

ORIGINAL ARTICLE OPEN ACCESS

# Bayesian Adaptive Polynomial Chaos Expansions

Kellin N. Rumsey<sup>1</sup> | Devin Francom<sup>1</sup> | Graham Gibson<sup>1</sup> | J. Derek Tucker<sup>2</sup> | Gabriel Huerta<sup>2</sup><sup>1</sup>Statistical Sciences, Los Alamos National Laboratory, Santa Fe, New Mexico, USA | <sup>2</sup>Statistical Sciences, Sandia National Laboratory, Albuquerque, New Mexico, USA**Correspondence:** Kellin Rumsey ([knrumsey@lanl.gov](mailto:knrumsey@lanl.gov))**Received:** 28 October 2025 | **Revised:** 9 December 2025 | **Accepted:** 11 February 2026**Keywords:** emulator | ordinal regression | sensitivity analysis | surrogate model | Wiener chaos

## ABSTRACT

Polynomial chaos expansions (PCEs) are widely used for uncertainty quantification (UQ) tasks, particularly in the applied mathematics community. However, PCE has received comparatively less attention in the statistics literature, and fully Bayesian formulations remain rare—especially with implementations in R. Motivated by the success of adaptive Bayesian machine learning models such as BART, BASS and BPPR, we develop a new fully Bayesian adaptive PCE method with an efficient and accessible R implementation: *khaos*. Our approach includes a novel proposal distribution that enables data-driven interaction selection and supports a modified *g*-prior tailored to PCE structure. Through simulation studies and real-world UQ applications, we demonstrate that the Bayesian adaptive PCE provides competitive performance for surrogate modeling, global sensitivity analysis and ordinal regression tasks.

## 1 | Introduction

Polynomial chaos expansions (PCEs), originally described by Wiener (1938), have become a widely used tool for surrogate modeling and uncertainty quantification (UQ), particularly in fields such as physics, engineering and applied mathematics (Ghanem et al. 1991; Xiu and Karniadakis 2002; Novak and Novak 2018). PCEs represent the response surface of a computer model as a linear combination of tensor products of orthogonal polynomials in the model's input variables. By projecting model outputs onto these polynomial bases, PCE provides a functional approximation of the input–output relationship. The technique has a long and established history, particularly for propagating uncertainty in simulations involving physical systems (Lüthen et al. 2021). PCE is also widely used for global sensitivity analysis, where Sobol or derivative-based indices can be derived analytically from the polynomial coefficients (Sudret 2008; Shao et al. 2017).

Despite its strengths, the broader use of PCE in statistical modeling has been somewhat limited by concerns related to overfitting in high-degree expansions, challenges with UQ and

sensitivity to input distributions (O'Hagan et al. 2013). At the same time, recent years have seen the success of fully Bayesian, nonparametric regression tools such as Bayesian additive regression trees (BART; Chipman et al. 2010), Bayesian adaptive spline surfaces (BASS; Denison et al. 1998; Francom and Sansó 2020) and Bayesian projection pursuit regression (BPPR; Collins et al. 2024). These models provide flexible, adaptive representations of complex surfaces, while offering natural UQ and strong empirical performance across a variety of tasks.

Inspired by these developments, we propose a new fully Bayesian implementation of adaptive PCE. The method incrementally selects polynomial basis functions using a Reversible Jump Markov Chain Monte Carlo (RJMC) algorithm. This allows the model to adapt its complexity to the data, enabling a dynamic balance between parsimony and flexibility. A novel proposal distribution governs the selection of interaction terms, leading to efficient exploration of the model space. We also consider a modified *g*-prior for the regression coefficients, which induces shrinkage based on the complexity of a basis function and leverages a Laplace approximation for fast and tuning-free inference.

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *Stat* published by John Wiley & Sons Ltd.

The rest of this article is organized as follows. Section 2 reviews relevant background on PCE and the sparse Bayesian PCE approach of Shao et al. (2017). Section 3 develops our proposed model, KHAOS (implementation in R at <https://github.com/knrumsey/khaos>). A simulation study comparing the method to several popular alternatives is presented in Section 4, and sensitivity analyses conducted with KHAOS are presented in Section 5 for two real-world datasets. Concluding remarks are given in Section 6.

## 2 | PCEs

### 2.1 | PCE Framework

In PCE, a function  $f(\mathbf{x})$  with input variables  $\mathbf{x} \in [0, 1]^p$  is approximately represented as

$$f(\mathbf{x}) \approx \sum_{m=0}^M \beta_m \prod_{j=1}^p \psi_{\alpha_{mj}}(x_j),$$

where  $\psi_\alpha(\cdot)$  is the *standardized shifted Legendre polynomial* of degree  $\alpha$ . These orthogonal polynomials are equal to  $\psi_\alpha(x) = \sqrt{2\alpha + 1} P_\alpha(2x - 1)$  where  $P_\alpha(\cdot)$  are the Legendre polynomials which satisfy the recurrence relation  $(\alpha + 1)P_{\alpha+1} = (2\alpha + 1)xP_\alpha(x) - \alpha P_{\alpha-1}(x)$  with  $P_0(x) = 1$ ,  $P_1(x) = x$ . We note that more general definitions exist, but the above is sufficient for our purposes.

For a basis function with multi-index  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p) \in \mathbb{N}^p$ , the *degree* is  $d(\boldsymbol{\alpha}) = \sum_{j=1}^p \alpha_j$  and the *order* is  $q(\boldsymbol{\alpha}) = \sum_{j=1}^p 1(\alpha_j > 0)$ , where  $1(\cdot)$  is the indicator function. A PCE representation is said to be *full* with respect to degree  $d$  and order  $q$  if all coefficients are nonzero and it contains a term for every multi-index  $\boldsymbol{\alpha}$  in the set

$$\mathcal{A}_{p,d,q} = \{\boldsymbol{\alpha} \in \mathbb{N}^p : d(\boldsymbol{\alpha}) \leq d \text{ and } q(\boldsymbol{\alpha}) \leq q\}.$$

A PCE is said to be *sparse* if it contains terms for only a subset of  $\mathcal{A}_{p,d,q}$  (or equivalently, if any of the coefficients are exactly zero). We note that, for PCE models with maximum degree  $d$  and maximum order  $q$ , there are

$$|\mathcal{A}_{p,d,q}| = \sum_{i=1}^q \sum_{j=1}^d \binom{p}{i} \binom{j-1}{i-1} = \mathcal{O}\left(\frac{(pd)^q}{(q!)^2}\right) = \mathcal{O}\left(\left(\frac{pd}{q^2}\right)^q\right)$$

permissible basis functions.

For this to remain feasible for even moderately sized input dimensions ( $p$ ), one must either (i) place restrictions on  $d$  and/or  $q$ , or (ii) induce a high level of sparsity. A wide range of solvers have been proposed for sparse PCE, including convex optimization methods such as LASSO and LARS, greedy stepwise algorithms like orthogonal matching pursuit and Bayesian compressive sensing approaches based on variational inference or EM algorithms (see Lüthen et al. (2021) for an extensive review). Most of these approaches rely on point estimates and cross-validation to select model complexity and do not provide full posterior UQ.

