Conformalized Polynomial Chaos Expansion for Uncertainty-aware Surrogate Modeling

Dimitrios Loukrezis [1,*] and Dimitris G. Giovanis [2]

[1]Centrum Wiskunde & Informatica, Scientific Computing, Amsterdam, The Netherlands
[2]Johns Hopkins University, Department of Civil & Systems Engineering, Baltimore, USA
[*]Corresponding author: `d.loukrezis@cwi.nl`

Abstract

This work introduces a method to equip data-driven polynomial chaos expansion surrogate models with intervals that quantify the predictive uncertainty of the surrogate. To that end, jackknife-based conformal prediction is integrated into regression-based polynomial chaos expansions. The jackknife algorithm uses leave-one-out residuals to generate predictive intervals around the predictions of the polynomial chaos surrogate. The jackknife+ extension additionally requires leave-one-out model predictions. Both methods allow to use the entire dataset for model training and do not require a hold-out dataset for prediction interval calibration. The key to efficient implementation is to leverage the linearity of the polynomial chaos regression model, so that leave-one-out residuals and, if necessary, leave-one-out model predictions can be computed with analytical, closed-form expressions. This eliminates the need for repeated model re-training. The conformalized polynomial chaos expansion method is first validated on four benchmark models and then applied to two electrical engineering design use-cases. The method produces predictive intervals that provide the target coverages, even for low-accuracy models trained with small datasets. At the same time, training data availability plays a crucial role in improving the empirical coverage and narrowing the predictive interval, as well as in reducing their variability over different training datasets.

Keywords: conformal prediction, jackknife, polynomial chaos expansion, regression, surrogate modeling, uncertainty quantification.

1. Introduction

Originally developed as an efficient uncertainty quantification (UQ) method [1–3], the polynomial chaos expansion (PCE) is in recent years routinely employed as a machine learning (ML) regression model [4]. Within this context, regression-based PCEs are often used as data-driven surrogate models in various science and engineering applications [5–10]. Their use has also been extended to scientific ML tasks like manifold learning [11–14], operator learning [15], and physics-informed ML [16, 17]. In contrast to the use of PCEs for UQ purposes, PCE-based data-driven predictive models are purely deterministic. That is, they yield point predictions that lack any type of uncertainty or confidence metrics. Models that provide such metrics are, for example, Gaussian processes (GPs) [18] and Bayesian neural networks (NNs) [19]. Ensemble modeling and bootstrapping are common ways of providing such predictive uncertainty intervals to data-driven models, including PCEs [20, 21]. Other relevant approaches include combined PCE-GP models [22, 23] or stochastic PCE formulations [24]. However, these approaches induce either specific modeling limitations or significant computational overhead due to model retraining.

Considering predictive UQ for ML models, conformal prediction (CP) [25] is an established method for producing statistically valid prediction intervals given a point predictor, only under the assumption of data exchangeability. The resulting predictive models are commonly called “conformalized” [26–30]. CP is a distribution-free UQ method, meaning that it does not require any assumptions about the underlying data distribution or the structure of the data-driven model. Additionally, certain CP methods provide statistically rigorous, finite-sample guarantees regarding prediction interval coverage. On its downsides, CP does not yield full probability distributions like Bayesian methods do. Moreover, its coverage is marginal instead of conditional, hence possibly under- or overconfindent in different regions of the input space. Adaptive CP methods that address the latter issue are actively pursued [31, 32]. Data demand can also be an issue, in particular considering so-called inductive or split CP methods [33, 34], which require that part of the available data is excluded from training and reserved for prediction interval estimation. Remedies have been sought in so-called cross CP approaches leveraging cross-validation techniques [35, 36], including jackknife-based variants that utilize leave-one-out (LOO) residuals [34, 37]. Similar to bootstrapping and ensemble modeling, the downside of these methods is the need for repeated model retraining, which can result in undesired computational overhead.

This work proposes a novel combination of regression-based PCEs with CP based on the jackknife method [34] and its jackknife+ extension [37]. The resulting conformalized PCE models are equipped with predictive intervals that quantify the uncertainty in their predictions. Combining CP with data-driven surrogate models has been explored before, e.g., for GPs [30] or various NNs architectures [38]. However, the use of regression-based PCEs results in an uncertainty-aware, data-driven surrogate modeling framework, which is additionally very computationally efficient. The key enabler is the linear form of PCE regression models, which allows to compute the LOO residuals required by the jackknife method analytically in closed form, without the need to re-train the PCE. The same is true for the LOO model predictions needed by the jackknife+ method. Due to this synergy of jackknife-based CP and PCE regression, the predictive intervals are estimated with minimal computational overhead. The suggested approach belongs to the class of cross CP methods, thus allowing the full available dataset to be used for training. This is a crucial advantage in small-data learning regimes. To the authors’ knowledge, this is the first work to develop a conformalized PCE regression method.

The method is implemented for total-degree PCEs utilizing ordinary least squares (OLS) for the computation of the PCE coefficients. Extensions to other types of polynomial bases and coefficient estimation methods are conceptually straightforward, with the requirement for a fixed polynomial basis so that the closed-form expressions for the LOO residuals and model predictions remain valid (see section 2.3). The method is first validated on four benchmark models and then applied to two electrical engineering design use-cases. The first use-case concerns the design of a heat sink for power electronics cooling [6]. The second use-case concerns the design of a Stern-Gerlach electromagnet [39]. In all numerical experiments, we assess whether the conformalized PCE surrogate provides predictive uncertainty intervals that result in empirical coverages similar to the required target levels. Additionally, we investigate the impact of different conformalized PCE configurations, with an emphasis on training data availability.

The remaining of this paper is organized as follows. Section 2 presents the methodology behind the suggested method, first recalling the necessary prerequisites of PCE and CP methods, and then presenting the conformalized PCE. The numerical experiments are presented in section 3. Section 4 summarizes the findings of this work and discusses follow-up research opportunities.

2. Methodology

We begin by presenting the PCE regression model in section 2.1, followed by a presentation of CP section 2.2. The latter focuses on the jackknife and jackknife+ CP methods. Section 2.3 shows how LOO residuals and predictions of PCE regression models can be computed with analytical, closed-form expressions, as to enable the cost-effective estimation of predictive intervlas based on jackknife and jackknife+ CP. Last, section 2.4 discusses non-conformity score normalization for conformalized PCEs.

2.1 Regression-based polynomial chaos expansion

In the following we consider a computational model, denoted as $\mu(\mathbf{x})$, which receives an input vector $\mathbf{x} \in \mathbb{R}^N$ and computes a scalar output $y \in \mathbb{R}$, such that $y = \mu(\mathbf{x})$. While scalar outputs are considered throughout this work, vector-valued outputs can also be considered by an element-wise application of all operations described next. The form of the model can range from a simple analytical function to a complex numerical solver. The only assumption is that the model is deterministic, i.e., the same output y is observed given the same input $\mathbf{x}$.

Next, we introduce a multivariate random variable (or random vector) $\mathbf{X} = (X_1, \dots, X_N)$, where the individual random variables X_n, $n = 1, \dots, N$, are assumed to be mutually independent. The random vector is defined on the probability space (Ω, Σ, P), where Ω is the sample space containing all measurable outcomes $\omega \in \Omega$, Σ denotes the σ-algebra of events (grouped outcomes), and $P : \Sigma \to [0,1]$ is a probability measure. The latter is defined based on the assumed probability density function (PDF) $f_{\mathbf{X}} : \Xi \to \mathbb{R}_{\geq 0}$, where $\Xi \subset \mathbb{R}^N$ denotes the state-space of the random vector. Then, the random vector is a map, $\mathbf{X} : \Omega \to \Xi$, with random realizations $\mathbf{x} = \mathbf{X}(\omega) \in \Xi$.

Next, we assume that the random realizations $\mathbf{x} = \mathbf{X}(\omega)$ coincide with the inputs of the computational model $\mu(\mathbf{x})$ and that the Doob-Dynkin lemma is satisfied [40]. Then, by propagation of uncertainty, the model's response is a random variable dependent on the input random vector, i.e., $Y = \mu(\mathbf{X})$. Provided that its variance is finite, the random output can be represented in the form of a PCE, such that

$$y = \mu(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}_0^N} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\mathbf{x}), \tag{1}$$

where $\Psi_{\boldsymbol{\alpha}}$ are multivariate polynomials, $c_{\boldsymbol{\alpha}} \in \mathbb{R}$ are the associated coefficients, and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N)$ are unique multi-indices defined by the polynomial's partial degree in each input dimension. The polynomials are chosen such that they form an orthogonal basis with respect to the input PDF, i.e., they satisfy:

$$\mathbb{E}\left[\Psi_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\beta}}\right] = \int_{\Xi} \Psi_{\boldsymbol{\alpha}}(\mathbf{x}) \Psi_{\boldsymbol{\beta}}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \mathbb{E}\left[\Psi_{\boldsymbol{\alpha}}^2\right] \delta_{\boldsymbol{\alpha}\boldsymbol{\beta}}. \tag{2}$$

For simplicity, we assume an orthonormal basis, such that $\mathbb{E}\left[\Psi_{\boldsymbol{\alpha}}^2\right] = 1$.

In practice, the infinite series (1) must be truncated to a finite number of K terms based on a set of selected multi-indices. This truncation yields a PCE approximation of the form

$$y = \mu(\mathbf{x}) \approx \widehat{\mu}(\mathbf{x}) = \sum_{k=1}^{K} c_k \Psi_k(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \Lambda} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\mathbf{x}), \tag{3}$$

where $\Lambda \subset \mathbb{N}_0^N$ is a multi-index set with cardinality $\#\Lambda = K$. Each index $k = 1, \dots, K$ corresponds to a unique multi-index $\boldsymbol{\alpha} \in \Lambda$. One common truncation method is to use a so-called total-degree basis, where $\Lambda = \{\boldsymbol{\alpha} : |\boldsymbol{\alpha}|_1 \leq P\}$ and P denotes the maximum polynomial degree. The size of the total-degree basis is equal to $K = \frac{(N+P)!}{N!P!}$ and becomes intractable for high polynomial degrees or input dimensions. To alleviate this problem, several methods have been suggested for constructing sparse bases [41], e.g., by means of dimension-adaptive algorithms [42].

Given a multi-index set Λ and the corresponding polynomial basis, the remaining task is to estimate the PCE coefficients $\mathbf{c} = (c_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \Lambda}^{\top} \in \mathbb{R}^{K \times 1}$. We opt here for a regression-based approach [43–45]. Alternatives include include pseudo-spectral projection [46, 47] or interpolation [48–50]. For the regression-based coefficient estimation, we assume a set of input parameter realizations along with the corresponding model responses $\left\{\mathbf{x}^{(m)}, y^{(m)}\right\}_{m=1}^{M}$, commonly called the experimental design or, in an ML-context, the training data set. We then construct the so-called design matrix $\mathbf{D} \in \mathbb{R}^{M \times K}$, such that $d_{mk} = \Psi_k\left(\mathbf{x}^{(m)}\right)$. Collecting the model responses in a vector $\mathbf{y} = \left(y^{(1)}, \dots, y^{(M)}\right)^{\top} \in \mathbb{R}^{M \times 1}$, the coefficients can be computed by solving the least squares minimization problem

$$\mathbf{c} = \underset{\hat{\mathbf{c}} \in \mathbb{R}^K}{\arg\min} \left\{ \frac{1}{M} \|\mathbf{y} - \mathbf{D}\hat{\mathbf{c}}\|_2 \right\}, \tag{4}$$

which admits the closed-form solution

$$\mathbf{c} = \left(\mathbf{D}^\top \mathbf{D}\right)^{-1} \mathbf{D}^\top \mathbf{y}. \tag{5}$$

For the regression problem (4) to be well posed, the experimental design must be at minimum equal to the polynomial basis. Larger experimental designs are typically needed for well determined problems. In fact, the size of the experimental design is crucial for the stability and accuracy of the resulting PCE [43, 44]. This limitation can be overcome with sparse PCE methods, where the problem (4) is complemented with a penalty term that induces sparsity in its solution. Common techniques include the least absolute shrinkage and selection operator (LASSO) and compressive sensing [41].

2.2 Conformal prediction

Following the notation introduced in section 2.1, we consider a regression model $\widehat{\mu}(\mathbf{x})$ trained with the dataset $\mathcal{D}_{\text{train}} = \left\{\mathbf{x}^{(m)}, y^{(m)}\right\}_{m=1}^{M}$. Then, given a test point $\{\mathbf{x}^*, y^*\} \notin \mathcal{D}_{\text{train}}$, the trained model produces a prediction $\widehat{y}^* = \widehat{\mu}(\mathbf{x}^*)$ for the true value y^*. The goal of CP is to complement the model's prediction with an interval which is likely to contain the true output y^*. More formally, given a significance level $s \in (0, 1)$ corresponding to the target coverage level $(1-s)$, CP constructs a prediction interval $\widehat{C}_{1-s}(\mathbf{x}^*)$, such that

$$\mathbb{P}\left\{y^* \in \widehat{C}_{1-s}(\mathbf{x}^*)\right\} \geq 1 - s. \tag{6}$$

That is, the probability of the true output being included in the interval is at least equal to the target coverage level. Note that CP assumes that the training and test data are exchangeable, meaning that the joint data distribution is invariant to permutation.

A central concept of CP is the so-called non-conformity score, which measures how well a given data point conforms to the fitted model. To that end, a non-conformity function $\alpha(\mathbf{x}, y; \widehat{\mu})$ is evaluated on a set of calibration points $\left\{\mathbf{x}^{(j)}, y^{(j)}\right\}_{j=1}^{J}$, thus producing the non-conformity scores $\alpha_j = \alpha\left(\mathbf{x}^{(j)}, y^{(j)}; \widehat{\mu}\right)$, $j = 1, \dots, J$. We denote the $(1-s)$-quantile of the empirical distribution of the non-conformity scores $\{\alpha_j\}_{j=1}^{J}$ as $\widehat{q}_{1-s}\{\alpha_j\}$. The prediction interval for a test input $\mathbf{x}^*$ is then defined in general by

$$\widehat{C}_{1-s}(\mathbf{x}^*) \;=\; \{\, y : \; \alpha(\mathbf{x}^*, y; \widehat{\mu}) \leq \widehat{q}_{1-s}\{\alpha_j\}\,\}, \tag{7}$$

such that $y^* \in \widehat{C}_{1-s}(\mathbf{x}^*)$ implies that $\alpha(\mathbf{x}^*, y^*; \widehat{\mu}) \leq \widehat{q}_{1-s}\{\alpha_j\}$ and vice versa. A commonly used non-conformity score is the absolute residual, such that $\alpha_j = |r_j| = \left|y^{(j)} - \widehat{\mu}\left(\mathbf{x}^{(j)}\right)\right|$. This choice results in the construction of the symmetric predictive interval

$$\widehat{C}_{1-s}(\mathbf{x}^*) = \left[\widehat{\mu}(\mathbf{x}^*) - \widehat{q}_{1-s}\{|r_j|\}, \widehat{\mu}(\mathbf{x}^*) + \widehat{q}_{1-s}\{|r_j|\}\right]. \tag{8}$$

Note that this predictive interval does not generally adapt to possible data heteroskedasticity.

