.

References

[1] International Roadmap for Devices and Systems™, More Moore, 2023, URL: https://irds.ieee.org/images/files/pdf/2023/2023IRDS_MM.pdf.
[2] International Roadmap for Devices and Systems™, Yield Enhancement, 2023, URL: https://irds.ieee.org/images/files/pdf/2023/2023IRDS_YE.pdf.
[3] R. Spence, R.S. Soin, Tolerance Design of Electronic Circuits, World Scientific, 1997.
[4] A. Kaintura, T. Dhaene, D. Spina, Review of polynomial chaos-based methods for uncertainty quantification in modern integrated circuits, Electronics 7 (3) (2018) 30, http://dx.doi.org/10.3390/electronics7030030.
[5] H.R. Fairbanks, L. Jofre, G. Geraci, G. Iaccarino, A. Doostan, Bi-fidelity approximation for uncertainty quantification and sensitivity analysis of irradiated particle-laden turbulence, J. Comput. Phys. 402 (2020) 108996, http://dx.doi.org/10.1016/j.jcp.2019.108996.
[6] J. Song, P. Wei, M.A. Valdebenito, M. Faes, M. Beer, Data-driven and active learning of variance-based sensitivity indices with Bayesian probabilistic integration, Mech. Syst. Signal Process. 163 (2022) 108106, http://dx.doi.org/10.1016/j.ymssp.2021.108106.
[7] D. Lee, E. Lavichant, B. Kramer, Global sensitivity analysis with limited data via sparsity-promoting D-MORPH regression: Application to char combustion, J. Comput. Phys. 511 (2024) 113116, http://dx.doi.org/10.1016/j.jcp.2024.113116.
[8] B. Bhattacharyya, Global sensitivity analysis: A Bayesian learning based polynomial chaos approach, J. Comput. Phys. 415 (2020) 109539, http://dx.doi.org/10.1016/j.jcp.2020.109539.
[9] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644, http://dx.doi.org/10.1137/S1064827501387826.
[10] D. Xiu, Fast numerical methods for stochastic computations: A review, Commun. Comput. Phys. 5 (2–4) (2009) 242–272.
[11] T.J. Sullivan, Introduction to Uncertainty Quantification, vol. 63, Springer, Cham, Switzerland, 2015.
[12] R.G. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, New York, NY, USA, 1991.
[13] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Commun. Comput. Phys. 2 (2) (2007) 293–309.
[14] S. Zein, B. Colson, F. Glineur, An efficient sampling method for regression-based polynomial chaos expansion, Commun. Comput. Phys. 13 (4) (2013) 1173–1188, http://dx.doi.org/10.4208/cicp.020911.200412a.
[15] M. Hadigol, A. Doostan, Least squares polynomial chaos expansion: A review of sampling strategies, Comput. Methods Appl. Mech. Engrg. 332 (2018) 382–407, http://dx.doi.org/10.1016/j.cma.2017.12.019.
[16] N. Lüthen, S. Marelli, B. Sudret, Sparse polynomial chaos expansions: Literature survey and benchmark, SIAM/ASA J. Uncertain. Quantif. 9 (2) (2021) 593–649, http://dx.doi.org/10.1137/20M1315774.
[17] G. Blatman, B. Sudret, Adaptive sparse polynomial chaos expansion based on least angle regression, J. Comput. Phys. 230 (6) (2011) 2345–2367, http://dx.doi.org/10.1016/j.jcp.2010.12.021.
[18] A. O'Hagan, et al., Polynomial chaos: A tutorial and critique from a statistician's perspective, SIAM/ASA J. Uncertain. Quantif. 20 (2013) 1–20.
[19] S. Roy, P. Manfredi, D. Vande Ginste, S. Guglani, Z. Zhang, R. Trinchero, F. Canavero, F. Feng, J. Zhang, Q.-J. Zhang, S. Lallèchére, D. Spina, T. Dhaene, F. Grassi, in: S. Roy (Ed.), Uncertainty Quantification of Electromagnetic Devices, Circuits, and Systems, IET, Stevenage, United Kingdom, 2021.