Fully Bayesian approaches to sparse PCE are less common. One recent example is the method of Shao et al. (2017), which combines a likelihood-based model with sparsity-inducing priors and uses a forward-selection algorithm for model construction. While this approach does not sample from the full posterior distribution, it borrows strength from Bayesian modeling and offers a computationally efficient alternative to traditional MCMC. In the following section, we briefly review this approach, which we include in the simulation study of Section 4.

### 2.2 | Sparse Bayesian PCE

Several algorithms for constructing sparse Bayesian PCE models have been proposed in recent years. For example, Bürkner et al. (2023) employ an R2D2 regularization prior on the coefficients of a dense polynomial basis using MCMC, and Pan et al. (2020) obtain a sparse set of polynomials by adding observations sequentially using an optimality criterion. Here, we briefly describe the popular algorithm proposed by Shao et al. (2017) (SBPCE), and we discuss a few optional modifications which are available in the `khaos` implementation. This algorithm is not fully Bayesian in the sense that  $M$  and the design matrix  $\Psi$  (whose rows contain the basis evaluations  $\boldsymbol{\psi}_m \equiv \boldsymbol{\psi}_m(\mathbf{x}|\boldsymbol{\alpha}_m)$ ) are determined algorithmically rather than being inferred as part of the posterior. The SBPCE approach proceeds as follows:

1. For fixed maximum degree  $d_{\max}$  and maximum order  $q_{\max}$ , generate the complete set of  $|\mathcal{A}_{p,d_{\max},q_{\max}}|$  basis functions.
2. Initialize a model which returns the sample mean of the response vector  $\mathbf{y} = (y_1, y_2, \dots, y_n)$ , i.e.,  $(y_1 + \dots + y_n)/n$ , for all  $\mathbf{x}$ .
3. For each basis function, compute the sample correlation  $r_m = \text{cor}(\boldsymbol{\psi}_m(\mathbf{x}|\boldsymbol{\alpha}_m), \mathbf{y})$  and reorder the basis columns so that  $r_m^2 \geq r_{m+1}^2$ .
4. For each basis function, compute the squared partial correlation component  $\rho_{m|1,\dots,m-1}^2$ . Reorder the basis functions again so that  $\rho_{m|1,\dots,m-1}^2 \geq \rho_{m+1|1,\dots,m}^2$ .
5. For every  $m \in \{0, \dots, M\}$ , consider the model  $\mathcal{M}_m$  with basis functions  $\boldsymbol{\psi}_m, \dots, \boldsymbol{\psi}_0$ . Take  $m^*$  to be the largest  $M$  such that the Kashyap information criteria (KIC) for model  $\mathcal{M}_m$  is larger than that of  $\mathcal{M}_{m+1}$ .
6. **Enrichment:** If  $\mathcal{M}_{m^*}$  model contains a maximally complex term (i.e. one with degree  $d_{\max}$  and/or order  $q_{\max}$ ), then we (i) increment  $d_{\max}$  and/or  $q_{\max}$ , (ii) enrich the set of candidate basis functions and (iii) return to step 2. Otherwise, return  $\mathcal{M}_{m^*}$ .

The original enrichment scheme of SBPCE is quite restrictive, leading to a fast and parsimonious training algorithm. Unfortunately, it can permanently cut out certain input variables and leads to a strong dependence on the initial choice of  $d_{\max}$  and  $q_{\max}$ . In Appendix A of the [Supporting Information](#), we discuss several alternative enrichment strategies which can improve the accuracy of the SBPCE approach (and reduce dependence on tuning parameters) at the cost of increased

computation. In Section 3.4, we also show how Step 5 can be replaced with a closed-form Bayes factor based on the modified g-prior.

### 2.3 | Sobol Indices

One appealing feature of PCE is that it makes it easy to compute Sobol indices, which are widely used for global sensitivity analysis (Sobol 2001; Sudret 2008; Francom et al. 2018).

In a Sobol analysis, the function of interest is assumed to admit an ANOVA-like decomposition:

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^p f_i(x_i) + \sum_{i<j}^p f_{ij}(x_i, x_j) + \dots + f_{1,\dots,p}(x_1, \dots, x_p) = \sum_{m=0}^M f_{\mathbf{u}_m}(\mathbf{x}_{\mathbf{u}_m}),$$

with every term being orthogonal and centered at zero (except for  $f_0$ ). It follows that the variance of  $f(\mathbf{x})$  can then be decomposed as

$$\text{Var}(f(\mathbf{x})) = \sum_{i=1}^p V_i + \sum_{i<j}^p V_{ij} + \dots + V_{1,\dots,p} = \sum_{m=1}^M V_{\mathbf{u}_m}.$$

The  $V_{\mathbf{u}}$  terms are usually rescaled (so that they sum to unity) as  $S_{\mathbf{u}} = V_{\mathbf{u}}/\text{Var}(f(\mathbf{x}))$  and called *partial sensitivity indices*. The *total sensitivity index* for the  $i^{\text{th}}$  input is defined as  $T_i = \sum_{\mathbf{u}:i \in \mathbf{u}} S_{\mathbf{u}}$  which are only guaranteed to sum to at least 1.

The main insight is that, by construction, PCE models are already expressed in this orthogonal form—assuming the inputs are independent and uniformly distributed on  $[0, 1]$ . In particular, each term in the PCE expansion can be associated with a specific subset  $\mathbf{u}$  of input variables, and the contribution to the variance is  $V_{\mathbf{u}} = \sum_{m \in \mathcal{A}_{\mathbf{u}}} \beta_m^2$ , where  $\mathcal{A}_{\mathbf{u}}$  indexes all basis functions that depend on exactly the variables in  $\mathbf{u}$ . In words, the partial sensitivity index for a subset  $\mathbf{u}$  is the sum of squared coefficients for all PCE terms that involve exactly those variables. For further discussion, see Sudret (2008).

## 3 | Adaptive Bayesian PCE

Following the principle of NUAP (no unnecessary acronyms please; Nature Methods Editorial (2011)), we avoid labeling our approach with a cumbersome acronym. Instead, we refer to this method as KHAOS, in reference to the `khaos` R package that implements it, which was named in turn for the primordial void of Greek mythology (<https://github.com/knrumsey/khaos>). Despite the name, the KHAOS algorithm (or model, or approach) refers simply to the adaptive Bayesian PCE described in this section.