A naive approach to estimate the predictive interval (8) would be to use the residuals on the training data $\mathcal{D}_{\text{train}} = \left\{\mathbf{x}^{(m)}, y^{(m)}\right\}_{m=1}^{M}$. However, training data residuals are expected to be smaller compared to residuals obtained from previously unseen test data points. Hence, the resulting predictive interval will most probably not provide the required coverage. Split or inductive CP approaches [33, 34] use a separate validation dataset $\mathcal{D}_{\text{val}} \neq \mathcal{D}_{\text{train}}$ to compute the non-conformity scores and construct the prediction interval (8). Split CP guarantees the required finite-sample coverage (6), however, this comes at the cost of a larger data demand. To avoid the need for additional data, cross CP [35, 36] utilizes K-fold cross-validation, such that the training dataset $\mathcal{D}_{\text{train}}$ is split into K disjoint subsets $\mathcal{D}_1, \dots, \mathcal{D}_K$

of (approximately) equal size. Then, the regression models $\widehat{\mu}_{\sim k}(\mathbf{x})$ are trained using the datasets $\mathcal{D}_{\sim k} = \mathcal{D}_{\text{train}} \setminus \mathcal{D}_k$, $k = 1, \ldots, K$, and the cross-validation residuals

$$r_m^{\text{cv}} = y^{(m)} - \widehat{\mu}_{\sim k(m)}\left(\mathbf{x}^{(m)}\right), \tag{9}$$

are used as non-conformity scores (in absolute value), where the index $k(m)$ denotes that $\left(\mathbf{x}^{(m)}, y^{(m)}\right) \in \mathcal{D}_k$. Cross CP typically yields an empirical coverage close to the target level $(1-s)$, but does not provide formal coverage guarantees. A major downside of cross CP is that K different regression models must be trained, which can lead to an undesired computational cost.

A special case of cross-validation is the jackknife method [51]. Using the jackknife in the context of CP [34], the non-conformity scores coincide with the absolute values of the LOO residuals

$$r_m^{\text{LOO}} = y^{(m)} - \widehat{\mu}_{\sim m}\left(\mathbf{x}^{(m)}\right), \tag{10}$$

where the regression model $\widehat{\mu}_{\sim m}(\mathbf{x})$ is trained using the dataset $\mathcal{D}_{\sim m} = \mathcal{D}_{\text{train}} \setminus \left\{\left(\mathbf{x}^{(m)}, y^{(m)}\right)\right\}$. Then, the predictive interval (8) takes two equivalent forms, i.e.,

$$\widehat{C}_{1-s}^{\text{jack}}(\mathbf{x}^*) = \left[\widehat{\mu}(\mathbf{x}^*) - \widehat{q}_{1-s}\left\{\left|r_m^{\text{LOO}}\right|\right\}, \widehat{\mu}(\mathbf{x}^*) + \widehat{q}_{1-s}\left\{\left|r_m^{\text{LOO}}\right|\right\}\right] \tag{11a}$$

$$= \left[\widehat{q}_s\left\{\widehat{\mu}(\mathbf{x}^*) - \left|r_m^{\text{LOO}}\right|\right\}, \widehat{q}_{1-s}\left\{\widehat{\mu}(\mathbf{x}^*) + \left|r_m^{\text{LOO}}\right|\right\}\right]. \tag{11b}$$

Being a cross CP variant, jackknife CP also yields an empirical coverage close to the target level, also without formal coverage guarantees. Barber et al. [37] suggested an extension of jackknife CP, called jackknife+, where the predictive interval is constructed as

$$C_{1-s}^{\text{jack+}}(\mathbf{x}^*) = \left[\widehat{q}_s\left\{\widehat{\mu}_{\sim m}(\mathbf{x}^*) - \left|r_m^{\text{LOO}}\right|\right\}, \widehat{q}_{1-s}\left\{\widehat{\mu}_{\sim m}(\mathbf{x}^*) + \left|r_m^{\text{LOO}}\right|\right\}\right]. \tag{12}$$

In contrast to the jackknife interval (11) which is centered at the prediction $\widehat{\mu}(\mathbf{x}^*)$ of the fully trained model, the jackknife+ interval (12) is based on the ensemble of the LOO predictions $\widehat{\mu}_{\sim m}(\mathbf{x}^*)$, $m = 1, \ldots, M$. Hence, the jackknife+ interval (12) is asymmetric. This slight modification results in a formal coverage guarantee of $(1-2s)$, while the empirical coverage remains close to the target level $(1-s)$, similar to cross CP and jackknife CP.

In practice, both the jackknife and jackknife+ CP methods achieve an empirical coverage close to the target level $(1-s)$ and produce almost identical predictive intervals [37]. Moreover, both approaches entail training M regression models, thus being extreme cases of cross CP in terms of computational cost. However, for linear regression models like the PCE, the LOO residuals r_m^{LOO} and the LOO predictions $\widehat{\mu}_{\sim m}(\mathbf{x}^*)$ can be obtained without model retraining, as shown in section 2.3.

2.3 Conformalized polynomial chaos regression

We now assume that $\widehat{\mu}(\mathbf{x})$ is a regression-based PCE in the form of (3) and trained with a dataset $\mathcal{D}_{\text{train}} = \left\{\mathbf{x}^{(m)}, y^{(m)}\right\}_{m=1}^{M}$. Then, the predictive intervals (11) (for jackknife) and (12) (for jackknife+) can be estimated without the need to retrain the PCE M times.

Starting with the jackknife method, only the LOO residuals defined in (10) are needed to estimate the prediction interval (11). Recalling from section 2.1 the design matrix $\mathbf{D} \in \mathbb{R}^{M \times K}$, the so-called hat matrix $\mathbf{H} \in \mathbb{R}^{M \times M}$ is defined as

$$\mathbf{H} = \mathbf{D}\left(\mathbf{D}^\top \mathbf{D}\right)^{-1} \mathbf{D}^\top. \tag{13}$$

The LOO residuals can then be computed analytically via the closed-form expression

$$r_m^{\text{LOO}} = \frac{y^{(m)} - \widehat{\mu}\left(\mathbf{x}^{(m)}\right)}{1 - h_{mm}}, \tag{14}$$

where $h_{mm} \in \mathbb{R}$ is the m-th diagonal term of $\mathbf{H}$. The derivation of the analytical expression (14) is given in appendix A.

For the predictive interval (12) based on the jackknife+, the LOO predictions $\{\widehat{\mu}_{\sim m}(\mathbf{x}^*)\}_{m=1}^{M}$ for the test input $\mathbf{x}^*$ are additionally needed. Let us denote with $\mathbf{d}_m^\top \in \mathbb{R}^{1\times K}$ the m-th row of the design matrix $\mathbf{D} \in \mathbb{R}^{M\times K}$, which contains the evaluation of the PCE basis on the training input $\mathbf{x}^{(m)}$, such that $\mathbf{d}_m^\top = \left(\Psi_1\left(\mathbf{x}^{(m)}\right), \dots, \Psi_K\left(\mathbf{x}^{(m)}\right)\right)$. Accordingly, let us denote with $\mathbf{d}_*^\top \in \mathbb{R}^{1\times K}$ the vector of basis evaluations at the test input $\mathbf{x}^*$, such that $\mathbf{d}_*^\top = \left(\Psi_1\left(\mathbf{x}^*\right), \dots, \Psi_K\left(\mathbf{x}^*\right)\right)$. Then, the LOO predictions can be analytically computed as

$$\begin{aligned}\widehat{\mu}_{\sim m}\left(\mathbf{x}^*\right) &= \widehat{\mu}\left(\mathbf{x}^*\right) - \mathbf{d}_*^\top\left(\mathbf{D}^\top\mathbf{D}\right)^{-1}\mathbf{d}_m\frac{y^{(m)} - \widehat{\mu}\left(\mathbf{x}^{(m)}\right)}{1 - h_{mm}} \\ &= \widehat{\mu}\left(\mathbf{x}^*\right) - \mathbf{d}_*^\top\left(\mathbf{D}^\top\mathbf{D}\right)^{-1}\mathbf{d}_m r_m^{\mathrm{LOO}}. \end{aligned} \tag{15}$$

The derivation of the analytical expression (15) is given in appendix B.

The analytical expressions (14) and (15) enable the efficient computation of jackknife and jackknife+ predictive intervals for PCE-based regression models, without the need for repeated model retraining. Note that these closed-form relations are valid for PCEs with fixed polynomial basis. Should the basis be adaptively constructed, for example by means of sparsity-promoting algorithms [41, 42], explicit model refitting is required during the basis selection procedure. However, once the final basis is available, the analytical expressions remain applicable.

2.4 Normalized non-conformity scores

In the previous section, the non-conformity scores were equal to the absolute LOO residuals, i.e., $\alpha_m = \left|r_m^{\mathrm{LOO}}\right|$, $m = 1, \dots, M$. Based on the relative LOO error, defined as

$$\epsilon_{\mathrm{rel}}^{\mathrm{LOO}} = \frac{\epsilon^{\mathrm{LOO}}}{\mathrm{Var}\left(\mathrm{Y}\right)} = \frac{\frac{1}{M}\sum_{m=1}^{M}\left(r_m^{\mathrm{LOO}}\right)^2}{\mathrm{Var}\left(\mathrm{Y}\right)}, \tag{16}$$

where Var(Y) denotes the variance of the output, normalized non-conformity scores can be defined as

$$\alpha_m^{\mathrm{norm}} = \frac{\left|r_m^{\mathrm{LOO}}\right|}{\sqrt{\mathrm{Var(Y)}}}. \tag{17}$$

Then, the quantile of the CP interval is taken on these dimensionless scores and later rescaled as

$$q_{1-s}^{\mathrm{orig}} = q_{1-s}^{\mathrm{norm}}\sqrt{\mathrm{Var}(Y)}. \tag{18}$$

The main advantage of using normalized non-conformity scores is that they adapt to heteroskedastic data, thus offering prediction intervals that vary in width according to local uncertainty. The main downside is that an output variance estimate is required, which may be challenging. For a PCE model, the output variance can be estimated cost-effectively by post-processing its coefficients, such that

$$\mathrm{Var}(Y) \approx \sum_{\boldsymbol{\alpha}\in\Lambda\setminus\mathbf{0}} c_{\alpha}^2, \tag{19}$$

assuming an orthonormal polynomial basis. Note that this estimate is affected by the accuracy of the PCE. Alternatively, the ouput variance can be estimated from the experimental design, assuming that there exist enough data points to allow for a sufficiently accurate estimate.

3. Numerical experiments

The conformalized PCE method is first applied to four benchmark models that are commonly used for the validation of surrogate modeling methods (section 3.1). Then, the method is applied to two electrical engineering use-cases concerning the design of, first, a power module heat sink (section 3.2) and, second, a Stern-Gerlach electromagnet (section 3.3). Different conformalized PCE configurations are examined, depending on jackknife method, maximum polynomial degree, non-conformity score type and, most importantly, training data availability. The conformalized PCE models are assessed with respect to their ability to construct predictive intervals that provide the required coverage. Separate test datasets are used for that purpose. Each experiment is repeated for 100 random seeds to estimate coverage and predictive interval statistics.

3.1 Validation on benchmark models

A univariate model is used in section 3.1.1, while multivariate models with six, seven, and ten input dimensions are used in sections 3.1.2, 3.1.3, and 3.1.4, respectively. The multivariate models are taken from the "Virtual Library of Simulation Experiments: Test Functions and Datasets" by Surjanovic and Bingham [52]. For all benchmark models, the experimental design used to train the PCE consists of samples drawn randomly from the distribution of the input parameters. The target coverage is fixed at 95% and the test datasets consist of 10^4 random samples. A discussion based on the numerical results obtained from the benchmark models is available in section 3.1.5.

3.1.1 Meromorphic function

We consider the one-dimensional meromorphic function

$$\mu(x) = \frac{1}{a + bx}, \tag{20}$$

with $a = 1$ and $b = 0.5$. We assume that the input x takes values as uniformly distributed in $[-1, 1]$. The conformalized PCE is constructed with maximum polynomial degrees $P \in \{2, 3\}$. Since the input is one-dimensional, the polynomial basis consists of $P + 1$ terms. The experimental design has size $M = C\,(P + 1)^2$, where $C \in \{2, 3, 5, 10\}$ is an oversampling coefficient. Note that the quadratic scaling of the experimental design in relation to the polynomial basis is necessary to ensure a well conditioned least squares problem (4) in the one-dimensional input case [44].

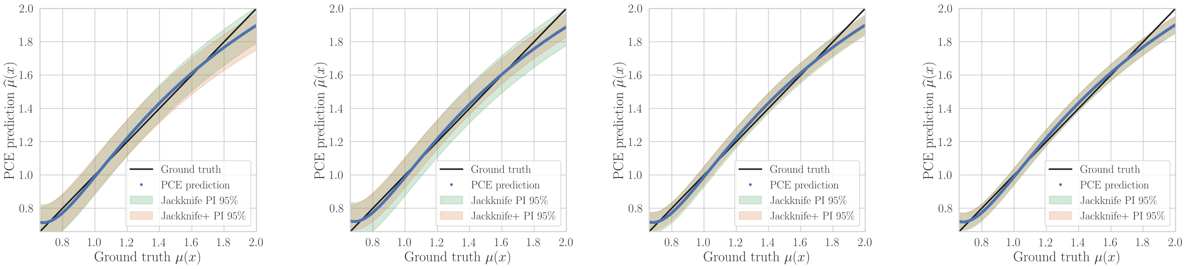
Figure 1 compares the different configurations of the conformalized PCE against the true meromorphic function for a single random seed. Figure 2 shows the empirical coverage results, where the box plots summarize the distribution of the empirical coverage over the 100 random seeds. Similarly, Figure 3 shows box plots regarding the corresponding predictive intervals. Note that, for this particular test-case, non-conformity score normalization has a negligible impact on coverage and predictive interval width, therefore only one set of results is presented.