[20] H.N. Najm, Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics, Annu. Rev. Fluid Mech. 41 (1) (2009) 35–52, http://dx.doi.org/10.1146/annurev.fluid.010908.165248.
[21] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, Probab. Eng. Mech. 25 (2) (2010) 183–197, http://dx.doi.org/10.1016/j.probengmech.2009.10.003.
[22] S. Chakraborty, R. Chowdhury, An efficient algorithm for building locally refined hp–adaptive H-PCFE: Application to uncertainty quantification, J. Comput. Phys. 351 (2017) 59–79, http://dx.doi.org/10.1016/j.jcp.2017.09.024.
[23] Y. Zhou, Z. Lu, K. Cheng, Adaboost-based ensemble of polynomial chaos expansion with adaptive sampling, Comput. Methods Appl. Mech. Engrg. 388 (2022) 114238, http://dx.doi.org/10.1016/j.cma.2021.114238.
[24] N. Alemazkoor, A. Louhghalam, M. Tootkaboni, A multi-fidelity polynomial chaos-greedy Kaczmarz approach for resource-efficient uncertainty quantification on limited budget, Comput. Methods Appl. Mech. Engrg. 389 (2022) 114290, http://dx.doi.org/10.1016/j.cma.2021.114290.

[25] L. Cao, J. Liu, C. Jiang, G. Liu, Optimal sparse polynomial chaos expansion for arbitrary probability distribution and its application on global sensitivity analysis, Comput. Methods Appl. Mech. Engrg. 399 (2022) 115368, http://dx.doi.org/10.1016/j.cma.2022.115368.
[26] K. Kontolati, D. Loukrezis, D.G. Giovanis, L. Vandanapu, M.D. Shields, A survey of unsupervised learning methods for high-dimensional uncertainty quantification in black-box-type problems, J. Comput. Phys. 464 (2022) 111313, http://dx.doi.org/10.1016/j.jcp.2022.111313.
[27] X. Zeng, R. Ghanem, Projection pursuit adaptation on polynomial chaos expansions, Comput. Methods Appl. Mech. Engrg. 405 (2023) 115845, http://dx.doi.org/10.1016/j.cma.2022.115845.
[28] P.-C. Bürkner, I. Kröker, S. Oladyshkin, W. Nowak, A fully Bayesian sparse polynomial chaos expansion approach with joint priors on the coefficients and global selection of terms, J. Comput. Phys. 488 (2023) 112210, http://dx.doi.org/10.1016/j.jcp.2023.112210.
[29] K.D. Kantarakias, G. Papadakis, Sensitivity-enhanced generalized polynomial chaos for efficient uncertainty quantification, J. Comput. Phys. 491 (2023) 112377, http://dx.doi.org/10.1016/j.jcp.2023.112377.
[30] Y. Li, Y. Luo, Z. Zhong, An active sparse polynomial chaos expansion approach based on sequential relevance vector machine, Comput. Methods Appl. Mech. Engrg. 418 (2024) 116554, http://dx.doi.org/10.1016/j.cma.2023.116554.
[31] M. Thapa, S.B. Mulani, A. Paudel, S. Gupta, R.W. Walters, Classifier-based adaptive polynomial chaos expansion for high-dimensional uncertainty quantification, Comput. Methods Appl. Mech. Engrg. 422 (2024) 116829, http://dx.doi.org/10.1016/j.cma.2024.116829.
[32] N. Pepper, F. Montomoli, K. Kantarakias, SeAr PC: Sensitivity enhanced arbitrary polynomial chaos, Comput. Methods Appl. Mech. Engrg. 431 (2024) 117269, http://dx.doi.org/10.1016/j.cma.2024.117269.
[33] H. Sharma, L. Novák, M. Shields, Physics-constrained polynomial chaos expansion for scientific machine learning and uncertainty quantification, Comput. Methods Appl. Mech. Engrg. 431 (2024) 117314, http://dx.doi.org/10.1016/j.cma.2024.117314.
[34] M.B. Yelten, T. Zhu, S. Koziel, P.D. Franzon, M.B. Steer, Demystifying surrogate modeling for circuits and systems, IEEE Circuits Syst. Mag. 12 (1) (2012) 45–63, http://dx.doi.org/10.1109/MCAS.2011.2181095.