### 3.1 | The KHAOS Model

Let  $y_i$  denote the response variable and  $\mathbf{x}_i$  denote a vector of  $p$  covariates ( $i = 1, \dots, n$ ). Without loss of generality, we assume that  $\mathbf{x} \in [0, 1]^p$ . The response is modeled as

$$\begin{aligned} y_i &= f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2) \\ f(\mathbf{x}) &= \beta_0 + \sum_{m=1}^M \beta_m \Psi_m(\mathbf{x} | \boldsymbol{\alpha}_m) \\ \Psi_m(\mathbf{x} | \boldsymbol{\alpha}_m) &= \prod_{i=1}^p \psi_{\alpha_{mj}}(x_j), \end{aligned} \quad (1)$$

where each basis function  $\Psi_m$  is fully defined by the multi-index  $\boldsymbol{\alpha}_m$  (described in Section 2). We define  $\mathbf{A} = \{\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_M\}$  and specify the prior for the basis function parameters  $(\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_M, M)$  as

$$\begin{aligned} \boldsymbol{\alpha}_m | M &\stackrel{\text{iid}}{\sim} \text{Unif}(\mathcal{A}_{p, d_{\max}, q_{\max}}), \quad m = 1, \dots, M \\ M | \lambda &\sim \text{Poiss}(\lambda), \\ \lambda &\sim \text{Gamma}(\text{shape} = a_M, \text{rate} = b_M). \end{aligned} \quad (2)$$

Although a prior that penalizes complexity in the multi-indices (e.g., by degree or order) could be specified, we adopt a uniform prior over admissible basis functions and instead encourage parsimony through the modified g-prior on the coefficients, as described in Section 3.4.

For the remaining parameters  $(\boldsymbol{\beta}, \sigma^2)$ , we specify the prior

$$\begin{aligned} \boldsymbol{\beta} | M, \sigma^2, \mathbf{S}_0 &\sim \mathcal{N}_{M+1}(\mathbf{0}, \sigma^2 \mathbf{S}_0) \\ \sigma^2 &\sim \text{Inv-Gamma}(\text{shape} = a_\sigma, \text{rate} = b_\sigma), \end{aligned} \quad (3)$$

where  $\mathbf{S}_0$  is a prior covariance matrix whose structure we discuss in Section 3.4.

### 3.2 | Efficient Posterior Sampling

Fully Bayesian inference is complicated here by the fact that  $M$ , the number of basis functions, is allowed to grow and shrink. This requires transdimensional proposals, which we handle using a reversible jump Markov chain Monte Carlo (RJMCMC) algorithm. This framework has seen success in several modern contexts including (Francom and Sansó 2020; Rumsey, Francom, and Shen 2024; Collins et al. 2024).

At each iteration of the MCMC sampler, we propose to modify the current model using one of four possible moves:

1. *Birth*: Propose adding a new basis function.
2. *Death*: Propose removing an existing basis function.
3. *Mutation (degree)*: Modify the degree partition of an existing basis function.
4. *Mutation (variable)*: Swap a variable within an existing basis function.

These moves allow the model to flexibly explore the space of basis configurations. The remaining parameters  $(\boldsymbol{\beta}, \sigma^2)$  are updated via Gibbs steps, using their conditional posteriors described in Section 3.4.

Each proposed move is accepted with probability

$$\log \alpha_X = \log \left( \frac{p(\mathbf{y} | \mathcal{M}_{\text{cand}})}{p(\mathbf{y} | \mathcal{M}_{\text{curr}})} \right) + \log \left( \frac{p(\mathcal{M}_{\text{cand}})}{p(\mathcal{M}_{\text{curr}})} \right) + \log A_X, \quad (4)$$

where  $\mathcal{M}_{\text{curr}}$  and  $\mathcal{M}_{\text{cand}}$  refer to the current and proposed model, respectively. The final term  $\log A_X$  accounts for the proposal probabilities specific to move type  $X \in \{\text{Birth}, \text{Death}, \text{Mutate1}, \text{Mutate2}\}$ . The first two terms correspond to the log-likelihood ratio and the log-prior ratio, respectively. Explicit equations for  $p(\mathbf{y} | \mathcal{M})$  and  $p(\mathcal{M})$  are given in Appendix B of the [Supporting Information](#).

For each of the move types (discussed below), the prior ratio simplifies considerably since the difference in  $M$  is at most one:

$$\frac{p(\mathcal{M}_{\text{cand}})}{p(\mathcal{M}_{\text{curr}})} = \begin{cases} (M + a_M) [(M + 1)(b_M + 1) | \mathcal{A}_{p, d_{\text{max}}, q_{\text{max}}} | ]^{-1}, & \text{Birth} \\ (M - 1 + a_M)^{-1} M (b_M + 1) | \mathcal{A}_{p, d_{\text{max}}, q_{\text{max}}} |, & \text{Death} \\ 1, & \text{Mutate1, Mutate2} \end{cases}$$

### 3.2.1 | Birth Step

During a birth step, selected with probability  $P_B$ , we need only propose a new vector of degrees  $\alpha^*$ , in order to completely define the new basis function. Nott et al. (2005) suggest an efficient proposal that favors choosing variables which are already in the model—important when  $p$  is large and exploring all  $2^p$  interactions are not possible. However, their approach requires evaluating Wallenius' noncentral hypergeometric distribution, which rapidly becomes computationally burdensome or numerically unstable in many practical settings. As a result, Nott et al. (2005) restrict their algorithm to pairwise interactions, while Francom et al. (2018) extend it to three-way interactions. We introduce a related approach that achieves similar variable-selection goals without these limitations. Specifically, we use a weighted coin-flipping procedure that avoids the need for Wallenius' distribution and does not impose a hard cap on the maximum interaction order.

We begin by sampling an expected interaction order  $q_0$  from the set  $\{1, \dots, q_{\text{max}}\}$  with weights proportional to  $q_0^{-s_{q_0}}$  (default  $s_q = 1$ ). Next, we construct the probability  $\eta_j$  that  $x_j$  will be active in the proposed basis function, such that  $\sum_{j=1}^p \eta_j = q_0 = E(q)$ . The idea is that  $\eta_j \geq \eta_j$  if  $x_j$  is more active than  $x_j$  in the current model (see Appendix C of the [Supporting Information](#) for details).

We then independently flip a coin for the inclusion of each input,  $\chi_j \sim \text{Bern}(\eta_j)$ , which gives us the proposed interaction order as  $q(\alpha^*) = \sum_{j=1}^M \chi_j$ . The total degree is sampled from the set  $\{q(\alpha^*), \dots, p\}$  with sampling weights  $d^{-s_d}$  (default  $s_d = 1$ ) and is randomly partitioned across the  $q$  active variables (i.e., those with  $\chi_j = 1$ ). This is done so that each suitable partitioning is equally likely, with probability  $\left( \frac{d(\alpha^*) - 1}{d(\alpha^*) - 1} \right)^{-1}$ .

For the Metropolis–Hastings acceptance ratio, the proposal term can be written as

$$A_{\text{Birth}} = \frac{P_D \binom{d-1}{q-1} \prod_{j=1}^p \eta_j^{\chi_j} (1 - \eta_j)^{1 - \chi_j}}{P_B (M + 1) q_0^{s_{q_0}} c_d d^{s_d} c_d}, \quad (5)$$

where  $c_q = \sum_{q_0=1}^{q_{\text{max}}} q_0^{-s_{q_0}}$  and  $c_d = \sum_{d=q}^p d^{-s_d}$ . Delayed rejection steps are also included to improve efficiency (Green and Mira 2001); see Appendix C in the [Supporting Information](#) for more details.