3.1.2 OTL circuit function

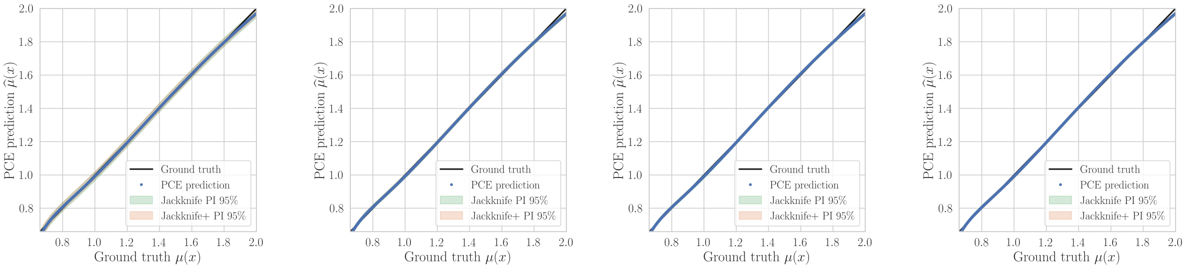
We consider the response of an output transformerless (OTL) push-pull circuit, given as

$$\mu(\mathbf{x}) = \frac{\left(\frac{12\,R_{b2}}{R_{b1}+R_{b2}} + 0.74\right)\beta\,(R_{c2}+9)}{\beta\,(R_{c2}+9)+R_f}, + \frac{11.35\,R_f}{\beta\,(R_{c2}+9)+R_f} + \frac{0.74\,R_f\,\beta\,(R_{c2}+9)}{(\beta\,(R_{c2}+9)+R_f)\,R_{c1}}, \tag{21}$$

where the input vector $\mathbf{x} \in \mathbb{R}^6$ consists of the resistances R_{b1}, R_{b2}, R_f, R_{c1}, R_{c2}, and the current gain β. The response corresponds to the midpoint voltage. The input parameters are assumed to be uniformly distributed within the value ranges given in Table 1.

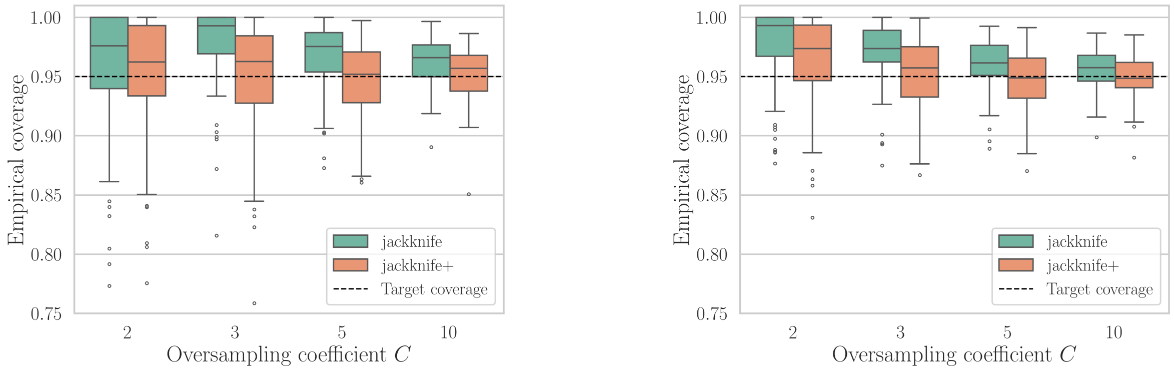


(a) $P = 2$, $C = 2$.
(b) $P = 2$, $C = 3$.
(c) $P = 2$, $C = 5$.
(d) $P = 2$, $C = 10$.



(e) $P = 3$, $C = 2$.
(f) $P = 3$, $C = 3$.
(g) $P = 3$, $C = 5$.
(h) $P = 3$, $C = 10$.

Figure 1: Parity plots comparing ground truth values of the meromorphic function against conformalized PCE predictions for different combinations of polynomial degree P and oversampling coefficient C. The results correspond to a single random seed. The results with and without non-conformity score normalization are identical.



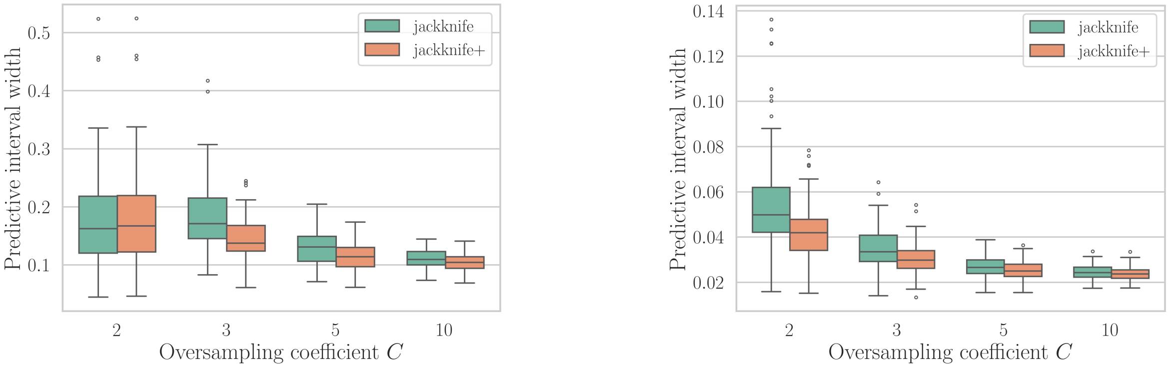
(a) $P = 2$.
(b) $P = 3$.

Figure 2: Box plots of the empirical coverage provided by conformalized PCE surrogates of the meromorphic function, for different combinations of polynomial degree P and oversampling coefficient C. The results with and without non-conformity score normalization are identical.

Table 1: Input parameters of the OTL circuit function.

Parameter	Units	Range
R_{b1}	kΩ	$[50, 150]$
R_{b2}	kΩ	$[25, 70]$
R_f	kΩ	$[0.5, 30]$
R_{c1}	kΩ	$[1.2, 2.50]$
R_{c2}	kΩ	$[0.25, 1.2]$
β	A	$[50, 300]$

Total-degree conformalized PCEs with maximum polynomial degrees $P \in \{1, 2, 3\}$ and oversampling coefficients $C \in \{2, 3, 5, 10\}$ are employed. Following Migliorati et al. [44], the experimental design now scales linearly with the size of the PCE basis, such that $M = CK = C\#\Lambda$. The different conformalized



(a) $P = 2$.

(b) $P = 3$.

Figure 3: Box plots of the predictive interval widths provided by conformalized PCE surrogates of the meromorphic function, for different combinations of polynomial degree P and oversampling coefficient C. The results with and without non-conformity score normalization are identical.

PCE configurations are compared against the true OTL circuit function for a single random seed in Figure 4. This plot does not compare normalized versus non-normalized conformity scores, as the differences between the corresponding predictive intervals are not easily discernible. Figures 5 and 6 shows box plots regarding the distributions of the conformalized PCEs' empirical coverages and predictive intervals over the 100 random seeds, also including the two different non-conformity scores.

3.1.3 Piston simulation function

We consider the circular motion of a piston within a cylinder, the cycle-time of which is given by the function

$$\mu(\mathbf{x}) = 2\pi \sqrt{\frac{M}{k + S^2 \frac{P_0 V_0}{T_0} \frac{T_0}{V^2}}}, \tag{22}$$

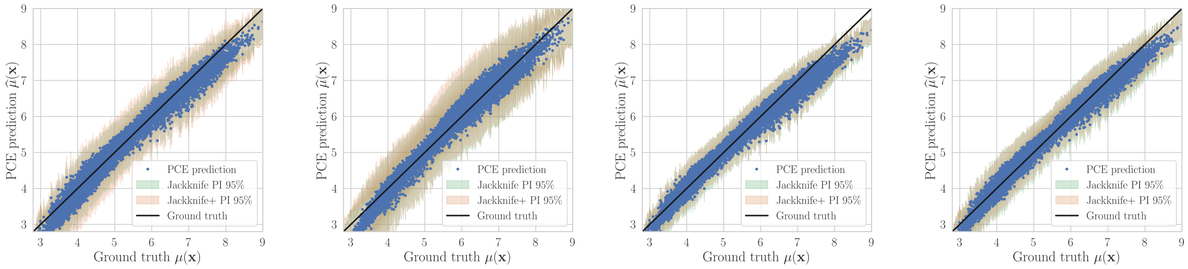
$$\text{where } V = \frac{S}{2k}\left(\sqrt{A^2 + 4k\frac{P_0 V_0}{T_0} T_a} - A\right) \text{ and } A = P_0\, S + 19.62\, M - \frac{k\, V_0}{S}.$$

The input vector $\mathbf{x} \in \mathbb{R}^7$ consists of the parameters listed in Table 2 along with their value ranges. All parameters are assumed to be uniformly distributed within their ranges.

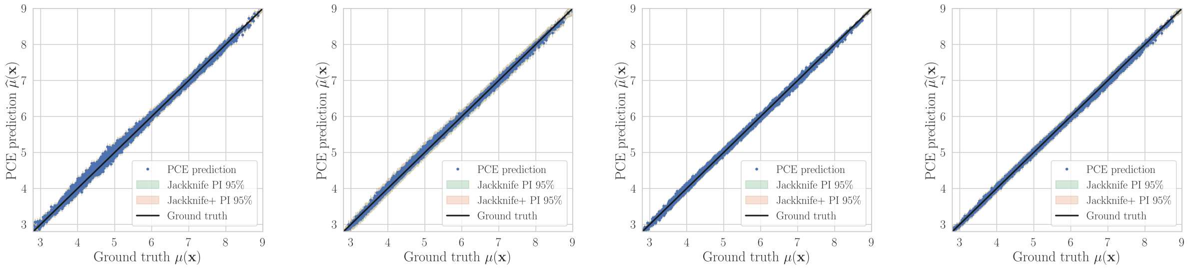
Table 2: Input parameters of the piston simulation function.

Parameter	Description	Units	Range
M	piston weight	kg	$[30, 60]$
S	piston surface area	m^2	$[0.005, 0.02]$
V_0	initial gas volume	m^3	$[0.002, 0.01]$
k	spring coefficient	N/m	$[1000, 5000]$
P_0	atmospheric pressure	$\mathrm{N/m}^2$	$[90000, 110000]$
T_a	ambient temperature	K	$[290, 296]$
T_0	filling gas temperature	K	$[340, 360]$

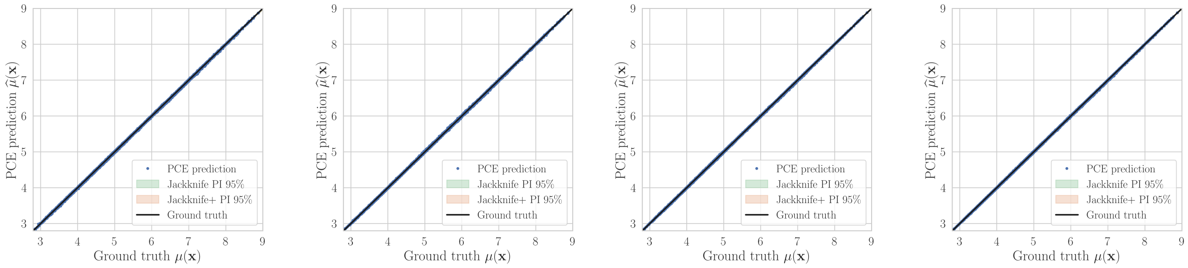
Total-degree conformalized PCEs with maximum polynomial degrees $P \in \{2, 3, 4\}$ and oversampling coefficients $C \in \{2, 3, 5, 10\}$ are employed. Similar to section 3.1.2, the experimental design scales linearly with the size of the PCE basis, i.e., $M = CK = C\#\Lambda$. Figure 7 compares the various configurations of the conformalized PCE against the true piston simulation function for a single random seed. Figures 8 and 9 show coverage and predictive interval statistics over the 100 random seeds.



(a) $P = 1$, $C = 2$. (b) $P = 1$, $C = 3$. (c) $P = 1$, $C = 5$. (d) $P = 1$, $C = 10$.



(e) $P = 2$, $C = 2$. (f) $P = 2$, $C = 3$. (g) $P = 2$, $C = 5$. (h) $P = 2$, $C = 10$.



(i) $P = 3$, $C = 2$. (j) $P = 3$, $C = 3$. (k) $P = 3$, $C = 5$. (l) $P = 3$, $C = 10$.

Figure 4: Parity plots comparing ground truth values of the OTL circuit function against conformalized PCE predictions for different combinations of polynomial degree P and oversampling coefficient C. The results correspond to a single random seed. The results with and without non-conformity score normalization are very similar, therefore, only one set of results in shown.

3.1.4 Wing weight function

We consider a function that models the weight of light aircraft wing, given as

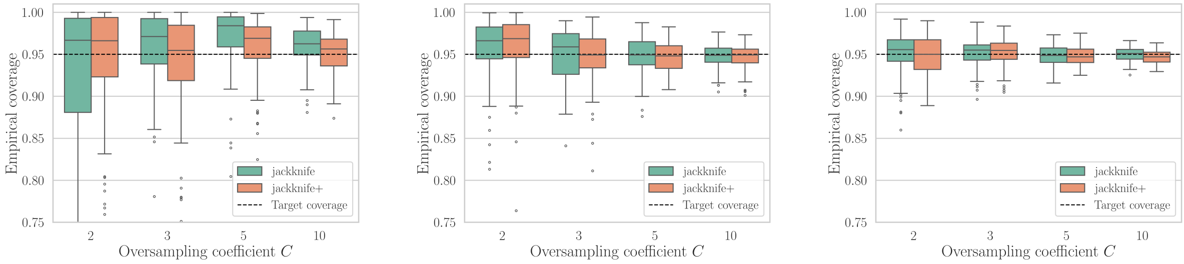
$$\mu(\mathbf{x}) = 0.036 S_{\mathrm{w}}^{0.758} W_{\mathrm{fw}}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100 t_{\mathrm{c}}}{\cos(\Lambda)}\right)^{-0.3} (N_{\mathrm{z}}\ W_{\mathrm{dg}})^{0.49} + S_{\mathrm{w}} W_{\mathrm{p}}, \tag{23}$$

where the input vector $\mathbf{x} \in \mathbb{R}^{10}$ consists of the parameters listed in Table 3, along with their value ranges. All parameters are assumed to be uniformly distributed within their ranges.

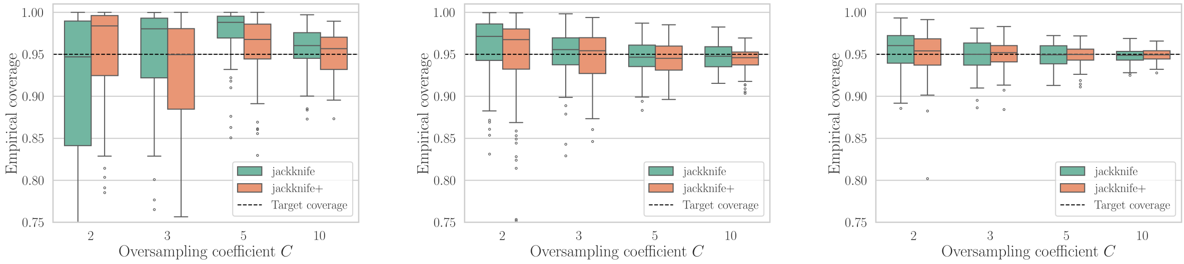
Total-degree conformalized PCEs with maximum polynomial degrees $P \in \{1, 2\}$ and oversampling coefficients $C \in \{2, 3, 5, 10\}$ are employed. The experimental design scales linearly with the size of the PCE basis, i.e., $M = CK = C\#\Lambda$. Figure 10 compares the different conformalized PCE configurations against the true wing weight function for a single random seed. Figures 8 and 9 present coverage and predictive interval statistics over the 100 random seeds.