[35] R. Trinchero, M. Larbi, H.M. Torun, F.G. Canavero, M. Swaminathan, Machine learning and uncertainty quantification for surrogate models of integrated devices with a large number of parameters, IEEE Access 7 (2018) 4056–4066, http://dx.doi.org/10.1109/ACCESS.2018.2888903.
[36] T. Nguyen, B. Shi, H. Ma, E.-P. Li, X. Chen, A.C. Cangellaris, J. Schutt-Aine, Comparative study of surrogate modeling methods for signal integrity and microwave circuit applications, IEEE Trans. Compon. Packag. Manuf. Technol. 11 (9) (2021) 1369–1379, http://dx.doi.org/10.1109/TCPMT.2021.3098666.
[37] J. Kudela, R. Matousek, Recent advances and applications of surrogate models for finite element method computations: A review, Soft Comput. 26 (24) (2022) 13709–13733, http://dx.doi.org/10.1007/s00500-022-07362-8.
[38] Y. Feng, D. Wu, M.G. Stewart, W. Gao, Past, current and future trends and challenges in non-deterministic fracture mechanics: A review, Comput. Methods Appl. Mech. Engrg. 412 (2023) 116102, http://dx.doi.org/10.1016/j.cma.2023.116102.
[39] H. Zhao, C. Fu, Y. Zhang, W. Zhu, K. Lu, E.M. Francis, Dimensional decomposition-aided metamodels for uncertainty quantification and optimization in engineering: A review, Comput. Methods Appl. Mech. Engrg. 428 (2024) 117098, http://dx.doi.org/10.1016/j.cma.2024.117098.
[40] Z. Azarhoosh, M.I. Ghazaan, A review of recent advances in surrogate models for uncertainty quantification of high-dimensional engineering applications, Comput. Methods Appl. Mech. Engrg. 433 (2025) 117508, http://dx.doi.org/10.1016/j.cma.2024.117508.
[41] S. Ranftl, W. von der Linden, M..S. Committee, Bayesian surrogate analysis and uncertainty propagation, in: Physical Sciences Forum, Vol. 3, MDPI, 2021, p. 6, http://dx.doi.org/10.3390/psf2021003006.
[42] C.E. Rasmussen, C.K. Williams, Gaussian Processes for Machine Learning, The MIT Press, Cambridge, MA, 2006.
[43] M. Swaminathan, H.M. Torun, H. Yu, J.A. Hejase, W.D. Becker, Demystifying machine learning for signal and power integrity problems in packaging, IEEE Trans. Compon. Packag. Manuf. Technol. 10 (8) (2020) 1276–1295, http://dx.doi.org/10.1109/TCPMT.2020.3011910.
[44] R. Garnett, Bayesian Optimization, Cambridge University Press, Cambridge, UK, 2023.
[45] Y. Morita, S. Rezaeiravesh, N. Tabatabaei, R. Vinuesa, K. Fukagata, P. Schlatter, Applying Bayesian optimization with Gaussian process regression to computational fluid dynamics problems, J. Comput. Phys. 449 (2022) 110788, http://dx.doi.org/10.1016/j.jcp.2021.110788.
[46] Y.-J. Chang, H.-Y. Huang, R.-L. Chern, Y.-J. Chou, A multiscale computational framework using active learning to model complex suspension flows, J. Comput. Phys. 493 (2023) 112481, http://dx.doi.org/10.1016/j.jcp.2023.112481.
[47] Y. Xiang, T. Han, Y. Li, L. Shi, B. Pan, A multi-region active learning Kriging method for response distribution construction of highly nonlinear problems, Comput. Methods Appl. Mech. Engrg. 419 (2024) 116650, http://dx.doi.org/10.1016/j.cma.2023.116650.
[48] H. Zhan, N.-C. Xiao, Expected lifetime prediction for time-and space-dependent structural systems based on active learning surrogate model, Comput. Methods Appl. Mech. Engrg. 429 (2024) 117150, http://dx.doi.org/10.1016/j.cma.2024.117150.
[49] R. Haylock, A. O'Hagan, On inference for outputs of computationally expensive algorithms with uncertainty on the inputs, in: J. Bernardo, J. Berger, A. Dawid, A. Smith (Eds.), Bayesian Statistics 5, Oxford University Press, Oxford, 1996, pp. 629–637.