### 3.2.2 | Death Step

During a death step, selected with probability  $P_D$ , a basis function is randomly selected for deletion. Because this move reduces the model dimension, the reverse proposal corresponds to a birth step—where a specific multi-index  $\alpha^*$  would have been proposed using the weighted coin-flipping strategy described previously. The reverse move's proposal probability must marginalize over all values of the expected interaction order  $q_0$  that could have generated the deleted basis function.

The full proposal ratio for the Metropolis–Hastings acceptance probability is then as follows:

$$A_{\text{Death}} = \frac{P_B d^{s_d} c_d}{P_D M \binom{d-1}{q-1}} \cdot \left[ \frac{1}{c_q} \sum_{q_0=1}^{q_{\text{max}}} q_0^{-s_{q_0}} \prod_{j=1}^p \eta_j(q_0)^{\chi_j} (1 - \eta_j(q_0))^{1 - \chi_j} \right], \quad (6)$$

where  $q$  and  $d$  refer to the interaction order and total degree of the deleted basis function and  $\chi_j$  indicates whether variable  $x_j$  was included in that term.

To account for delayed rejection in the Birth step, we must condition on the fact that certain proposals would have been rejected (e.g., those yielding  $q = 0$  or  $q > q_{\text{max}}$ ). This requires evaluation of the Poisson binomial densities (or an efficient normal approximation). See the [Supporting Information](#) for additional details.

### 3.2.3 | Mutate Steps

When a mutation step is selected (with probability  $P_M = 1 - P_B - P_D$ ), a single basis function is modified without changing the model dimension. Two types of mutation are used: (i) resampling the degree partition across the active variables and (ii) swapping one active variable for a previously inactive one. The probability of selecting each type is adapted throughout the MCMC, based on their empirical acceptance rates but is never allowed to drop below 10% for either type (unless  $p \leq 3$  in which case variable mutation is unnecessary).

In a degree mutation, we change only the total degree  $d$  and randomly repartition it across the  $q$  active variables. The acceptance ratio includes the change in proposal density due to the total degree and its partitioning:

$$A_{\text{Mutate1}} = \frac{d_{\text{curr}}^{s_d} \cdot \binom{d_{\text{curr}} - 1}{q - 1}}{d_{\text{cand}}^{s_d} \cdot \binom{d_{\text{cand}} - 1}{q - 1}}, \quad (7)$$

where  $q$  is the (fixed) interaction order,  $d_{\text{cand}}$  is the proposed degree, and  $d_{\text{curr}}$  is the current degree. The two binomial terms reflect the uniform partitioning over the  $q$  active variables.

In a variable-swap mutation, one active variable in a basis function is randomly replaced by an inactive one. The proposal distribution is an adaptive categorical distribution, proportional to the current variable inclusion frequencies (plus a fixed baseline). To ensure detailed balance, we compute the Metropolis–Hastings proposal ratio using the forward and reverse selection probabilities:

$$A_{\text{Mutate2}} = \frac{\tilde{\pi}_{\text{rev}}(x_{\text{old}})}{\tilde{\pi}_{\text{fwd}}(x_{\text{new}})}, \quad (8)$$

where  $\tilde{\pi}_{\text{fwd}}$  and  $\tilde{\pi}_{\text{rev}}$  are the normalized empirical inclusion probabilities used to propose the new and old variables, respectively.

### 3.3 | Gibbs Steps

Given the current set of basis functions, the remaining model parameters  $(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\lambda})$  can be updated using standard conjugate Gibbs steps. The update for  $\boldsymbol{\lambda}$  is

$$\lambda_j \cdot \sim \text{Gamma}(\text{shape} = a_M + M, \text{rate} = b_M + 1). \quad (9)$$

The full conditional posteriors for  $\boldsymbol{\beta}$  and  $\sigma^2$  are conjugate under all of the priors considered in this work (discussed in the next section). Given the current design matrix  $\boldsymbol{\Psi}$ , define the following:

$$\boldsymbol{\Sigma}_n = (\boldsymbol{\Psi}^T \boldsymbol{\Psi} + \mathbf{S}_0^{-1})^{-1}, \quad \boldsymbol{\mu}_n = \boldsymbol{\Sigma}_n \boldsymbol{\Psi}^T \mathbf{y}.$$

Then the Gibbs updates are as follows:

$$\begin{aligned} \boldsymbol{\beta} | \cdot &\sim \mathcal{N}(\boldsymbol{\mu}_n, \sigma^2 \boldsymbol{\Sigma}_n) \\ \sigma^2 | \cdot &\sim \text{Inv-Gamma}\left(\text{shape} = a_\sigma + \frac{n}{2}, \text{rate} = b_\sigma + \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2\right), \end{aligned} \quad (10)$$

where  $\hat{\mathbf{y}} = \boldsymbol{\Psi} \boldsymbol{\beta}$ . The prior matrix  $\mathbf{S}_0$  depends on the choice of coefficient prior. Full specifications for  $\mathbf{S}_0$  under the ridge prior, g-prior and modified g-prior are provided in Section 3.4.

### 3.4 | Prior Structure on Coefficients

In this section, we describe the prior placed on the regression coefficients, focusing primarily on a modified g-prior that allows different levels of shrinkage for different basis terms. The traditional g-prior was introduced by Zellner (1986) as a computational convenient prior that helps to regularize the coefficients and perform model selection. The g-prior is akin to placing a constant prior on the mean of  $\mathbf{y}$ , rather than on  $\boldsymbol{\beta}$  (Robert et al. 2007). Our proposed modification is a “ $n$ -component” g-prior in the terminology of Zhang et al. (2016) and seeks to induce stronger regularization on the coefficients for higher complexity basis functions.

We begin by defining the vector  $\mathbf{g}$  with elements

$$g_m = \left( \frac{1}{1 + q(\boldsymbol{\alpha}_m) [d(\boldsymbol{\alpha}_m) + q(\boldsymbol{\alpha}_m) - 2]} \right)^{\zeta/2},$$

where  $\zeta \geq 0$  is a tuning parameter (with default  $\zeta = 1$ ) that controls how strong the penalty for complexity should be. By setting  $\zeta = 0$ , this method collapses to the traditional Zellner–Siow g-prior. Our modified prior is given by

$$\begin{aligned} \boldsymbol{\beta} | M, \sigma^2, \mathbf{g}_0^2 &\sim \mathcal{N}_{M+1}(\mathbf{0}, \sigma^2 \mathbf{g}_0^2 \mathbf{D}(\mathbf{g}) (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \mathbf{D}(\mathbf{g})) \\ \mathbf{g}_0^2 &\sim \text{Inv-Gamma}(a_g, b_g), \end{aligned} \quad (11)$$

where  $\mathbf{D}(\mathbf{g})$  is the diagonal matrix with the elements of  $\mathbf{g}$  on its diagonal. This is consistent with Equation (3) with  $\mathbf{S}_0 = \mathbf{g}_0^2 \mathbf{D}(\mathbf{g}) (\boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1} \mathbf{D}(\mathbf{g})$ . Although  $\boldsymbol{\Sigma}_n$  can be computed directly in terms of  $\mathbf{S}_0$ , we usually prefer to compute via the

$$\boldsymbol{\Sigma}_n = (\mathbf{G} \odot \boldsymbol{\Psi}^T \boldsymbol{\Psi})^{-1},$$

where  $\mathbf{G}$  is a matrix with elements

$$G_{m\ell} = \frac{g_0^2 g_m g_\ell + 1}{g_0^2 g_m g_\ell},$$

which makes obvious the connection to the traditional Zellner–Siow Cauchy g-prior, when  $\mathbf{g} = \mathbf{1}$  (Liang et al. 2008).