3.1.5 Discussion on benchmarks' results

The numerical results presented in sections 3.1.1-3.1.4 show a number of patterns and similarities. First, the conformalized PCE is in general capable of providing empirical coverages close to the target level, more commonly above it. For less accurate PCEs, e.g., ones corresponding to low polynomial degrees and/or

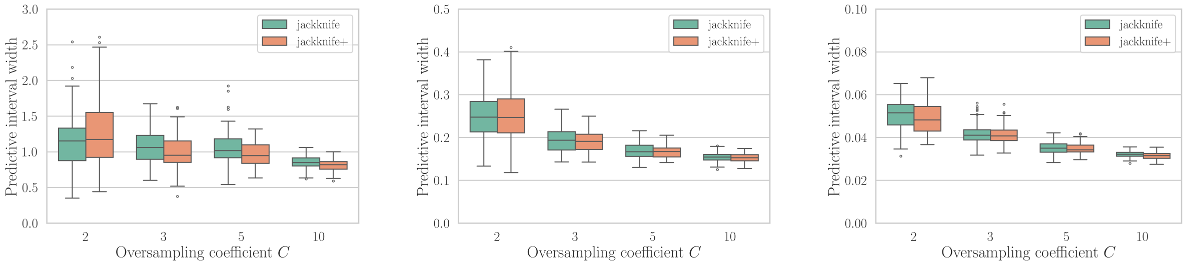


(a) $P = 1$, α_m. (b) $P = 2$, α_m. (c) $P = 3$, α_m.

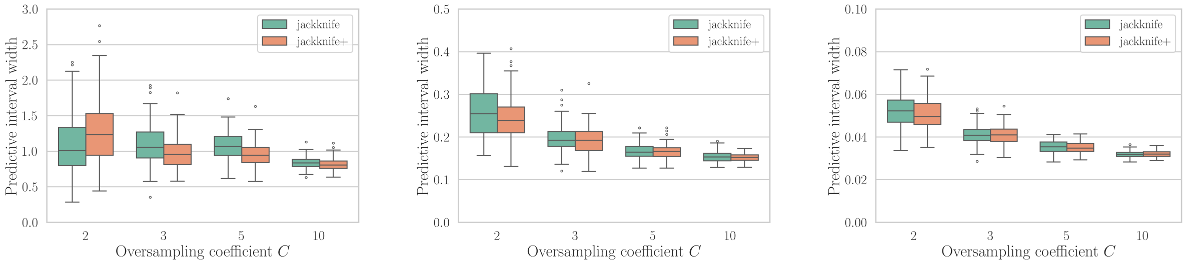


(d) $P = 1$, α_m^{norm}. (e) $P = 2$, α_m^{norm}. (f) $P = 3$, α_m^{norm}.

Figure 5: Box plots of the empirical coverage provided by conformalized PCE surrogates of the OTL circuit function, for different combinations of polynomial degree P, oversampling coefficient C, and non-conformity score type.



(a) $P = 1$, α_m. (b) $P = 2$, α_m. (c) $P = 3$, α_m.

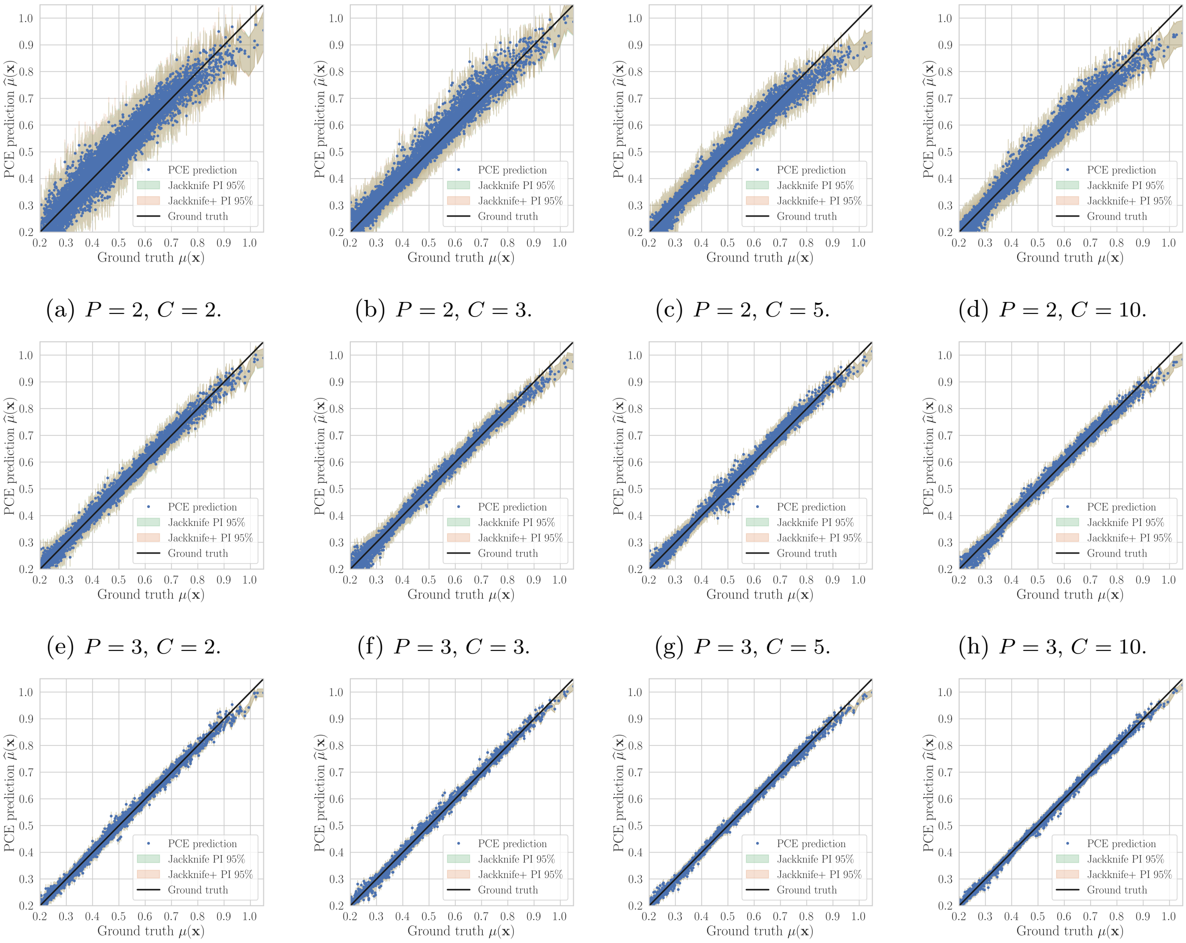


(d) $P = 1$, α_m^{norm}. (e) $P = 2$, α_m^{norm}. (f) $P = 3$, α_m^{norm}.

Figure 6: Box plots of the predictive interval widths provided by conformalized PCE surrogates of the OTL circuit function, for different combinations of polynomial degree P, oversampling coefficient C, and non-conformity score type.

small experimental designs, the coverage over the 100 random seeds varies significantly. As the accuracy of the PCE increases with larger experimental designs and higher polynomial degrees, the variability in coverage is reduced and the coverage moves closer to the target level. The size of the experimental design is particularly important, as a sufficiently training dataset enables even low-accuracy conformalized PCEs to reach the target coverages.

The coverage provided by the conformalized PCE is naturally connected to the corresponding predictive intervals. For the less accurate PCEs, the required coverage is accomplished by larger predictive



(a) $P = 2$, $C = 2$.

(b) $P = 2$, $C = 3$.

(c) $P = 2$, $C = 5$.

(d) $P = 2$, $C = 10$.

(e) $P = 3$, $C = 2$.

(f) $P = 3$, $C = 3$.

(g) $P = 3$, $C = 5$.

(h) $P = 3$, $C = 10$.

(i) $P = 4$, $C = 2$.

(j) $P = 4$, $C = 3$.

(k) $P = 4$, $C = 5$.

(l) $P = 4$, $C = 10$.

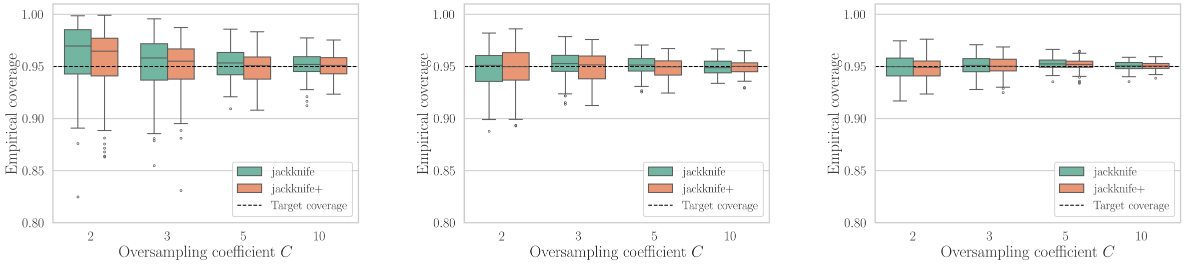
Figure 7: Parity plots comparing ground truth values of the piston simulation function against conformalized PCE predictions for different combinations of polynomial degree P and oversampling coefficient C. The results correspond to a single random seed. The results with and without non-conformity score normalization are very similar, therefore, only one set of results in shown.

Table 3: Input parameters of the wing weight function.

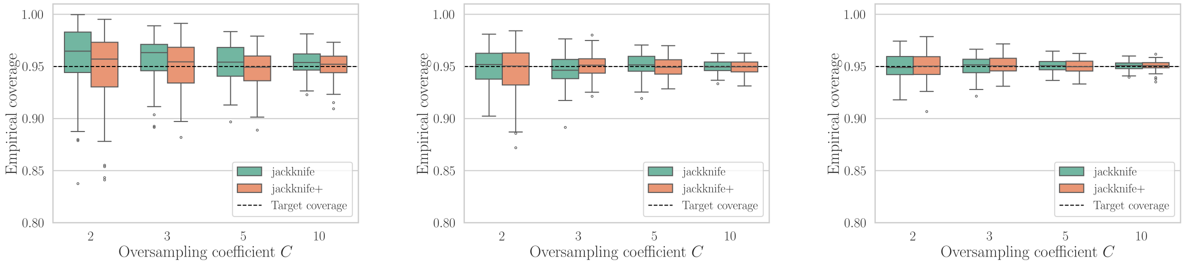
Parameter	Description	Units	Range
S_w	wing area	ft^2	$[150, 200]$
W_fw	wing fuel weight	lb	$[220, 300]$
A	aspect ratio	–	$[6, 10]$
Λ	quarter-chord sweep	$(^o)$	$[-10, 10]$
q	dynamic pressure at cruise	$\mathrm{lb/ft}^2$	$[16, 45]$
λ	taper ratio	–	$[0.5, 1]$
t_c	airfoil thickness to chord ratio	–	$[0.08, 0.18]$
N_z	ultimate load factor	–	$[2.5, 6]$
W_dg	flight design gross weight	lb	$[1700, 2500]$
W_p	paint weight	$\mathrm{lb/ft}^2$	$[0.025, 0.08]$

intervals, which also show a large variability over the 100 random seeds. As expected, the predictive intervals become smaller as the accuracy of the PCE increases. Importantly, even for low-accuracy PCEs, increasing the size of the experimental design leads not only to a reduced variability in the predictive interval, but also to a reduced interval width.

Comparing the jackknife and jackknife+ methods, their predictive intervals are in general very similar, especially for larger experimental designs and more accurate PCEs. The jackknife method is typically

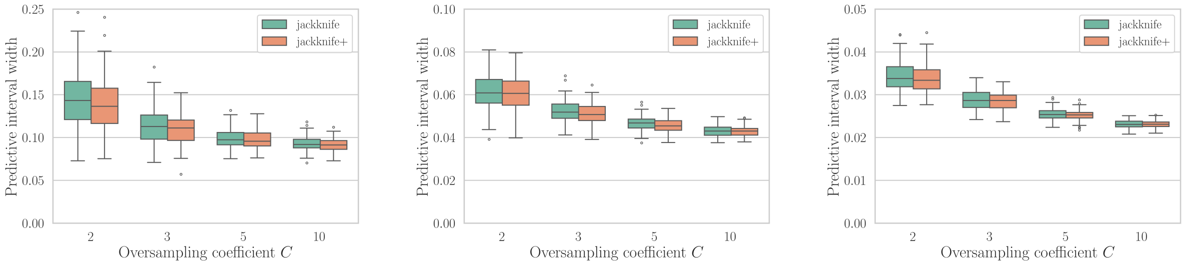


(a) $P = 2$, α_m. (b) $P = 3$, α_m. (c) $P = 4$, α_m.

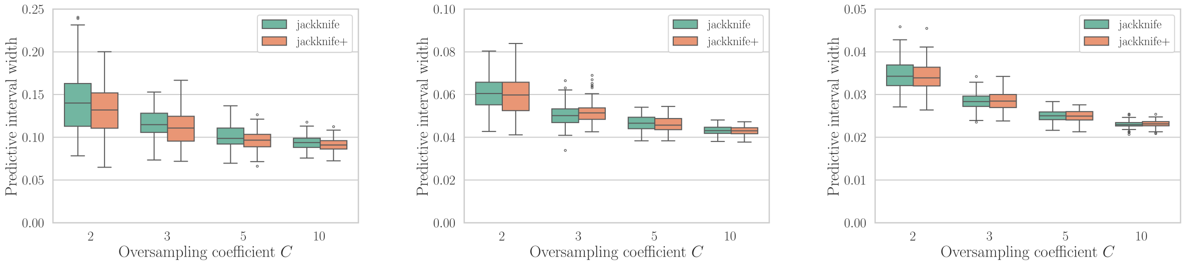


(d) $P = 2$, α_m^{norm}. (e) $P = 3$, α_m^{norm}. (f) $P = 4$, α_m^{norm}.

Figure 8: Box plots of the empirical coverage provided by conformalized PCE surrogates of the piston simulation function, for different combinations of polynomial degree P, oversampling coefficient C, and non-conformity score type.



(a) $P = 2$, α_m. (b) $P = 3$, α_m. (c) $P = 4$, α_m.