[50] J.E. Oakley, A. O'Hagan, Probabilistic sensitivity analysis of complex models: A Bayesian approach, J. R. Stat. Soc. Ser. B Stat. Methodol. 66 (3) (2004) 751–769, http://dx.doi.org/10.1111/j.1467-9868.2004.05304.x.
[51] A. O'Hagan, Bayesian analysis of computer code outputs: A tutorial, Reliab. Eng. Syst. Saf. 91 (10–11) (2006) 1290–1300, http://dx.doi.org/10.1016/j.ress.2005.11.025.
[52] I. Bilionis, N. Zabaras, Multi-output local Gaussian process regression: Applications to uncertainty quantification, J. Comput. Phys. 231 (17) (2012) 5718–5746, http://dx.doi.org/10.1016/j.jcp.2012.04.047.
[53] I. Bilionis, N. Zabaras, B.A. Konomi, G. Lin, Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification, J. Comput. Phys. 241 (2013) 212–239, http://dx.doi.org/10.1016/j.jcp.2013.01.011.
[54] P. Chen, N. Zabaras, I. Bilionis, Uncertainty propagation using infinite mixture of Gaussian processes and variational Bayesian inference, J. Comput. Phys. 284 (2015) 291–333, http://dx.doi.org/10.1016/j.jcp.2014.12.028.
[55] R. Tripathy, I. Bilionis, M. Gonzalez, Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation, J. Comput. Phys. 321 (2016) 191–223, http://dx.doi.org/10.1016/j.jcp.2016.05.039.
[56] I. Bilionis, N. Zabaras, Bayesian uncertainty propagation using Gaussian processes, in: R. Ghanem, D. Higdon, H. Owhadi (Eds.), Handbook of Uncertainty Quantification, Springer International Publishing Switzerland, Cham, Switzerland, 2016, pp. 555–599.
[57] D.G. Giovanis, M.D. Shields, Data-driven surrogates for high dimensional models using Gaussian process regression on the Grassmann manifold, Comput. Methods Appl. Mech. Engrg. 370 (2020) 113269, http://dx.doi.org/10.1016/j.cma.2020.113269.
[58] C. Chang, T. Zeng, A hybrid data-driven-physics-constrained Gaussian process regression framework with deep kernel for uncertainty quantification, J. Comput. Phys. 486 (2023) 112129, http://dx.doi.org/10.1016/j.jcp.2023.112129.
[59] Y.-H. Yeung, R. Tipireddy, D.A. Barajas-Solano, A.M. Tartakovsky, Conditional Karhunen–Loève regression model with Basis Adaptation for high-dimensional problems: Uncertainty quantification and inverse modeling, Comput. Methods Appl. Mech. Engrg. 418 (2024) 116487, http://dx.doi.org/10.1016/j.cma.2023.116487.
[60] P. Manfredi, Probabilistic uncertainty propagation using Gaussian process surrogates, Int. J. Uncertain. Quantif. 14 (6) (2024) http://dx.doi.org/10.1615/Int.J.UncertaintyQuantification.2024052162.
[61] P. Manfredi, R. Trinchero, Nonparametric formulation of polynomial chaos expansion based on least-square support-vector machines, Eng. Appl. Artif. Intell. 133 (2024) 108182, http://dx.doi.org/10.1016/j.engappai.2024.108182.

[62] K. Cheng, Z. Lu, Y. Zhen, Multi-level multi-fidelity sparse polynomial chaos expansion based on Gaussian process regression, Comput. Methods Appl. Mech. Engrg. 349 (2019) 360–377, http://dx.doi.org/10.1016/j.cma.2019.02.021.
[63] M. Guo, J.S. Hesthaven, Reduced order modeling for nonlinear structural analysis using Gaussian process regression, Comput. Methods Appl. Mech. Engrg. 341 (2018) 807–826, http://dx.doi.org/10.1016/j.cma.2018.07.017.
[64] B. Bhattacharyya, Uncertainty quantification of dynamical systems by a POD–Kriging surrogate model, J. Comput. Sci. 60 (2022) 101602, http://dx.doi.org/10.1016/j.jocs.2022.101602.