The posterior update for the global regularizer  $g_0^2$  is based on the conditional posterior

$$\pi(g_0^2 | \mathbf{y}) \propto g_0^{-2(a_g + M/2)} \exp(-b_g/g_0^2) |\boldsymbol{\Sigma}_n|^{1/2}. \quad (12)$$

There is no easy way to directly sample from Equation (12) (unless  $\mathbf{g} \propto \mathbf{1}$ ), but an efficient Laplace approximation can be computed based on the inverse gamma distribution (especially when  $\boldsymbol{\Psi}^T \boldsymbol{\Psi} \approx n\mathbf{I}$ , which occurs for PCE when the input design is orthogonal). We recommend sampling  $g_0^2$  using Metropolis–Hastings, with the Laplace approximation as the proposal distribution. Specifically, we find  $(\hat{a}_g, \hat{b}_g)$  so that  $g_0^2 | \mathbf{y} \stackrel{\text{aprx}}{\sim} \text{Inv-Gamma}(\hat{a}_g, \hat{b}_g)$ ; using this inverse gamma distribution for the proposal, the acceptance probability becomes

$$\min \left( 1, \frac{\pi(g_0^2_{\text{cand}} | \mathbf{y}) \text{IG}(g_0^2_{\text{curr}} | \hat{a}_g, \hat{b}_g)}{\pi(g_0^2_{\text{curr}} | \mathbf{y}) \text{IG}(g_0^2_{\text{cand}} | \hat{a}_g, \hat{b}_g)} \right), \quad (13)$$

where  $\text{IG}(\cdot | a, b)$  denotes the inverse-gamma density with shape  $a$  and rate  $b$ . To see how this prior can be used for the Sparse PCE approach of Shao et al. (2017) (replacing KIC with Bayes Factors based on the modified g-prior), see Appendix B of the [Supporting Information](#).

Note that `khaoS` also supports a ridge penalty, that is,  $\mathbf{S}_0 = \tau^{-2} \mathbf{I}$  with  $\tau^2$  fixed (default  $\tau^2 = 10^5$ ), which often works quite well for deterministic simulators but sometimes overfits (or needs tuning) for noisy data.

### 3.5 | Laplace Approximations

While directly sampling  $g_0^2$  from its conditional posterior is challenging, a Laplace approximation provides a fast and robust

solution in this setting. Our strategy will be to construct the approximation under the simplifying assumption that the design matrix satisfies  $\Psi^T \Psi = n\mathbf{I}$ , which holds exactly for orthogonal designs on  $\mathbf{x}$ . In many cases, this approximation may be sufficient (especially when using it as a proposal for Metropolis-Hastings). In cases where the orthogonality assumption may not be appropriate, we can instead construct a Laplace approximation to the exact conditional posterior via the Newton-Raphson iterations, using the orthogonal solution as an efficient starting place.

Under this simplifying assumption, the conditional posterior simplifies to

$$\pi(g_0^2 | \mathbf{y}, \text{orthogonal design}) \propto g_0^{-2(a_g + M/2)} \exp(-b_g/g_0^2) \prod_{m=1}^M \left( \frac{g_0^2 \sigma_m^2}{1 + g_0^2 \sigma_m^2} \right)^{1/2}.$$

The mode of the Laplace approximation can be obtained via fixed-point iteration on a monotonic function  $h(g_0^2)$ . We start by initializing  $\theta_1^* = b_g/a_g$ , and we alternate between computing  $G_k$  and  $\theta_{k+1}^*$  where

$$\theta_k^* = \frac{-a_g + \sqrt{a_g^2 + 4b_g G_k}}{2G_k} \quad G_k = \frac{1}{2} \sum_{m=1}^M \frac{g_m^2}{1 + \theta_{k-1}^* \sigma_m^2}.$$

We find that this sequence converges rapidly in practice to the mode  $m_\theta$ . The spread of the approximation is found the usual way:

$$s_\theta^2 = \left( -\frac{\partial^2}{\partial \theta^2} \log \pi(\theta | \mathbf{y}, \text{orthogonal design})|_{\theta=m_\theta} \right)^{-1}.$$

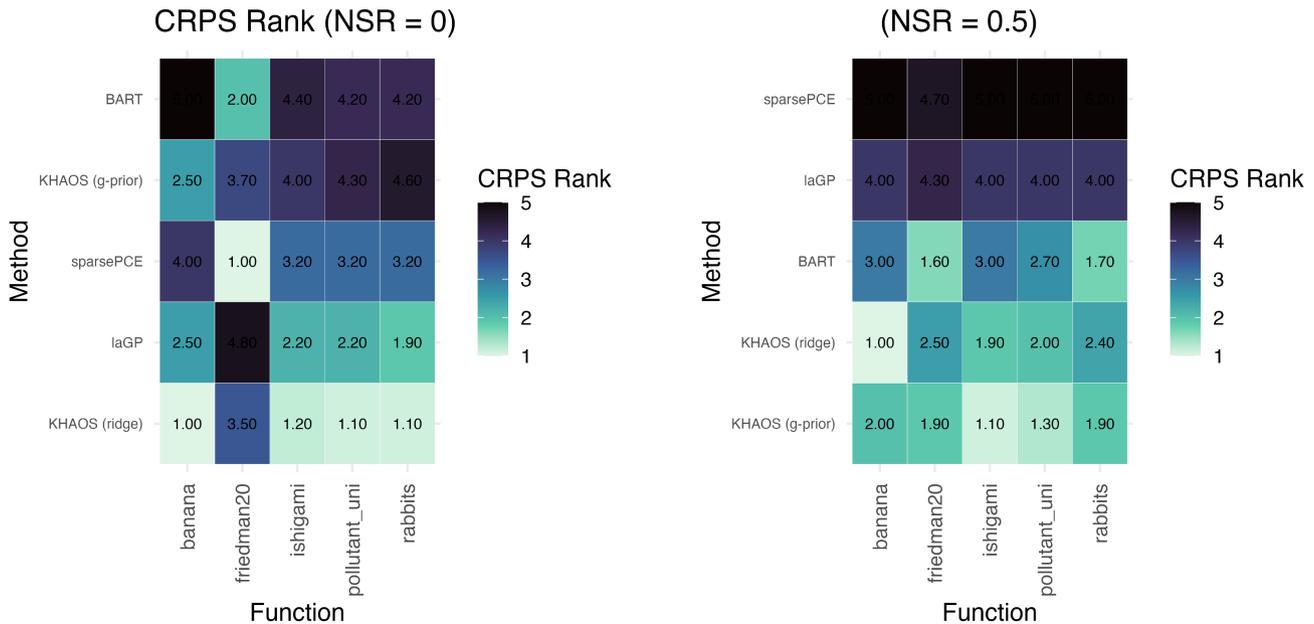
Finally, we solve for the corresponding inverse Gamma parameters as  $\hat{a}_g = 2 + m_\theta^2/s_\theta^2$  and  $\hat{b}_g = m_\theta \hat{a}_g$ . We find that, especially for computer experiments where Latin hypercube designs are common (McKay et al. 1979), this approximation is sufficient to get good acceptance from Metropolis-Hastings. If needed, however, the more general case can be found using Jacobi's formula and the Newton-Raphson iteration. See Appendix D of the [Supporting Information](#) for additional details and derivations.