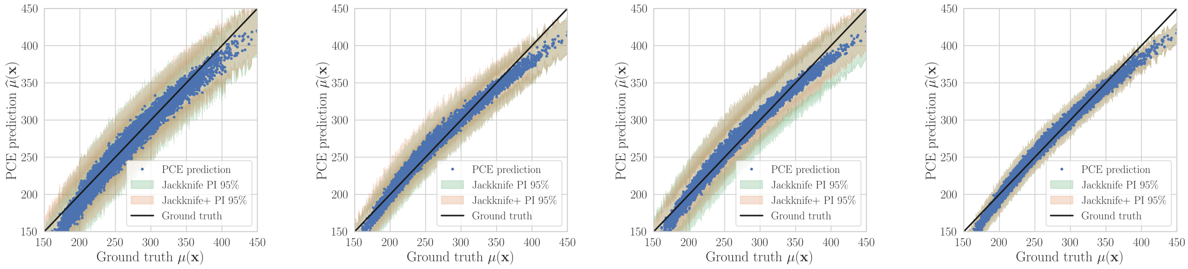


(d) $P = 2$, α_m^{norm}. (e) $P = 3$, α_m^{norm}. (f) $P = 4$, α_m^{norm}.

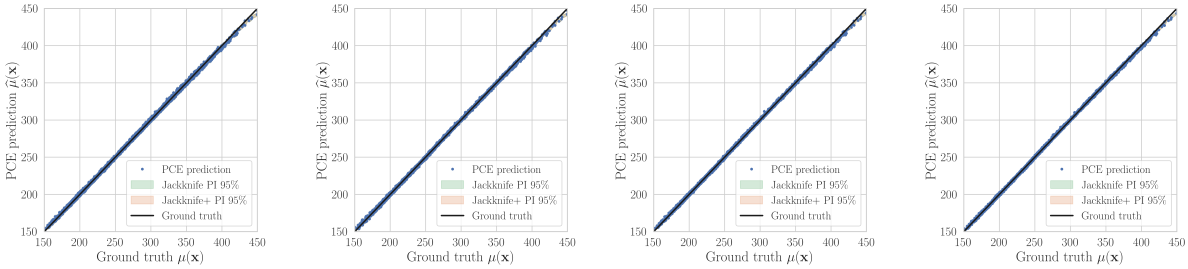
Figure 9: Box plots of the predictive interval widths provided by conformalized PCE surrogates of the piston simulation function, for different combinations of polynomial degree P, oversampling coefficient C, and non-conformity score type.

more conservative and results in comparatively larger intervals and to over-coverage, especially for smaller experimental designs. On the other hand, jackknife+ produces empirical coverages closer to the target level, in most cases. For both jackknife and jackknife+, it is clear that training data availability plays a significant role.

Last, comparing normalized and non-normalized non-conformity scores, no concrete conclusions can be made based on the available numerical results. There exist differences between the two, which are more common for PCEs trained with small experimental designs. However, these differences are generally

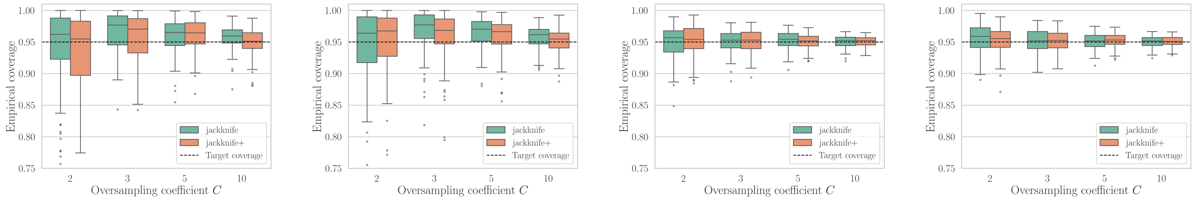


(a) $P = 1$, $C = 2$.
(b) $P = 1$, $C = 3$.
(c) $P = 1$, $C = 5$.
(d) $P = 1$, $C = 10$.



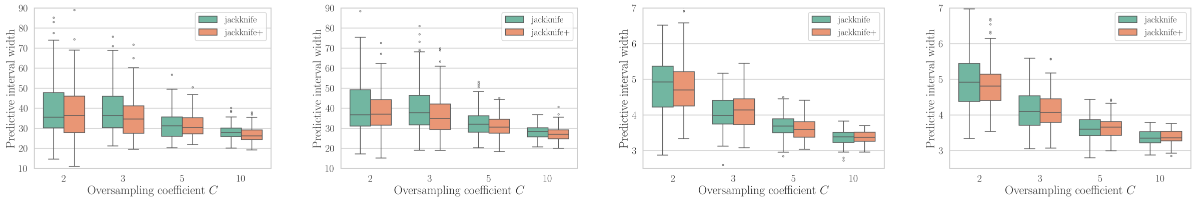
(e) $P = 2$, $C = 2$.
(f) $P = 2$, $C = 3$.
(g) $P = 2$, $C = 5$.
(h) $P = 2$, $C = 10$.

Figure 10: Parity plots comparing ground truth values of the wing weight function against conformalized PCE predictions for different combinations of polynomial degree P and oversampling coefficient C. The results correspond to a single random seed. The results with and without non-conformity score normalization are very similar, therefore, only one set of results in shown.



(a) $P = 1$, α_m.
(b) $P = 1$, α_m^{norm}.
(c) $P = 2$, α_m.
(d) $P = 2$, α_m^{norm}.

Figure 11: Box plots of the empirical coverage provided by conformalized PCE surrogates of the wing weight function, for different combinations of polynomial degree P, oversampling coefficient C, and non-conformity score type.



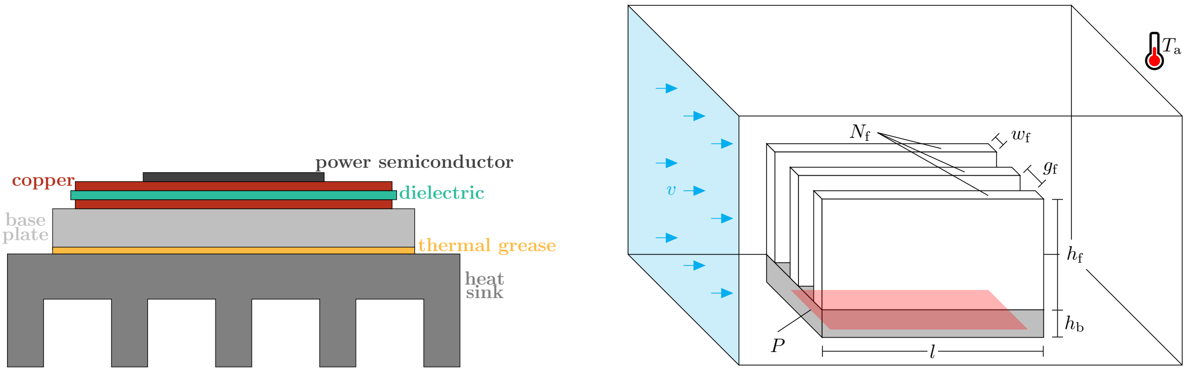
(a) $P = 1$, α_m.
(b) $P = 1$, α_m^{norm}.
(c) $P = 2$, α_m.
(d) $P = 2$, α_m^{norm}.

Figure 12: Box plots of the predictive interval widths provided by conformalized PCE surrogates of the wing weight function, for different combinations of polynomial degree P, oversampling coefficient C, and non-conformity score type.

small and not consistent. Overall, it can be said that the two non-conformity scores perform similarly.

3.2 Application to power module heat sink design

As first engineering use-case, we consider a power electronics design optimization problem concerning the design of a heat sink for cooling an insulated gate bipolar transistor (IGBT) power module [6]. The power



(a) Power electronic module.

(b) Heat sink simulation model.

Figure 13: Illustration of the power module and of the CFD heat sink simulation model.

module is illustrated in Figure 13a. The thermal behavior of the heat sink for various geometric designs and under different operating conditions is simulated using a 3D computational fluid dynamics (CFD) model. The model is based on the finite-volume discretization of the Reynolds-averaged Navier-Stokes equations [53] and utilizes the Boussinesq approximation for Reynold stress modeling [54] and the $k - \omega$ shear stress transport model for turbulence modeling [55]. It is implemented using the ANSYS Icepak® commercial software[1], where only the heat sink is explicitly modeled and the power module is represented only as a rectangular power source, as shown in Figure 13b. The model's nine input parameters are listed in Table 4, along with their value ranges. All parameters are assumed to be uniformly distributed within their value ranges, where N_{f} takes integer values only. The target output is the heat sink's thermal resistance, denoted as $R_{\mathrm{th,s}}$ and estimated as $R_{\mathrm{th,s}} = (T_{\mathrm{s}} - T_{\mathrm{a}})\,/P$ (in $\mathrm{K\,W^{-1}}$), where T_{s} denotes the heat sink's steady-state temperature [56].

Table 4: Input parameters of the power module heat-sink model.

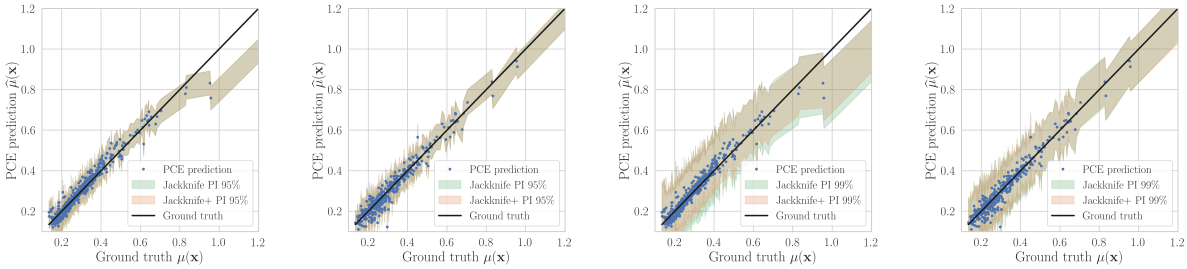
Parameter	Description	Unit	Range
l	heat sink length	mm	$[50, 200]$
g_{f}	fin gap	mm	$[3, 8]$
w_{f}	fin width	mm	$[1.4, 4]$
h_{f}	fin height	mm	$[16, 45]$
h_{b}	base height	mm	$[4, 15]$
N_{f}	number of fins	–	$[5, 25]$
v	air flow velocity	$\mathrm{m\,s^{-1}}$	$[1, 5]$
T_{a}	ambient temperature	°C	$[25, 45]$
P	power loss	W	$[115, 140]$

To enable computationally demanding design exploration studies such as heat sink design optimization [6], it is necessary to replace the CFD model with an inexpensive surrogate model. Here we are concerned with the surrogate modeling task only, with additional interest in the predictive uncertainty of the surrogate. Two significance levels are considered, namely, $s = 0.05$ and $s = 0.01$, respectively corresponding to target coverage levels of $1 - s = 0.95$ and $1 - s = 0.99$. The conformalized PCE has maximum polynomial degree $P \in \{2, 3\}$. A dataset consisting of 935 design configurations along with the corresponding thermal resistance values is readily available. Experimental designs of increasing size $M \in \{250, 300, \ldots, 500\}$ are employed, while $M' = 435$ data points are kept as test data. The initial dataset is reshuffled 100 times to estimate coverage and predictive interval statistics over the different dataset partitions.

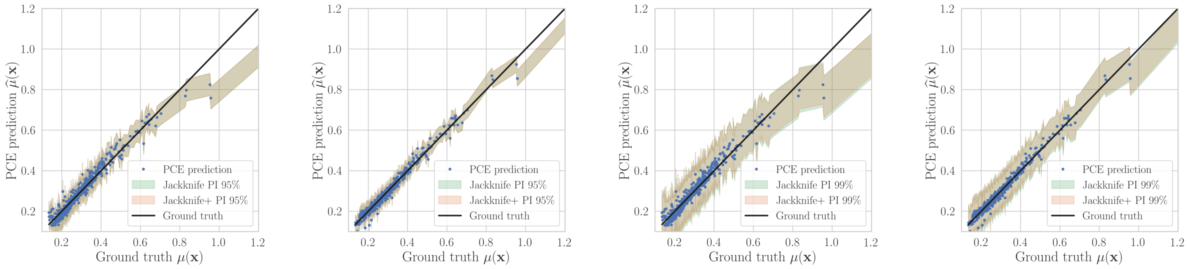
Figure 14 compares $R_{\mathrm{th,s}}$ predictions provided by the conformalized PCE against the true CFD model

---

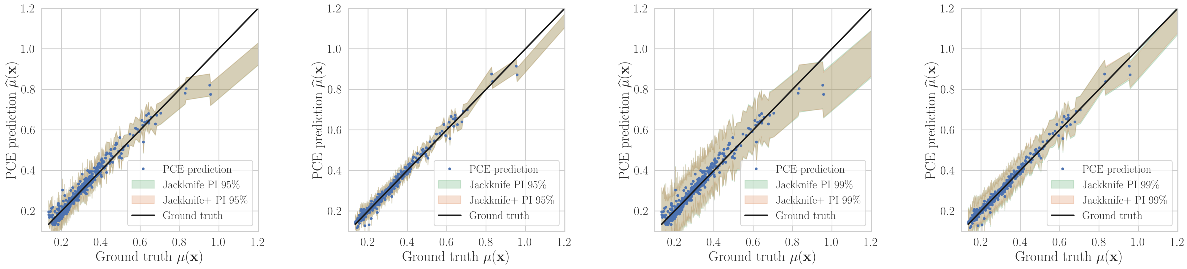
[1] www.ansys.com/products/electronics/ansys-icepak



(a) $P = 2,\ M = 300,\ s = 0.05$. (b) $P = 3,\ M = 300,\ s = 0.05$. (c) $P = 2,\ M = 300,\ s = 0.01$. (d) $P = 3,\ M = 300,\ s = 0.01$.

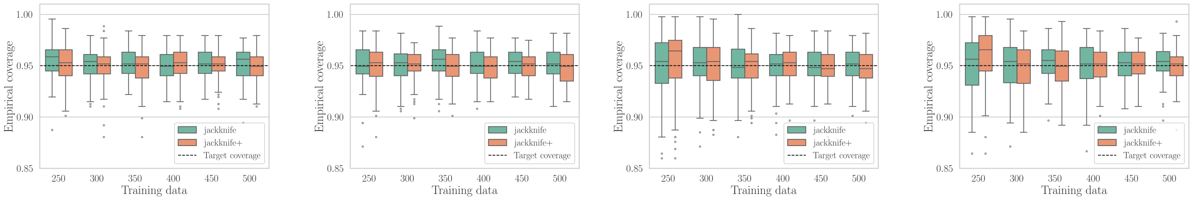


(e) $P = 2,\ M = 400,\ s = 0.05$. (f) $P = 3,\ M = 400,\ s = 0.05$. (g) $P = 2,\ M = 400,\ s = 0.01$. (h) $P = 3,\ M = 400,\ s = 0.01$.