[65] Q. Lu, L. Wang, L. Li, Efficient uncertainty quantification of stochastic problems in CFD by combination of compressed sensing and POD-Kriging, Comput. Methods Appl. Mech. Engrg. 396 (2022) 115118, http://dx.doi.org/10.1016/j.cma.2022.115118.
[66] C. Ding, H. Rappel, T. Dodwell, Full-field order-reduced Gaussian Process emulators for nonlinear probabilistic mechanics, Comput. Methods Appl. Mech. Engrg. 405 (2023) 115855, http://dx.doi.org/10.1016/j.cma.2022.115855.
[67] Z. Song, Z. Liu, H. Zhang, P. Zhu, An improved sufficient dimension reduction-based Kriging modeling method for high-dimensional evaluation-expensive problems, Comput. Methods Appl. Mech. Engrg. 418 (2024) 116544, http://dx.doi.org/10.1016/j.cma.2023.116544.
[68] A.F. Cortesi, G. Jannoun, P.M. Congedo, Kriging-sparse Polynomial Dimensional Decomposition surrogate model with adaptive refinement, J. Comput. Phys. 380 (2019) 212–242, http://dx.doi.org/10.1016/j.jcp.2018.10.051.
[69] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, Reliab. Eng. Syst. Saf. 93 (7) (2008) 964–979, http://dx.doi.org/10.1016/j.ress.2007.04.002.
[70] M. Eigel, C.J. Gittelson, C. Schwab, E. Zander, Adaptive stochastic Galerkin FEM, Comput. Methods Appl. Mech. Engrg. 270 (2014) 247–269, http://dx.doi.org/10.1016/j.cma.2013.11.015.
[71] P. Pettersson, H.A. Tchelepi, Stochastic Galerkin framework with locally reduced bases for nonlinear two-phase transport in heterogeneous formations, Comput. Methods Appl. Mech. Engrg. 310 (2016) 367–387, http://dx.doi.org/10.1016/j.cma.2016.07.013.
[72] S. Pranesh, D. Ghosh, Cost reduction of stochastic Galerkin method by adaptive identification of significant polynomial chaos bases for elliptic equations, Comput. Methods Appl. Mech. Engrg. 340 (2018) 54–69, http://dx.doi.org/10.1016/j.cma.2018.04.043.
[73] T.V. Mathew, A. Pramod, E.T. Ooi, S. Natarajan, An efficient forward propagation of multiple random fields using a stochastic Galerkin scaled boundary finite element method, Comput. Methods Appl. Mech. Engrg. 367 (2020) 112994, http://dx.doi.org/10.1016/j.cma.2020.112994.
[74] B. Sousedík, R. Price, A stochastic Galerkin method with adaptive time-stepping for the Navier–Stokes equations, J. Comput. Phys. 468 (2022) 111456, http://dx.doi.org/10.1016/j.jcp.2022.111456.
[75] A. Medaglia, L. Pareschi, M. Zanella, Stochastic Galerkin particle methods for kinetic equations of plasmas with uncertainties, J. Comput. Phys. 479 (2023) 112011, http://dx.doi.org/10.1016/j.jcp.2023.112011.
[76] M. Zhong, T. Xiao, M.J. Krause, M. Frank, S. Simonis, A stochastic Galerkin lattice Boltzmann method for incompressible fluid flows with uncertainties, J. Comput. Phys. (2024) 113344, http://dx.doi.org/10.1016/j.jcp.2024.113344.
[77] Y.C. Pati, R. Rezaiifar, P.S. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, in: Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, IEEE, 1993, pp. 40–44, http://dx.doi.org/10.1109/ACSSC.1993.342465.
[78] P. Diaz, A. Doostan, J. Hampton, Sparse polynomial chaos expansions via compressed sensing and D-optimal design, Comput. Methods Appl. Mech. Engrg. 336 (2018) 640–666, http://dx.doi.org/10.1016/j.cma.2018.03.020.
[79] S.D. Babacan, R. Molina, A.K. Katsaggelos, Bayesian compressive sensing using Laplace priors, IEEE Trans. Image Process. 19 (1) (2009) 53–63, http://dx.doi.org/10.1109/TIP.2009.2032894.