## 4 | Simulation Study

We compare the performance of KHAOS under (i) a ridge prior and (ii) the modified  $g$ -prior from Section 3.4, against several fast competitors. Specifically, we compare to Bayesian additive regression trees (BART; Chipman et al. 2010), the local approximate Gaussian process (laGP; Gramacy and Apley 2015) and a sparse PCE method (Shao et al. 2017), implemented as `sparse_khaos` in the accompanying `khaos` package. This implementation uses a full rebuild enrichment strategy with early stopping to bound the computational complexity. All emulators are run at default settings, and R code for reproduction is included in the [Supporting Information](#).

Simulations are conducted using the `duqling` R package, designed for transparent and reproducible benchmarking (K. Rumsey 2023). We evaluate the five methods on five test functions:

- *banana*: A  $p = 2$  version of Rosenbrock's classic banana function.
- *ishigami*: A  $p = 3$  test function commonly used in the PCE literature (Ishigami and Homma 1990).



**FIGURE 1** | (a) Average CRPS rankings across 10 replications of each test function. In the noise free ( $NSR = 0$ ) setting, KHAOS with a ridge prior has the best average ranking. (b) Average CRPS rankings across 10 replications of each test function. In the high-noise ( $NSR = 0.5$ ) setting, KHAOS with the modified  $g$ -prior has the best average ranking.

- *rabbits*: A  $p = 3$  logistic growth model (Gotelli et al. 2004).
- *pollutant\_uni*: A  $p = 4$  scalar-output model of pollutant diffusion in a river (Bliznyuk et al. 2008).
- *friedman20*: A  $p = 20$  function with only the first five variables active (Friedman 1991).

See the above references or `duqling` documentation for further details.

For each test function, we generated a training set of  $n = 1000$  points using maximin Latin hypercube sampling (McKay et al. 1979). Responses include additive noise under two noise-to-signal ratio (NSR) settings: noise-free emulation ( $NSR = 0$ ) and high-noise regression ( $NSR = 0.5$ ).

We evaluate each emulation method using continuous ranked probability scores (CRPS), a proper scoring rule that balances

precision and accuracy of a distributional prediction (Gneiting and Raftery 2007). The CRPS is defined as

$$\begin{aligned} \text{CRPS}(F, y_{\text{true}}) &= \int_{-\infty}^{\infty} (F(z) - \mathbf{1}\{z \geq y_{\text{true}}\})^2 dz \\ &= \mathbb{E}_F |Y - y_{\text{true}}| - \frac{1}{2} \mathbb{E}_F |Y - Y'|, \end{aligned}$$

where  $Y$  and  $Y'$  are independent draws from the predictive distribution  $F$ . Each method is tested on an independent test set of size 1000. All simulation scenarios are replicated 10 times with fresh designs and noise.

## 4.1 | Results

A visual summary of the results for the noise free setting is given in Figure 1a, which shows the average CRPS ranking of

**TABLE 1** | Results for the simulation study in the noise free ( $NSR = 0$ ) setting.

Method	Function	Avg. CRPS	Avg. time	Within 1% rate
KHAOS (ridge)	banana	< 0.0001	7.263	1
KHAOS (g-prior)	banana	1.204	8.855	0
sparsePCE	banana	6.837	0.049	0
BART	banana	12.572	7.578	0
laGP	banana	1.143	20.546	0
KHAOS (ridge)	ishigami	0.012	13.623	0.9
KHAOS (g-prior)	ishigami	0.181	11.258	0
sparsePCE	ishigami	0.066	0.486	0
BART	ishigami	0.173	7.056	0
laGP	ishigami	0.030	20.223	0.1
KHAOS (ridge)	rabbits	0.001	44.251	0.9
KHAOS (g-prior)	rabbits	0.016	17.578	0
sparsePCE	rabbits	0.004	0.367	0
BART	rabbits	0.007	7.869	0
laGP	rabbits	0.001	20.463	0.1
KHAOS (ridge)	pollutant_uni	0.0003	12.339	0.9
KHAOS (g-prior)	pollutant_uni	0.024	8.460	0.1
sparsePCE	pollutant_uni	0.010	0.070	0
BART	pollutant_uni	0.011	6.592	0
laGP	pollutant_uni	0.008	22.280	0
KHAOS (ridge)	friedman20	0.938	9.276	0
KHAOS (g-prior)	friedman20	0.998	9.855	0
sparsePCE	friedman20	0.079	12.534	1
BART	friedman20	0.209	6.556	0
laGP	friedman20	1.354	40.450	0

Note: The “within 1 % rate” column gives the proportion of the time that the CRPS of an emulator was within 1 % of the best CRPS across all five emulators.

each emulator across the 10 replications. Complete results including timing and raw CRPS averages are given in Table 1. In the high-noise setting, equivalent figures and tables are given by Figure 1b and Table 2.

Some takeaways of this analysis include the following:

- No single emulator is ever the best across all five test functions.
- In the noise free setting, the KHAOS approach with a ridge prior has the best average CRPS rank.
- In the high-noise setting, the KHAOS approach with a modified  $g$ -prior has the best average CRPS rank. This is likely due to the  $g$ -priors ability to reduce potential for overfitting.
- The Sparse PCE approach does reasonably well in the noise free setting (and always has the best CRPS for ‘friedman20’) but appears to overfit in the high-noise setting.

- When  $NSR = 0$ , the laGP emulator performs well. When  $NSR = 0.5$ , BART demonstrates good performance. Both of these findings are consistent with previous work.

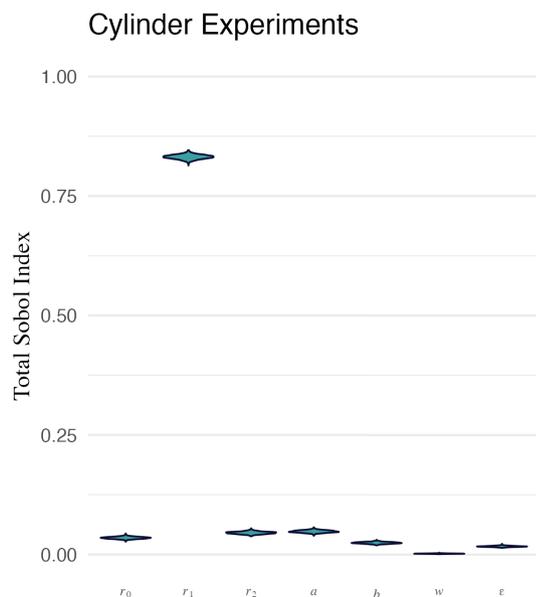
While emulator performance is problem-dependent, KHAOS performs consistently well across functions and demonstrates robustness to both low- and high-noise settings. For additional figures, including boxplots of CRPS, heatmaps based on RMSE and a Pareto plot comparing speed and accuracy, see Appendix F in the [Supporting Information](#). Appendix E also demonstrates convergence of the estimated Sobol indices for the Ishigami function.