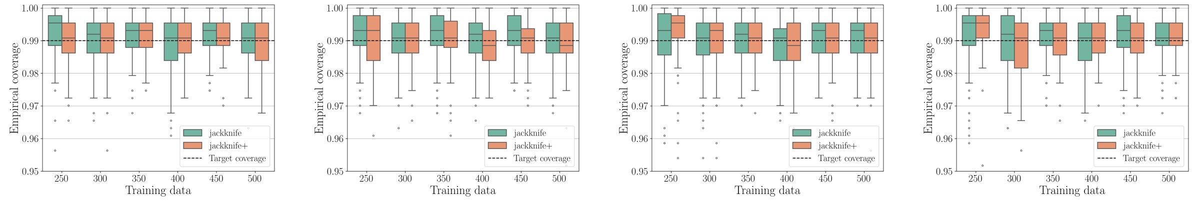


(i) $P = 2,\ M = 500,\ s = 0.05$. (j) $P = 3,\ M = 500,\ s = 0.05$. (k) $P = 2,\ M = 500,\ s = 0.01$. (l) $P = 3,\ M = 500,\ s = 0.01$.

Figure 14: Parity plots comparing ground truth $R_{\mathrm{th,s}}$ values given by the heat-sink simulation model against conformalized PCE predictions for different combinations of polynomial degree P, training dataset size M and significance level s. The results correspond to a single partition of the available design dataset. The results with and without non-conformity score normalization are very similar, therefore, only one set of results in shown.

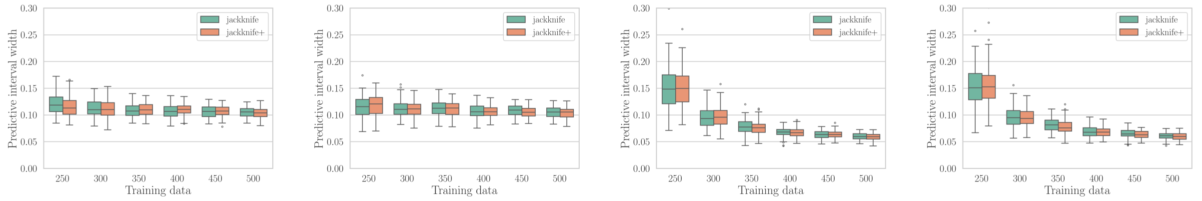


(a) $P = 2,\ \alpha_m,\ s = 0.05$. (b) $P = 2,\ \alpha_m^{\mathrm{norm}},\ s = 0.05$. (c) $P = 3,\ \alpha_m,\ s = 0.05$. (d) $P = 3,\ \alpha_m^{\mathrm{norm}},\ s = 0.05$.

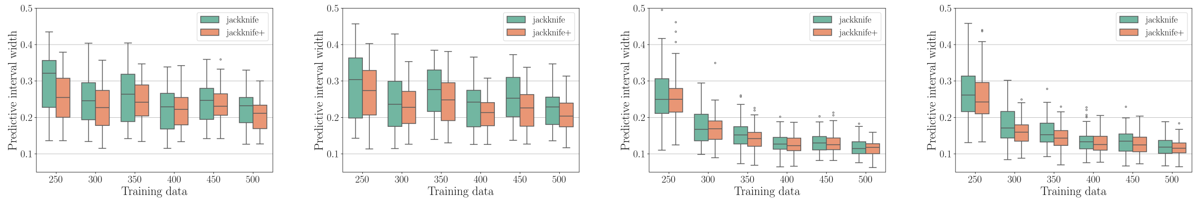


(e) $P = 2,\ \alpha_m,\ s = 0.01$. (f) $P = 2,\ \alpha_m^{\mathrm{norm}},\ s = 0.01$. (g) $P = 3,\ \alpha_m,\ s = 0.01$. (h) $P = 3,\ \alpha_m^{\mathrm{norm}},\ s = 0.01$.

Figure 15: Box plots of the empirical coverage provided by conformalized PCE surrogates of the heat-sink model, for different combinations of polynomial degree P, training dataset size M, significance level s, and non-conformity score type.



(a) $P = 2$, α_m, $s = 0.05$. (b) $P = 2$, α_m^{norm}, $s = 0.05$. (c) $P = 3$, α_m, $s = 0.05$. (d) $P = 3$, α_m^{norm}, $s = 0.05$.



(e) $P = 2$, α_m, $s = 0.01$. (f) $P = 2$, α_m^{norm}, $s = 0.01$. (g) $P = 3$, α_m, $s = 0.01$. (h) $P = 3$, α_m^{norm}, $s = 0.01$.

Figure 16: Box plots of the predictive interval widths provided by conformalized PCE surrogates of the heat-sink model, for different combinations of training dataset size M, significance level s, and non-conformity score type.

outputs for training dataset sizes $M \in \{300, 400, 500\}$. Only the original dataset is used for these plots, without data reshuffling. Figures 15 and 16 show coverage and predictive interval statistics over the 100 different dataset partitions and for all considered training dataset sizes. The empirical coverages provided by the conformalized PCEs are typically on par or slightly above the target levels. Jackknife CP is again more conservative compared to jackknife+, especially for the more demanding significance level $s = 0.01$. Nonetheless, both approaches perform very similarly as the training dataset size increases. This is also reflected on the predictive interval widths, where, as would be expected, the significance level $s = 0.01$ results in wider interval widths. The use of normalized non-conformity scores results in differences regarding coverage and predictive interval width. However, there is no consistent benefit or drawback in using one instead of the other.

3.3 Application to electromagnet design

As second engineering use-case, we consider a so-called Stern-Gerlach electromagnet, typically employed for the magnetic separation of atom beams [39]. A 3D model of the magnet is depicted in Figure 17a, where half of the geometry is shown. This model is generated with the CST Studio Suite® commercial software[2]. A key design requirement of such magnets is a magnetic field with a homogeneous and strong gradient. Field homogeneity is particularly important in the magnet's beam area which is included in the pole region of the magnet, as shown in Figure 17b.

The target model output is the average magnetic field gradient in the beam area, denoted as $\overline{\tau}_{\text{b}}$ and measured in $\text{T}\,\text{m}^{-1}$. For computational efficiency, $\overline{\tau}_{\text{b}}$ is computed with a 2D isogeometric analysis (IGA) model of the pole region, while the coils and yoke of the original 3D model are replaced by equivalent circuit models [57]. Then, $\overline{\tau}_{\text{b}}$ is estimated as

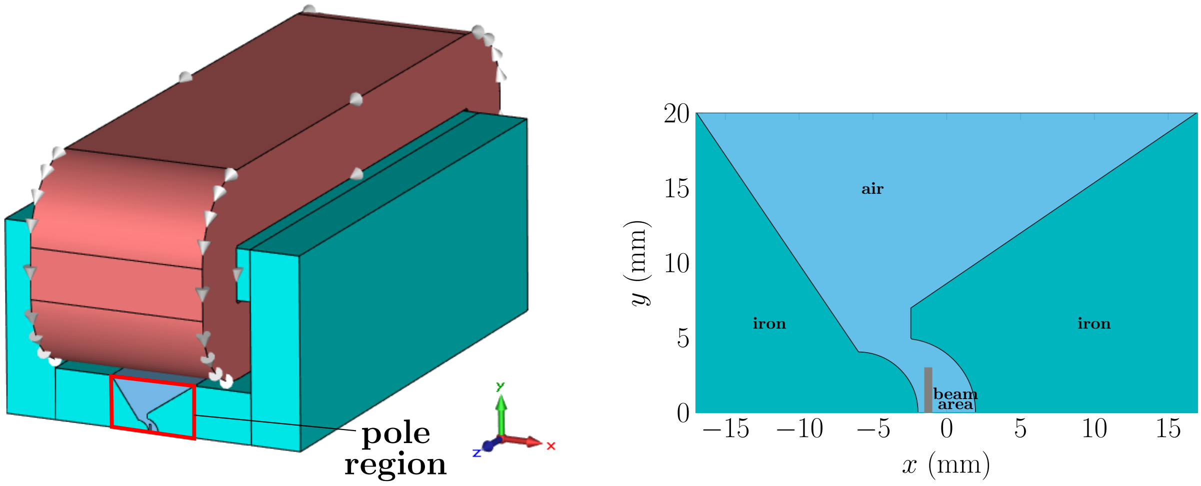
$$\overline{\tau}_{\text{b}} = \frac{1}{\Omega_{\text{b}}} \int_{\Omega_{\text{b}}} \tau(x, y) \mathrm{d}\Omega, \tag{24}$$

where Ω denotes the full 2D computational domain, Ω_{b} corresponds to the beam area, and $\tau(x, y) = \frac{\partial |\vec{B}|}{\partial x}$ is the magnetic field gradient in the x-direction, where $\vec{B}$ denotes the magnetic flux density. The latter is obtained by the IGA model's field solution.

While much more efficient compared to the original 3D model, the circuit-coupled IGA model is still

---

[2] www.3ds.com/products/simulia/cst-studio-suite



(a) Stern-Gerlach magnet model (half geometry).

(b) Pole region geometry.

Figure 17: Illustration of the Stern-Gerlach magnet model.

too computationally expensive for design optimization [39, 57] or UQ [58] studies, hence the need for a surrogate model. Here, the surrogate model is used to approximate $\overline{\tau}_{\mathrm{b}}$ under random shape deformations in the pole region. The geometry of the pole region is described using non-uniform B-splines (NURBS) $\mathbf{R}(\xi)$, $\xi \in [0,1]$, defined by control points $\mathbf{P}_i$, $i = 1, \dots, N_{\mathrm{cp}}$, given as

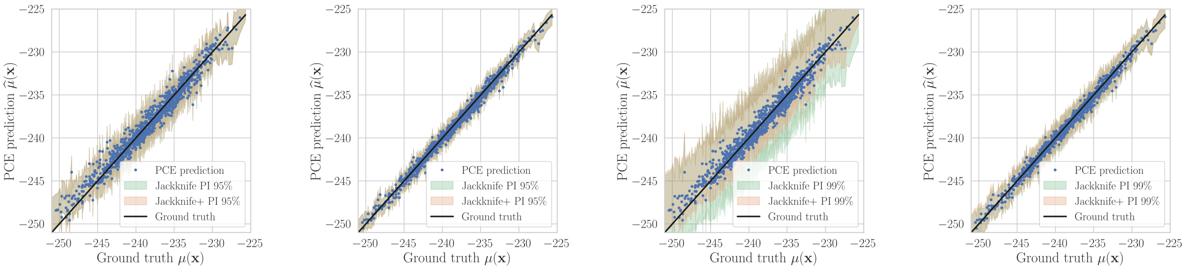
$$\mathbf{R}(\xi) = \sum_{i=1}^{N_{\mathrm{cp}}} \mathbf{P}_i \frac{w_i B_i^p(\xi)}{\sum_{j=1}^{N_{\mathrm{cp}}} w_j B_j^p(\xi)}, \tag{25}$$

where $B_i^p(\xi)$ are B-spline basis functions of order p and w_i are their corresponding weights. Random shape deformations are introduced by modeling the coordinates of five control points and four of the corresponding weights as uniform random variables. All input parameters and their ranges are listed in Table 5.

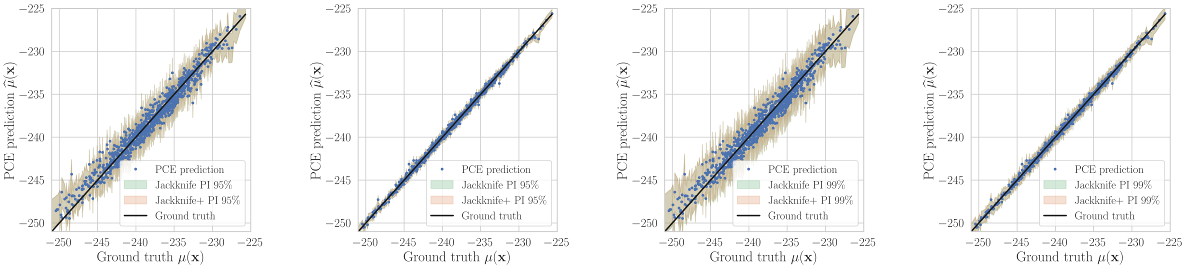
Two significance levels $s = 0.05$ and $s = 0.01$ are considered regarding the predictive uncertainty of the surrogate model. The conformalized PCE is applied with maximum polynomial degrees $P \in \{1, 2\}$. A dataset with 1000 different shape deformations of the pole region is available, along with the corresponding $\overline{\tau}_{\mathrm{b}}$ vaues. Training datasets of increasing size $M \in \{150, 200, \dots, 500\}$ are employed. The remaining $M' = 500$ data points are used for validation. The dataset is shuffled 100 times to estimate coverage and

Table 5: Input parameters of the Stern-Gerlach magnet model.

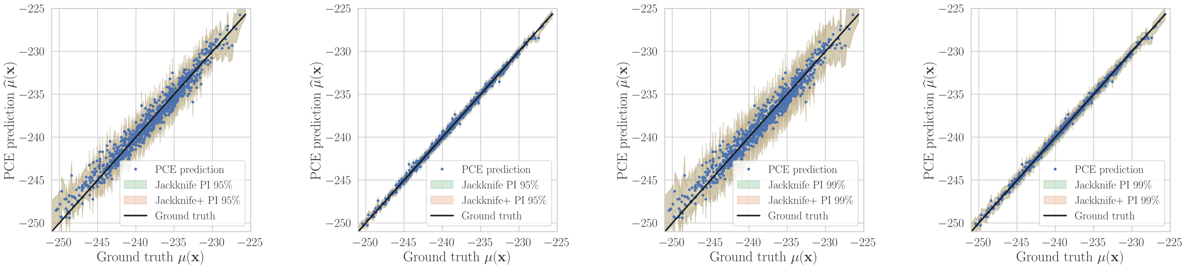
Parameter	Description	Unit	Range
x_1	x-coordinate, control point 1	mm	$[-3.38, -1.38]$
y_1	y-coordinate, control point 1	mm	$[5.96, 7.96]$
x_2	x-coordinate, control point 2	mm	$[-3.38, -1.38]$
y_2	y-coordinate, control point 2	mm	$[3.96, 5.96]$
x_3	x-coordinate, control point 3	mm	$[16, 18]$
y_3	y-coordinate, control point 3	mm	$[19, 21]$
x_4	x-coordinate, control point 4	mm	$[-18, -16]$
y_4	y-coordinate, control point 4	mm	$[19, 21]$
x_5	x-coordinate, control point 5	mm	$[-7, -5]$
y_5	y-coordinate, control point 5	mm	$[3, 2]$
w_1	weight, control point 1	–	$[0, 1]$
w_2	weight, control point 2	–	$[0, 1]$
w_3	weight, control point 3	–	$[0, 1]$
w_4	weight, control point 4	–	$[0, 1]$



(a) $P = 1,\ M = 150,\ s = 0.05.$ (b) $P = 2,\ M = 150,\ s = 0.05.$ (c) $P = 1,\ M = 150,\ s = 0.01.$ (d) $P = 2,\ M = 150,\ s = 0.01.$



(e) $P = 1,\ M = 300,\ s = 0.05.$ (f) $P = 2,\ M = 300,\ s = 0.05.$ (g) $P = 1,\ M = 300,\ s = 0.01.$ (h) $P = 2,\ M = 300,\ s = 0.01.$



(i) $P = 1,\ M = 450,\ s = 0.05.$ (j) $P = 2,\ M = 450,\ s = 0.05.$ (k) $P = 1,\ M = 450,\ s = 0.01.$ (l) $P = 2,\ M = 450,\ s = 0.01.$

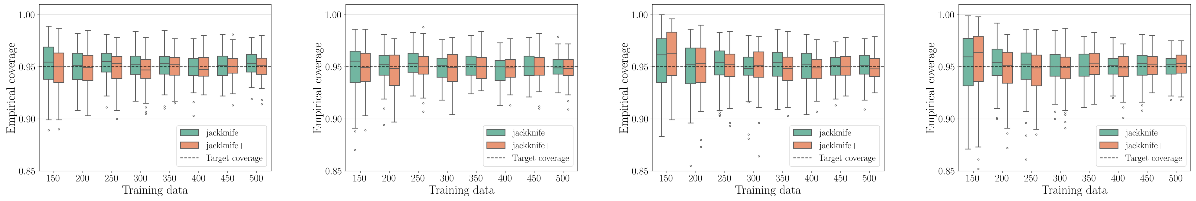
Figure 18: Parity plots comparing ground truth $\overline{\tau}_{\mathrm{b}}$ values given by the Stern-Gerlach magnet model against conformalized PCE predictions for different combinations of polynomial degree P, training dataset size M and significance level s. The results correspond to a single partition of the available design dataset. The results with and without non-conformity score normalization are very similar, therefore, only one set of results in shown.

predictive interval statistics.