[80] K. Sargsyan, C. Safta, H.N. Najm, B.J. Debusschere, D. Ricciuto, P. Thornton, Dimensionality reduction for complex models via Bayesian compressive sensing, Int. J. Uncertain. Quantif. 4 (1) (2014) http://dx.doi.org/10.1615/Int.J.UncertaintyQuantification.2013006821.
[81] S. Marelli, N. Lüthen, B. Sudret, UQLab User Manual – Polynomial Chaos Expansions, Technical Report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2022, Report UQLab-V2.0-104.
[82] Q. Shao, A. Younes, M. Fahs, T.A. Mara, Bayesian sparse polynomial chaos expansion for global sensitivity analysis, Comput. Methods Appl. Mech. Engrg. 318 (2017) 474–496, http://dx.doi.org/10.1016/j.cma.2017.01.033.
[83] B. Bhattacharyya, Structural reliability analysis by a Bayesian sparse polynomial chaos expansion, Struct. Saf. 90 (2021) 102074, http://dx.doi.org/10.1016/j.strusafe.2020.102074.
[84] C. Lataniotis, D. Wicaksono, S. Marelli, B. Sudret, UQLab User Manual – Kriging (Gaussian Process Modeling), Technical Report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2022, Report UQLab-V2.0-105.
[85] I.M. Sobol, Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, Math. Comput. Simulation 55 (1–3) (2001) 271–280, http://dx.doi.org/10.1016/S0378-4754(00)00270-6.
[86] F.G. Mehler, Ueber die Entwicklung einer Function von Beliebig Vielen Variablen nach Laplaceschen Functionen Höherer Ordnung, Walter de Gruyter, Berlin, NY, 1866, http://dx.doi.org/10.1515/crll.1866.66.161.
[87] W. Kibble, An extension of a theorem of Mehler's on Hermite polynomials, in: Mathematical Proceedings of the Cambridge Philosophical Society, Vol. 41, Cambridge University Press, 1945, pp. 12–15, http://dx.doi.org/10.1017/S0305004100022313.
[88] F. Anli, S. Gungor, Some useful properties of Legendre polynomials and its applications to neutron transport equation in slab geometry, Appl. Math. Model. 31 (4) (2007) 727–733, http://dx.doi.org/10.1016/j.apm.2005.12.005.
[89] V.N. Vapnik, Statistical Learning Theory, Wiley-Interscience, New York, NY, 1998.
[90] A.C. Rencher, G.B. Schaalje, Linear Models in Statistics, John Wiley & Sons, 2008.
[91] A. Das, W.S. Geisler, A method to integrate and classify normal distributions, J. Vis. 21 (10) (2021) 1, http://dx.doi.org/10.1167/jov.21.10.1.
[92] S. Marelli, B. Sudret, UQLab: A framework for uncertainty quantification in Matlab, in: Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management, 2014, pp. 2554–2563, http://dx.doi.org/10.1061/9780784413609.257.
[93] A. Das, Generalized chi-square distribution, version 1.8.7, 2021, URL: https://it.mathworks.com/matlabcentral/fileexchange/85028-generalized-chi-square-distribution.
[94] Synopsis Inc., HSPICE – release version B-2008.09, 2008, URL: https://www.synopsys.com/implementation-and-signoff/ams-simulation/primesim-hspice.html.
[95] L. Ambrogioni, E. Maris, Complex-valued Gaussian process regression for time series analysis, Signal Process. 160 (2019) 215–228, http://dx.doi.org/10.1016/j.sigpro.2019.02.011.
[96] N. Soleimani, R. Trinchero, Compressed complex-valued least squares support vector machine regression for modeling of the frequency-domain responses of electromagnetic structures, Electronics 11 (4) (2022) 551, http://dx.doi.org/10.3390/electronics11040551.
[97] K.I. Park, Fundamentals of Probability and Stochastic Processes with Applications to Communications, Springer, Cham, Switzerland, 2018.
[98] M. Katzfuss, J. Guinness, A general framework for Vecchia approximations of Gaussian processes, 2021, http://dx.doi.org/10.1214/19-STS755.