## 5 | Real Data Examples

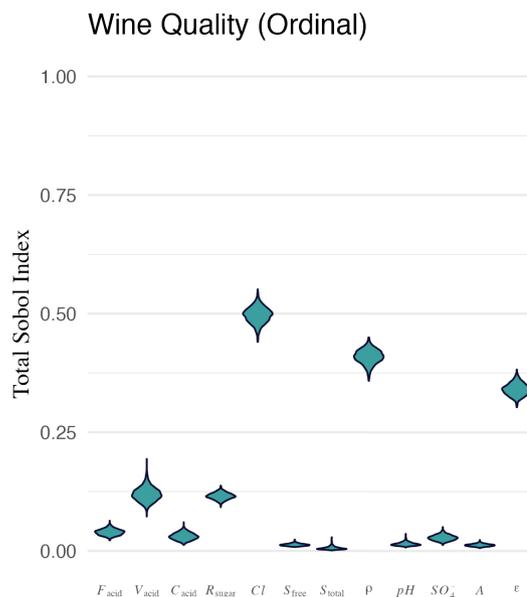
We illustrate the flexibility of KHAOS on two real datasets. The first is a physics-based computer model with  $p = 6$  inputs, which simulates an exploding cylinder with a gold liner; see K.

**TABLE 2** | Results for the simulation study in the high-noise ( $NSR = 0.5$ ) setting.

Method	Function	Avg. CRPS	Avg. time	Within 1% rate
KHAOS (ridge)	banana	12.498	7.032	1
KHAOS ( $g$ -prior)	banana	22.224	12.967	0
sparsePCE	banana	76.245	3.991	0
BART	banana	47.175	6.972	0
laGP	banana	63.553	20.782	0
KHAOS (ridge)	ishigami	0.402	8.118	0.100
KHAOS ( $g$ -prior)	ishigami	0.359	12.362	0.900
sparsePCE	ishigami	87,626	482.144	0
BART	ishigami	0.557	7.425	0
laGP	ishigami	0.795	21.441	0
KHAOS (ridge)	rabbits	0.035	7.516	0.100
KHAOS ( $g$ -prior)	rabbits	0.033	10.601	0.500
sparsePCE	rabbits	2.659	409.167	0
BART	rabbits	0.032	7.254	0.500
laGP	rabbits	0.050	23.226	0
KHAOS (ridge)	pollutant_uni	0.084	7.711	0.300
KHAOS ( $g$ -prior)	pollutant_uni	0.071	9.387	0.800
sparsePCE	pollutant_uni	0.413	635.362	0
BART	pollutant_uni	0.094	6.689	0.100
laGP	pollutant_uni	0.254	21.201	0
KHAOS (ridge)	friedman20	1.059	8.778	0.200
KHAOS ( $g$ -prior)	friedman20	0.900	9.112	0.400
sparsePCE	friedman20	2.531	294.623	0
BART	friedman20	0.860	7.368	0.400
laGP	friedman20	2.317	39.172	0



(a) Total Sobol indices for the Cylinder Experiments dataset, showing high sensitivity to the input  $r_1$ . The variance not explainable by KHAOS (due to  $\epsilon$ ) is negligible.



(b) Total Sobol indices for the Wine Quality dataset with ordinal response. Several variables are deemed important, to varying degrees, and a substantial amount of the latent variance is left unexplained.

**FIGURE 2** | (a) Total Sobol indices for the cylinder experiments dataset, showing high sensitivity to the input  $r_1$ . The variance not explainable by KHAOS (due to  $\epsilon$ ) is negligible. (b) Total Sobol indices for the Wine Quality dataset with ordinal response. Several variables are deemed important, to varying degrees, and a substantial amount of the latent variance is left unexplained.

N. Rumsey et al. (2025) for details. The second is the UCI white wine quality dataset, where the response is ordinal (Cortez et al. 2009).

For the ordinal data, we follow the latent Gaussian approach described by Hoff (2009), applying KHAOS to the latent space to enable Sobol decompositions of variance. This implementation is available in the `ordinal_khaos` function in the `khaos` package.

Figure 2a shows the total Sobol indices for the cylinder experiments, with dominant sensitivity to input  $r_1$  and negligible unexplained variance (denoted as  $\epsilon$  in the each subpanel of Figure 2). In contrast, Figure 2b shows that in the wine dataset, several inputs contribute meaningfully to the latent response, but a substantial portion of the variance remains unexplained.

## 6 | Conclusion

There are many effective emulators available, and no single method works best across all problems. As suggested by no-free-lunch theorems, emulator performance depends on the structure of the function, noise levels and the evaluation criteria. KHAOS is not a one-size-fits-all solution, but it is a robust and flexible tool that performs well across a range of settings.

Like other additive Bayesian methods (e.g., BASS, BPPR and BART), KHAOS models complex functions through structured basis expansions with full posterior inference. It builds on polynomial chaos ideas and naturally supports global sensitivity analysis via posterior Sobol indices (even in latent data settings). This leads to interpretable UQ alongside competitive

predictive accuracy. Future work might focus on extending the use of KHAOS for sensitivity studies via (e.g.,) Shapley effects (Owen 2014) or dimension reduction via (e.g.,) active subspaces (Constantine 2015; Rumsey, Francom, and Vander Wiel 2024).

The `khaos` R package fills a gap in the R ecosystem by providing a fully Bayesian PCE implementation with support for UQ and sensitivity analysis—tools that are useful in both emulator evaluation and scientific applications.

### Author Contributions

Kellin N. Rumsey implemented the software, conducted the numerical experiments, and prepared the initial manuscript draft. All authors contributed to the development of the ideas presented in this work. Devin Francom, Graham Gibson, and J. Derek Tucker provided input on the software design and implementation. All authors participated in manuscript revision and approved the final version.

### Acknowledgements

The authors thank the two anonymous reviewers for their constructive feedback and Dr. Thierry Mara for valuable discussions and correspondence during the development of this work. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. This work was supported by the NA-22 office of the National Nuclear Security Administration.

### Data Availability Statement

The data that support the findings of this study are available in the [Supporting Information](#) of this article.