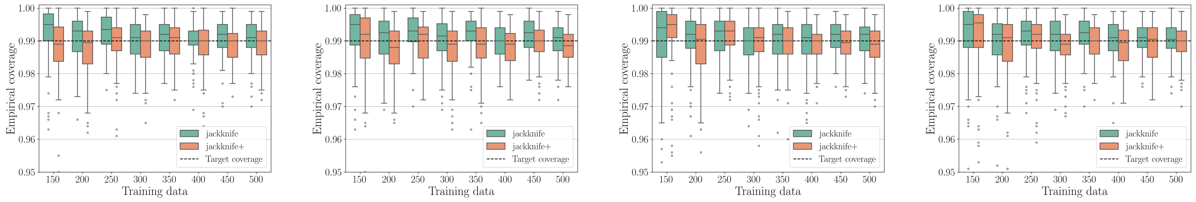
Figure 18 compares ground truth $\overline{\tau}_{\mathrm{b}}$ values provided by the IGA model against conformalized PCE predictions for the original dataset without reshuffling. Training dataset sizes $M \in \{150, 300, 450\}$ are considered in this figure. Figures 19 and 20 show boxplots that summarize the statistics concerning empirical coverage and predictive interval width over the 100 dataset reshuffles and for all considered training dataset sizes. The conformalized PCE provides the required coverage, even for the inaccurate $P = 1$ models and the more demanding significance level $s = 0.01$. As would be expected, the predictive intervals become narrower for $P = 2$, while still preserving the required coverage. Similar to the previous test-cases, the empirical coverages and the predictive interval widths present high variations for smaller experimental designs that progressively stabilize as more training data become available. The jackknife method is once again the more conservative option, while non-conformity score normalization does not seem to have a significant impact.

4. Summary, conclusions, and outlook

This work presented a new method that combines regression-based PCEs with jackknife-based CP for constructing uncertainty-aware, data-driven surrogate models. Using this method, data-driven PCEs are

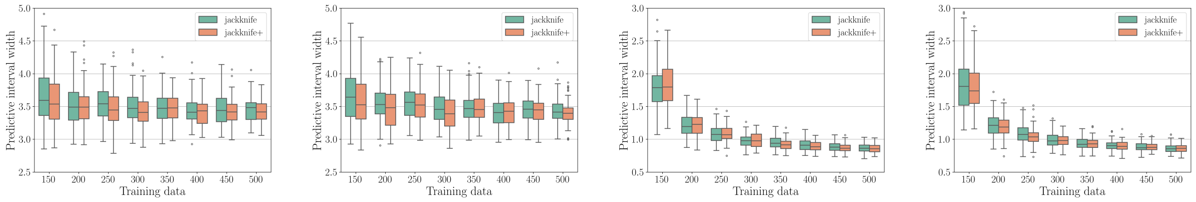


(a) $P = 1$, α_m, $s = 0.05$. (b) $P = 1$, α_m^{norm}, $s = 0.05$. (c) $P = 2$, α_m, $s = 0.05$. (d) $P = 2$, α_m^{norm}, $s = 0.05$.

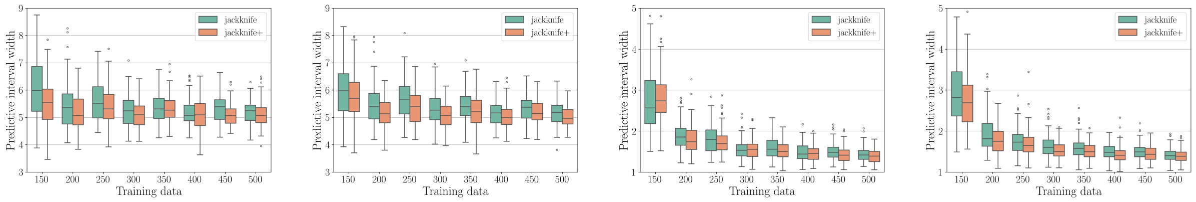


(e) $P = 1$, α_m, $s = 0.01$. (f) $P = 1$, α_m^{norm}, $s = 0.01$. (g) $P = 2$, α_m, $s = 0.01$. (h) $P = 2$, α_m^{norm}, $s = 0.01$.

Figure 19: Box plots of the empirical coverage provided by conformalized PCE surrogates of the Stern-Gerlach magnet model, for different combinations of polynomial degree P, training dataset size M, significance level s, and non-conformity score type.



(a) $P = 1$, α_m, $s = 0.05$. (b) $P = 1$, α_m^{norm}, $s = 0.05$. (c) $P = 2$, α_m, $s = 0.05$. (d) $P = 2$, α_m^{norm}, $s = 0.05$.



(e) $P = 1$, α_m, $s = 0.01$. (f) $P = 1$, α_m^{norm}, $s = 0.01$. (g) $P = 2$, α_m, $s = 0.01$. (h) $P = 2$, α_m^{norm}, $s = 0.01$.

Figure 20: Box plots of the predictive interval widths provided by conformalized PCE surrogates of the Stern-Gerlach magnet model, for different combinations of training dataset size M, significance level s, and non-conformity score type.

complemented with predictive intervals that quantify the uncertainty in their predictions. The predictive intervals are estimated with very low computational overhead, by leveraging the linearity of the PCE regression model. In particular, analytical, closed-form expressions can be used for computing the LOO residuals and LOO predictions employed by the jackknife and jackknife+ CP methods for predictive interval estimation.

The conformalized PCE method was demonstrated on four benchmark models and two electrical engineering design applications concerning a power electronics heat sink and a high-gradient-field electromagnet. Different conformalized PCE configurations were examined, based on maximum polynomial degree, experimental design size, CP method (jackknife/jackknife+), and non-conformity score type (normalized/non-normalized). Based on the numerical results, the conformalized PCE is in general able to provide empirical coverages close to the required target level. Training data availability, equivalently, experimental design size is a most important factor for providing the required coverage with the minimum possible interval width. Moreover, considering PCE models trained with different datasets, larger training datasets result in reduced variations in coverage and predictive interval width. The jackknife method typically results in more conservative predictive uncertainty estimates, with wider intervals and empirical coverages above the target level. The jackknife+ method yields coverages closer to the target level. Non-conformity score normalization does not seem to have a significant or consistent impact on predictive interval width and coverage, at least for the examined benchmark models and engineering use-cases.

The conformalized PCE method developed in this work shows definite promise as uncertainty-aware, data-driven, surrogate modeling method. Nonetheless, it can be improved in several aspects. First, this work has considered total-degree PCEs only, thus limiting the selected benchmark models and engineering use-cases to low dimensions and/or polynomial degrees. The integration of CP within sparse PCE algorithms [41, 42] would allow to extend the method to more challenging problem settings. Second, in its current form, the conformalized PCE method is applicable to scalar outputs only. In principle, it can be applied elementwise to address multidimensional outputs. However, this would be a suboptimal approach, as it does not account for correlations among the output's elements and would incur high computational costs with increasing output dimension. The development of a conformalized PCE method suitable for multivariate outputs seems to be a natural follow-up, for example based on the combination of multi-output PCEs [59–61] and multi-output CP [62, 63]. We aim to address these limitations in dedicated follow-up works.

Acknowledgements

Dimitris G. Giovanis was supported by the Department of Energy (DOE) under Grant No. DE-SC0024162.

Author contributions

Dimitrios Loukrezis: Conceptualization, Methodology, Software, Validation, Data curation, Writing - original draft, Writing - review & editing.
Dimitris G. Giovanis: Validation, Data curation, Writing - original draft, Writing - review & editing.

Appendices

A. Leave-one-out residuals in closed form

Let $\widehat{\mu}(\mathbf{x})$ denote a PCE regression model as in section 2.1. Removing the m-th sample from the training data, we get the LOO design matrix $\mathbf{D}_{\sim m} \in \mathbb{R}^{(M-1)\times K}$ and the LOO output vector $\mathbf{y}_{\sim m} \in \mathbb{R}^{M-1}$. The omitted m-th row of the original design matrix $\mathbf{D}$ is denoted with $\mathbf{d}_m^\top \in \mathbb{R}^{1\times K}$, where $\mathbf{d}_m^\top = \left(\Psi_1\left(\mathbf{x}^{(m)}\right), \dots, \Psi_K\left(\mathbf{x}^{(m)}\right)\right)$. Then, the LOO equivalent to the least-squares solution (5) is

$$\mathbf{c}_{\sim m} = \left(\mathbf{D}_{\sim m}^\top \mathbf{D}_{\sim m}\right)^{-1} \mathbf{D}_{\sim m}^\top \mathbf{y}_{\sim m}, \tag{26}$$

where it also holds that

$$\mathbf{D}_{\sim m}^\top \mathbf{D}_{\sim m} = \mathbf{D}^\top \mathbf{D} - \mathbf{d}_m \mathbf{d}_m^\top, \tag{27}$$

$$\mathbf{D}_{\sim m}^\top \mathbf{y}_{\sim m} = \mathbf{D}^\top \mathbf{y} - \mathbf{d}_m y^{(m)}. \tag{28}$$

Let $\mathbf{A} = \left(\mathbf{D}^\top \mathbf{D}\right)^{-1} \in \mathbb{R}^{K\times K}$. Using the Sherman-Morrison formula we get

$$\left(\mathbf{D}^\top \mathbf{D} - \mathbf{d}_m \mathbf{d}_m^\top\right)^{-1} = \mathbf{A} + \frac{\mathbf{A}\mathbf{d}_m \mathbf{d}_m^\top \mathbf{A}}{1 - \mathbf{d}_m^\top \mathbf{A} \mathbf{d}_m}, \tag{29}$$

assuming that $1 - \mathbf{d}_m^\top \mathbf{A} \mathbf{d}_m \neq 0$. Note that $\mathbf{d}_m^\top \mathbf{A} \mathbf{d}_m = h_{mm}$, where h_{mm} are the diagonal elements of the hat matrix $\mathbf{H} \in \mathbb{R}^{M \times M}$ defined in (13). Using expressions (27), (28), and (29), expression (26) can now be written as

$$\begin{aligned}
\mathbf{c}_{\sim m} &= \left(\mathbf{A} + \frac{\mathbf{A}\mathbf{d}_m \mathbf{d}_m^\top \mathbf{A}}{1 - h_{mm}}\right) \left(\mathbf{D}^\top \mathbf{y} - \mathbf{d}_m y^{(m)}\right) \\
&= \mathbf{A}\mathbf{D}^\top \mathbf{y} - \mathbf{A}\mathbf{d}_m y^{(m)} + \frac{\mathbf{A}\mathbf{d}_m \mathbf{d}_m^\top \mathbf{A}\mathbf{D}^\top \mathbf{y}}{1 - h_{mm}} - \frac{\mathbf{A}\mathbf{d}_m \mathbf{d}_m^\top \mathbf{A}\mathbf{d}_m y^{(m)}}{1 - h_{mm}} \\
&= \mathbf{c} - \mathbf{A}\mathbf{d}_m y^{(m)} + \frac{\mathbf{A}\mathbf{d}_m \mathbf{d}_m^\top \mathbf{c}}{1 - h_{mm}} - \frac{\mathbf{A}\mathbf{d}_m h_{mm} y^{(m)}}{1 - h_{mm}} \\
&= \mathbf{c} + \mathbf{A}\mathbf{d}_m \left(\frac{\mathbf{d}_m^\top \mathbf{c} - h_{mm} y^{(m)}}{1 - h_{mm}} - y^{(m)}\right) \\
&= \mathbf{c} + \mathbf{A}\mathbf{d}_m \frac{\mathbf{d}_m^\top \mathbf{c} - y^{(m)}}{1 - h_{mm}} \\
&= \mathbf{c} + \mathbf{A}\mathbf{d}_m \frac{\widehat{y}^{(m)} - y^{(m)}}{1 - h_{mm}} \\
&= \mathbf{c} - \mathbf{A}\mathbf{d}_m \frac{y^{(m)} - \widehat{y}^{(m)}}{1 - h_{mm}},
\end{aligned} \tag{30}$$

where we have used that $\mathbf{c} = \mathbf{A}\mathbf{D}^\top \mathbf{y}$ and $\widehat{y}^{(m)} = \widehat{\mu}\left(\mathbf{x}^{(m)}\right) = \mathbf{d}_m^\top \mathbf{c}$. The LOO model evaluation at $\mathbf{x}^{(m)}$ is then given by

$$\widehat{\mu}_{\sim m}\left(\mathbf{x}^{(m)}\right) = \mathbf{d}_m^\top \mathbf{c}_{\sim m} = \mathbf{d}_m^\top \mathbf{c} - \mathbf{d}_m^\top \mathbf{A}\mathbf{d}_m \frac{y^{(m)} - \widehat{y}^{(m)}}{1 - h_{mm}} = \widehat{y}^{(m)} - h_{mm} \frac{y^{(m)} - \widehat{y}^{(m)}}{1 - h_{mm}}. \tag{31}$$

Therefore, the LOO residual defined in (10) is given by

$$\begin{aligned}
r_m^{\mathrm{LOO}} &= y^{(m)} - \widehat{\mu}_{\sim m}\left(\mathbf{x}^{(m)}\right) \\
&= y^{(m)} - \widehat{y}^{(m)} + h_{mm} \frac{\widehat{y}^{(m)} - y^{(m)}}{1 - h_{mm}} \\
&= \frac{y^{(m)} - \widehat{y}^{(m)}}{1 - h_{mm}} \\
&= \frac{y^{(m)} - \widehat{\mu}\left(\mathbf{x}^{(m)}\right)}{1 - h_{mm}}.
\end{aligned} \tag{32}$$