## References

- Bliznyuk, N., D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan. 2008. "Bayesian Calibration and Uncertainty Analysis for Computationally Expensive Models Using Optimization and Radial Basis Function Approximation." *Journal of Computational and Graphical Statistics* 17, no. 2: 270–294.
- Bürkner, P.-C., I. Kröker, S. Oladshykhin, and W. Nowak. 2023. "A Fully Bayesian Sparse Polynomial Chaos Expansion Approach With Joint Priors on the Coefficients and Global Selection of Terms." *Journal of Computational Physics* 488: 112210.
- Chipman, H. A., E. I. George, and R. E. McCulloch. 2010. "BART: Bayesian Additive Regression Trees." *Annals of Applied Statistics* 4, no. 1: 266.
- Collins, G., D. Francom, and K. Rumsey. 2024. "Bayesian Projection Pursuit Regression." *Statistics and Computing* 34, no. 1: 29.
- Constantine, P. G. 2015. *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*. Vol. 2 SIAM.
- Cortez, P., A. Cerdeira, F. Almeida, T. Matos, and J. Reis. 2009. "Modeling Wine Preferences by Data Mining From Physicochemical Properties." *Decision Support Systems* 47, no. 4: 547–553.
- Denison, D. G., B. K. Mallick, and A. F. Smith. 1998. "Bayesian Mars." *Statistics and Computing* 8: 337–346.
- Francom, D., and B. Sansó. 2020. "Bass: An R Package for Fitting and Performing Sensitivity Analysis of Bayesian Adaptive Spline Surfaces." *Journal of Statistical Software* 94: 1–36.
- Francom, D., B. Sansó, A. Kupresanin, and G. Johannesson. 2018. "Sensitivity Analysis and Emulation for Functional Data Using Bayesian Adaptive Splines." *Statistica Sinica* 28: 791–816.
- Friedman, J. H. 1991. "Multivariate Adaptive Regression Splines." *Annals of Statistics* 19, no. 1: 1–67.
- Ghanem, R. G., P. D. Spanos, R. G. Ghanem, and P. D. Spanos. 1991. "Stochastic Finite Element Method: Response Statistics." In *Stochastic Finite Elements: A Spectral Approach*, 101–119.
- Gneiting, T., and A. E. Raftery. 2007. "Strictly Proper Scoring Rules, Prediction, and Estimation." *Journal of the American Statistical Association* 102, no. 477: 359–378.
- Gotelli, N. J., A. M. Ellison. 2004. *A Primer of Ecological Statistics*. Vol. 1 Sinauer Associates Sunderland.
- Gramacy, R. B., and D. W. Apley. 2015. "Local Gaussian Process Approximation for Large Computer Experiments." *Journal of Computational and Graphical Statistics* 24, no. 2: 561–578.
- Green, P. J., and A. Mira. 2001. "Delayed Rejection in Reversible Jump Metropolis–Hastings." *Biometrika* 88, no. 4: 1035–1053.
- Hoff, P. D. 2009. *A First Course in Bayesian Statistical Methods*. Vol. 580. Springer.
- Ishigami, T., and T. Homma. 1990. "An Importance Quantification Technique in Uncertainty Analysis for Computer Models." In *In: [1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis*, 398–403.
- Liang, F., R. Paulo, G. Molina, M. A. Clyde, and J. O. Berger. 2008. "Mixtures of g Priors for Bayesian Variable Selection." *Journal of the American Statistical Association* 103, no. 481: 410–423.
- Lüthen, N., S. Marelli, and B. Sudret. 2021. "Sparse Polynomial Chaos Expansions: Literature Survey and Benchmark." *SIAM/ASA Journal on Uncertainty Quantification* 9, no. 2: 593–649.
- McKay, M. D., R. J. Beckman, and W. J. Conover. 1979. "Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code." *Technometrics* 21, no. 2: 239–245.
- Nature Methods Editorial. 2011. "NUAP (No Unnecessary Acronyms Please)." *Nature Methods* 8: 521. <https://doi.org/10.1038/nmeth.1646>.
- Nott, D. J., A. Y. Kuk, and H. Duc. 2005. "Efficient Sampling Schemes for Bayesian Mars Models With Many Predictors." *Statistics and Computing* 15: 93–101.
- Novak, L., and D. Novak. 2018. "Polynomial Chaos Expansion for Surrogate Modelling: Theory and Software." *Beton- Und Stahlbetonbau* 113: 27–32.
- O'Hagan, A. 2013. "Polynomial Chaos: A Tutorial and Critique From a Statisticians Perspective Polynomial Chaos: A Tutorial and Critique From a Statisticians Perspective." *SIAM/ASA Journal on Uncertainty Quantification* 20: 1–20.
- Owen, A. B. 2014. "Sobol' Indices and Shapley Value." *SIAM/ASA Journal on Uncertainty Quantification* 2, no. 1: 245–251.
- Pan, Q., X. Qu, L. Liu, and D. Dias. 2020. "A Sequential Sparse Polynomial Chaos Expansion Using Bayesian Regression for Geotechnical Reliability Estimations." *International Journal for Numerical and Analytical Methods in Geomechanics* 44, no. 6: 874–889.
- Robert, C. P. 2007. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Vol. 2. Springer.
- Rumsey, K. (2023). *duqling* (Tech. Rep.). Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).
- Rumsey, K., D. Francom, and S. Vander Wiel. 2024. "Discovering Active Subspaces for High-Dimensional Computer Models." *Journal of Computational and Graphical Statistics* 33, no. 3: 896–908.
- Rumsey, K. N., D. Francom, and A. Shen. 2024. "Generalized Bayesian Mars: Tools for Stochastic Computer Model Emulation." *SIAM/ASA Journal on Uncertainty Quantification* 12, no. 2: 646–666.
- Rumsey, K. N., Z. K. Hardy, C. Ahrens, and S. Vander Wiel. 2025. "Co-Active Subspace Methods for the Joint Analysis of Adjacent Computer Models." *Technometrics* 67, no. 1: 133–146.
- Shao, Q., A. Younes, M. Fahs, and T. A. Mara. 2017. "Bayesian Sparse Polynomial Chaos Expansion for Global Sensitivity Analysis." *Computer Methods in Applied Mechanics and Engineering* 318: 474–496.
- Sobol, I. M. 2001. "Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates." *Mathematics and Computers in Simulation* 55, no. 1–3: 271–280.
- Sudret, B. 2008. "Global Sensitivity Analysis Using Polynomial Chaos Expansions." *Reliability Engineering & System Safety* 93, no. 7: 964–979.
- Wiener, N. 1938. "The Homogeneous Chaos." *American Journal of Mathematics* 60, no. 4: 897–936.
- Xiu, D. and G. E. Karniadakis. 2002. "The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations." *SIAM Journal on Scientific Computing* 24, no. 2: 619–644.
- Zellner, A. 1986. "On Assessing Prior Distributions and Bayesian Regression Analysis With G-Prior Distributions." *Bayesian Inference and Decision Techniques*.
- Zhang, H., X. Huang, J. Gan, W. Karmaus, and T. Sabo-Attwood. 2016. "A Two-Component G-Prior for Variable Selection." *Bayesian Analysis* 11, no. 2: 353–380.

## Supporting Information

Additional supporting information can be found online in the Supporting Information section. [khaos\\_SM.pdf](#).