B. Leave-one-out predictions in closed form

Using similar notation as in appendix A, the LOO prediction at a test input $\mathbf{x}^*$ is given by

$$\widehat{\mu}_{\sim m}\left(\mathbf{x}^*\right) = \mathbf{d}_*^\top \mathbf{c}_{\sim m}, \tag{33}$$

where $\mathbf{d}_*^\top = (\Psi_1(\mathbf{x}^*), \ldots, \Psi_K(\mathbf{x}^*))$. Substituting the LOO PCE coefficients from (26) into (33), we get

$$\begin{aligned}
\widehat{\mu}_{\sim m}(\mathbf{x}^*) &= \mathbf{d}_*^\top \left(\mathbf{c} - \mathbf{A}\mathbf{d}_m \frac{y^{(m)} - \widehat{y}^{(m)}}{1 - h_{mm}} \right) \\
&= \mathbf{d}_*^\top \mathbf{c} - \mathbf{d}_*^\top \mathbf{A}\mathbf{d}_m \frac{y^{(m)} - \widehat{y}^{(m)}}{1 - h_{mm}} \\
&= \widehat{\mu}(\mathbf{x}^*) - \mathbf{d}_*^\top \mathbf{A}\mathbf{d}_m \frac{y^{(m)} - \widehat{y}^{(m)}}{1 - h_{mm}} \\
&= \widehat{\mu}(\mathbf{x}^*) - \mathbf{d}_*^\top \mathbf{A}\mathbf{d}_m \frac{y^{(m)} - \widehat{\mu}\left(\mathbf{x}^{(m)}\right)}{1 - h_{mm}} \\
&= \widehat{\mu}(\mathbf{x}^*) - \mathbf{d}_*^\top \left(\mathbf{D}^\top\mathbf{D}\right)^{-1} \mathbf{d}_m \frac{y^{(m)} - \widehat{\mu}\left(\mathbf{x}^{(m)}\right)}{1 - h_{mm}} \\
&= \widehat{\mu}(\mathbf{x}^*) - \mathbf{d}_*^\top \left(\mathbf{D}^\top\mathbf{D}\right)^{-1} \mathbf{d}_m r_m^{\mathrm{LOO}},
\end{aligned} \tag{34}$$

where we used that $\mathbf{A} = \left(\mathbf{D}^\top\mathbf{D}\right)^{-1}$, $\widehat{\mu}(\mathbf{x}^*) = \mathbf{d}_*^\top \mathbf{c}$, and the analytical expression (32) for the LOO residual.

References

[1] Pol D Spanos and Roger G Ghanem. ‘Stochastic finite element expansion for random media’. In: *Journal of engineering mechanics* 115.5 (1989), pp. 1035–1053.

[2] Dongbin Xiu and George Em Karniadakis. ‘The Wiener–Askey polynomial chaos for stochastic differential equations’. In: *SIAM Journal on Scientific Computing* 24.2 (2002), pp. 619–644.

[3] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach.* Courier Corporation, 2003.

[4] Emiliano Torre et al. ‘Data-driven polynomial chaos expansion for machine learning regression’. In: *Journal of Computational Physics* 388 (2019), pp. 601–623.

[5] Rad Haghi and Curran Crawford. ‘Surrogate models for the blade element momentum aerodynamic model using non-intrusive polynomial chaos expansions’. In: *Wind Energy Science* 7.3 (2022), pp. 1289–1304.

[6] Dimitrios Loukrezis and Herbert De Gersem. ‘Power module heat sink design optimization with ensembles of data-driven polynomial chaos surrogate models’. In: *e-Prime-Advances in Electrical Engineering, Electronics and Energy* 2 (2022), p. 100059.

[7] Joventino Oliveira Campos et al. ‘Polynomial chaos expansion surrogate modeling of passive cardiac mechanics using the Holzapfel–Ogden constitutive model’. In: *Journal of Computational Science* 71 (2023), p. 102039.

[8] Paolo Manfredi and Riccardo Trinchero. ‘Nonparametric formulation of polynomial chaos expansion based on least-square support-vector machines’. In: *Engineering Applications of Artificial Intelligence* 133 (2024), p. 108182.

[9] Wei Dai et al. ‘A cost surrogate model for TSO-DSO coordination based on polynomial chaos expansion’. In: *IEEE Transactions on Power Systems* (2025).

[10] Jiyoung Lee et al. ‘Surrogate models for multiregime flow problems’. In: *Physical Review Fluids* 10.2 (2025), p. 024703.

[11] Christian Soize and Roger G Ghanem. ‘Polynomial chaos representation of databases on manifolds’. In: *Journal of Computational Physics* 335 (2017), pp. 201–221.

[12] Katiana Kontolati et al. ‘Manifold learning-based polynomial chaos expansions for high-dimensional surrogate models’. In: *International Journal for Uncertainty Quantification* 12.4 (2022).

[13] Katiana Kontolati et al. ‘A survey of unsupervised learning methods for high-dimensional uncertainty quantification in black-box-type problems’. In: *Journal of Computational Physics* 464 (2022), p. 111313.

[14] Dimitris G Giovanis et al. ‘Polynomial chaos expansions on principal geodesic Grassmannian submanifolds for surrogate modeling and uncertainty quantification’. In: *Journal of Computational Physics* 519 (2024), p. 113443.

[15] Himanshu Sharma, Lukáš Novák and Michael D Shields. ‘Polynomial chaos expansion for operator learning’. In: *arXiv preprint arXiv:2508.20886* (2025).

[16] Lukáš Novák, Himanshu Sharma and Michael D Shields. ‘Physics-informed polynomial chaos expansions’. In: *Journal of Computational Physics* 506 (2024), p. 112926.

[17] Himanshu Sharma, Lukáš Novák and Michael Shields. ‘Physics-constrained polynomial chaos expansion for scientific machine learning and uncertainty quantification’. In: *Computer Methods in Applied Mechanics and Engineering* 431 (2024), p. 117314.

[18] Christopher K. I. Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning.* Cambridge, MA, USA: The MIT Press, 2006. ISBN: 026218253X.

[19] Laurent Valentin Jospin et al. ‘Hands-on Bayesian neural networks—A tutorial for deep learning users’. In: *IEEE Computational Intelligence Magazine* 17.2 (2022), pp. 29–48.

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

[21] Xiaobing Shang et al. ‘Active learning of ensemble polynomial chaos expansion method for global sensitivity analysis’. In: *Reliability Engineering & System Safety* 249 (2024), p. 110226.

[22] Roland Schobi, Bruno Sudret and Joe Wiart. ‘Polynomial-chaos-based Kriging’. In: *International Journal for Uncertainty Quantification* 5.2 (2015).

[23] Pierric Kersaudy et al. ‘A new surrogate modeling technique combining Kriging and polynomial chaos expansions–Application to uncertainty analysis in computational dosimetry’. In: *Journal of Computational Physics* 286 (2015), pp. 103–117.

[24] Xujia Zhu and Bruno Sudret. ‘Stochastic polynomial chaos expansions to emulate stochastic simulators’. In: *International Journal for Uncertainty Quantification* 13.2 (2023).

[25] Matteo Fontana, Gianluca Zeni and Simone Vantini. ‘Conformal prediction: a unified review of theory and new challenges’. In: *Bernoulli* 29.1 (2023), pp. 1–23.

[26] Evgeny Burnaev and Ivan Nazarov. ‘Conformalized kernel ridge regression’. In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA).* IEEE. 2016, pp. 45–52.

[27] Yaniv Romano, Evan Patterson and Emmanuel Candes. ‘Conformalized quantile regression’. In: *Advances in Neural Information Processing Systems* 32 (2019).

[28] Kexin Huang et al. ‘Uncertainty quantification over graph with conformalized graph neural networks’. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 26699–26721.

[29] Lena Podina, Mahdi Torabi Rad and Mohammad Kohandel. ‘Conformalized physics-informed neural networks’. In: *arXiv preprint arXiv:2405.08111* (2024).

[30] Edgar Jaber et al. ‘Conformal approach to Gaussian process surrogate evaluation with marginal coverage guarantees’. In: *Journal of Machine Learning for Modeling and Computing* 6.3 (2025).

[31] Margaux Zaffran et al. ‘Adaptive conformal predictions for time series’. In: *International Conference on Machine Learning.* PMLR. 2022, pp. 25834–25866.

[32] Nicolas Deutschmann, Mattia Rigotti and Maria Rodriguez Martinez. ‘Adaptive conformal regression with split-jackknife+ scores’. In: *Transactions on Machine Learning Research* (2024).

[33] Harris Papadopoulos et al. ‘Inductive confidence machines for regression’. In: *European Conference on Machine Learning*. Springer. 2002, pp. 345–356.

[34] Jing Lei et al. ‘Distribution-free predictive inference for regression’. In: *Journal of the American Statistical Association* 113.523 (2018), pp. 1094–1111.

[35] Vladimir Vovk. ‘Cross-conformal predictors’. In: *Annals of Mathematics and Artificial Intelligence* 74.1 (2015), pp. 9–28.

[36] Vladimir Vovk et al. ‘Cross-conformal predictive distributions’. In: *Conformal and Probabilistic Prediction and Applications*. PMLR. 2018, pp. 37–51.

[37] Rina Foygel Barber et al. ‘Predictive inference with the jackknife+’. In: *The Annals of Statistics* 49.1 (2021), pp. 486–507.

[38] Vignesh Gopakumar et al. ‘Uncertainty quantification of surrogate models using conformal prediction’. In: *arXiv preprint arXiv:2408.09881* (2024).

[39] Bert Masschaele et al. ‘Design of a strong gradient magnet for the deflection of nanoclusters’. In: *IEEE Transactions on Applied Superconductivity* 22.3 (2011), pp. 3700604–3700604.

[40] Malempati Madhusudana Rao and Randall J Swift. *Probability Theory with Applications*. Springer, 2006.

[41] Nora Lüthen, Stefano Marelli and Bruno Sudret. ‘Sparse polynomial chaos expansions: Literature survey and benchmark’. In: *SIAM/ASA Journal on Uncertainty Quantification* 9.2 (2021), pp. 593–649.

[42] Nora Lüthen, Stefano Marelli and Bruno Sudret. ‘Automatic selection of basis-adaptive sparse polynomial chaos expansions for engineering applications’. In: *International Journal for Uncertainty Quantification* 12.3 (2022).

[43] Giovanni Migliorati et al. ‘Approximation of quantities of interest in stochastic PDEs by the random discrete L^2 projection on polynomial spaces’. In: *SIAM Journal on Scientific Computing* 35.3 (2013), A1440–A1460.

[44] Giovanni Migliorati et al. ‘Analysis of discrete L^2 projection on polynomial spaces with random evaluations’. In: *Foundations of Computational Mathematics* 14.3 (2014), pp. 419–456.

[45] Mohammad Hadigol and Alireza Doostan. ‘Least squares polynomial chaos expansion: A review of sampling strategies’. In: *Computer Methods in Applied Mechanics and Engineering* 332 (2018), pp. 382–407.

[46] Paul G Constantine, Michael S Eldred and Eric T Phipps. ‘Sparse pseudospectral approximation method’. In: *Computer Methods in Applied Mechanics and Engineering* 229 (2012), pp. 1–12.

[47] Patrick R Conrad and Youssef M Marzouk. ‘Adaptive Smolyak pseudospectral approximations’. In: *SIAM Journal on Scientific Computing* 35.6 (2013), A2643–A2670.

[48] Gregery T Buzzard. ‘Global sensitivity analysis using sparse grid interpolation and polynomial chaos’. In: *Reliability Engineering & System Safety* 107 (2012), pp. 82–89.

[49] Giovanni Porta et al. ‘Inverse modeling of geochemical and mechanical compaction in sedimentary basins through Polynomial Chaos Expansion’. In: *Water Resources Research* 50.12 (2014), pp. 9414–9431.

[50] Armin Galetzka et al. ‘An hp-adaptive multi-element stochastic collocation method for surrogate modeling with information re-use’. In: *International Journal for Numerical Methods in Engineering* 124.12 (2023), pp. 2902–2930.

[51] Michael R Chernick. ‘The jackknife: a resampling method with connections to the bootstrap’. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4.2 (2012), pp. 224–226.

[52] Sonja Surjanovic and Derek Bingham. *Virtual library of simulation experiments: test functions and datasets*. Retrieved October 22, 2025, from http://www.sfu.ca/ ssurjano. 2013.

[53] Stefano Lorenzi et al. ‘POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations’. In: *Computer Methods in Applied Mechanics and Engineering* 311 (2016), pp. 151–179.

[54] Antonio Barletta. ‘The Boussinesq approximation for buoyant flows’. In: *Mechanics Research Communications* 124 (2022), p. 103939.

[55] László Könözsy. ‘The k-ω shear-stress transport (SST) turbulence model’. In: *A new hypothesis on the anisotropic Reynolds stress tensor for turbulent flows: Volume I: theoretical background and development of an anisotropic hybrid k-omega shear-stress transport/stochastic turbulence model.* Springer, 2019, pp. 57–66.

[56] Rui Wu et al. ‘A temperature-dependent thermal model of IGBT modules suitable for circuit-level simulations’. In: *IEEE Transactions on Industry Applications* 52.4 (2016), pp. 3306–3314.

[57] Andreas Pels et al. ‘Optimization of a Stern–Gerlach magnet by magnetic field–circuit coupling and isogeometric analysis’. In: *IEEE Transactions on Magnetics* 51.12 (2015), pp. 1–7.

[58] Dimitrios Loukrezis, Ulrich Römer and Herbert De Gersem. ‘Assessing the performance of Leja and Clenshaw-Curtis collocation for computational electromagnetics with random input data’. In: *International Journal for Uncertainty Quantification* 9.1 (2019).

[59] Lara Hawchar, Charbel-Pierre El Soueidy and Franck Schoefs. ‘Principal component analysis and polynomial chaos expansion for time-variant reliability problems’. In: *Reliability Engineering & System Safety* 167 (2017), pp. 406–416.

[60] Biswarup Bhattacharyya, Eric Jacquelin and Denis Brizard. ‘Uncertainty quantification of stochastic impact dynamic oscillator using a proper orthogonal decomposition-polynomial chaos expansion technique’. In: *Journal of Vibration and Acoustics* 142.6 (2020), p. 061013.

[61] Dimitrios Loukrezis, Eric Diehl and Herbert De Gersem. ‘Multivariate sensitivity-adaptive polynomial chaos expansion for high-dimensional surrogate modeling and uncertainty quantification’. In: *Applied Mathematical Modelling* 137 (2025), p. 115746.

[62] Ruiyao Zhang, Ping Zhou and Tianyou Chai. ‘Improved Copula-based conformal prediction for uncertainty quantification of multi-output regression’. In: *Journal of Process Control* 129 (2023), p. 103036.

[63] Chen Xu and Yao Xie. ‘Conformal prediction for time series’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.10 (2023), pp. 11575–11587.