International Journal for Uncertainty Quantification, 14(4):83–103 (2024)

EXTREME LEARNING MACHINES FOR VARIANCE-BASED GLOBAL SENSITIVITY ANALYSIS

John E. Darges,[1,*] Alen Alexanderian,[1] & Pierre A. Gremaud[1,2]

[1]*Department of Mathematics, North Carolina State University, Raleigh, NC*
[2]*The Graduate School, North Carolina State University, Raleigh, NC*

*Address all correspondence to: John E. Darges, Department of Mathematics, North Carolina State University, Raleigh, 27607, NC, E-mail: jedarges@ncsu.edu

Original Manuscript Submitted: 7/3/2023; Final Draft Received: 2/24/2024

Variance-based global sensitivity analysis (GSA) can provide a wealth of information when applied to complex models. A well-known Achilles' heel of this approach is its computational cost, which often renders it unfeasible in practice. An appealing alternative is to instead analyze the sensitivity of a surrogate model with the goal of lowering computational costs while maintaining sufficient accuracy. Should a surrogate be "simple" enough to be amenable to the analytical calculations of its Sobol' indices, the cost of GSA is essentially reduced to the construction of the surrogate. We propose a new class of sparse-weight extreme learning machines (ELMs), which, when considered as surrogates in the context of GSA, admit analytical formulas for their Sobol' indices and, unlike the standard ELMs, yield accurate approximations of these indices. The effectiveness of this approach is illustrated through both traditional benchmarks in the field and on a chemical reaction network.

KEY WORDS: *uncertainty quantification, machine learning, stochastic sensitivity analysis, adaptivity*

1. INTRODUCTION

A key insight into the behavior of a generic model of the following form:

$$y = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d,\ y \in \mathbb{R}, \tag{1}$$

is to understand the impact of the uncertainty in the entries of $\boldsymbol{x} = (x_1, \ldots, x_d)$ on the uncertainty in the model output y; this can be addressed by performing global sensitivity analysis (GSA) [1–3]. We focus on variance-based GSA [4–6] whereby one seeks to quantify the relative contributions of the entries of $\boldsymbol{x}$ to the variance of f. Specifically, we rely on the Sobol' indices [5]. While the method we introduce is applicable for a broader class of distributions, we assume that the entries of $\boldsymbol{x}$ in Eq. (1) can be regarded as independent uniformly distributed random variables, which is common in applications.

The standard approach to computing Sobol' indices uses pick-freeze estimators [1,7]. This generally involves a costly sampling procedure and multiple applications of Monte Carlo integration. A large input dimension d, common in engineering applications, increases the cost dramatically. Under typical cost constraints, we cannot afford the number of samples required for reliable estimation.

Often, an initial parameter screening procedure can be applied to reduce the input dimension, before a more detailed variance-based GSA is conducted; see e.g., [8]. Nonetheless, when evaluating the model f is prohibitively expensive, Monte Carlo methods for computing Sobol' indices should be avoided. In such cases, a widely used approach is to construct a surrogate model $\hat{f} \approx f$, whose Sobol' indices can be computed efficiently [9,10].

A number of surrogate models have been proposed for accelerating variance-based GSA. These include polynomial chaos (PC) expansions (PCE) [11,12], multivariate adaptive regression splines (MARS) [13,14], Gaussian processes [15–17], Bayesian adaptive regression trees (BART) [18], random forests [19], support vector machines [20], artificial neural networks (ANNs) [21], autoregressive models [22], and radial basis functions (RBF) [23]. See [9] and [24] for an overview of surrogate-based methods for GSA.

2152–5080/24/$35.00 © 2024 by Begell House, Inc. www.begellhouse.com

In surrogate-based approaches, the user builds the surrogate model and estimates the surrogate's Sobol' indices. Apart from special surrogate models, which admit analytic formulas for Sobol' indices, surrogate models require Monte Carlo sampling to estimate Sobol' indices. This process introduces two sources of error to the Sobol' index estimates. The first one is due to the approximation error of the surrogate, and the second one is due to the error in estimating the surrogate's Sobol' indices. Sampling of the surrogate is often considerably cheaper than sampling the underlying model. However, this sampling can still be costly. In the best case, analytic formulas for the surrogate's Sobol' indices are known. Then, the issues caused by Monte Carlo sampling disappear altogether. Certain statistical surrogates, such as Gaussian processes [16,17] and BART [18], offer analytic Sobol' indices along with accompanying uncertainty bounds. The statistical nature of these surrogates, however, means sampling is needed to compute Sobol' indices. This could be potentially costly, especially compared to certain numerical surrogate models for which Sobol' indices can be computed analytically for practically no additional cost [9]. Examples of such approaches include PCEs [11,12] or RBFs [25,26].

Sobol' indices of PCE surrogates can be expressed analytically in terms of the PC coefficients [11,12]. Unsurprisingly, PCE has become a workhorse method for computing Sobol' indices and uncertainty analysis in general. This has spawned a number of methods for computing and using PCEs for uncertainty analysis. Nonintrusive project-based approaches involve computing the PC coefficients using quadrature. For high-dimensional inputs, this is untenable and motivates regression-based approaches. A major challenge with PCE is that the PC basis can grow to gargantuan sizes as the degree of the expansion is increased. This necessitates sparse regression methods to reduce the size of the basis [27]. A survey of these methods can be found in [28]. We also point to [29–33], for recent developments regarding PCE-based approaches for uncertainty quantification.

In this article, we present a new approach that uses extreme learning machine (ELM) [34,35] surrogates—a class of neural networks—for which Sobol' indices can be computed analytically. The existing surrogate modeling approaches that have this feature each have their strengths and weaknesses, making them suitable for specific applications of interest. When studying black box models, it is seldom obvious which surrogate is the best choice. In such applications where one does not know the true values of the sensitivity indices, it is beneficial to have access to different tools to compute the indices. This can be useful for confirming the reliability of one's results. An example of this is [36], where comparing the Sobol' indices of PCE and ANN surrogates provides verifiable GSA. Notably, however, the ANN surrogate requires more resources to produce GSA results, partly because Sobol' indices cannot be computed analytically. Our contribution expands the limited toolbox of fast surrogate models with analytic formulas for Sobol' indices. Having ready access to such analytic formulas enables the users to compute these indices without any costly Monte Carlo sampling procedures.

Single-layer neural networks (SLNNs) constructed via ELM are random weight neural networks. Unlike in traditional neural networks, one draws weights and biases of the hidden layer randomly. Consequently, training amounts to solving a linear least-squares problem to estimate the output layer weights; see Section 2 for more details. Such methods are more suitable than tuning all parameters in the neural network for certain UQ applications. When model evaluations are expensive, we can only afford to generate a scarce amount of data. The amount of generated data is likely less than the number of tunable parameters, meaning training the neural network is an underdetermined problem. Random weight methods avoid this issue.

In the field of machine learning, there are several different random weight approaches. The differences between them are subtle. Some of the first implementations were Schmidt neural networks [37] and the random vector functional link (RVFL) [38,39]. The neural network ELM [34] emerged later as one of the most popular random weight methods. Helpful surveys of these approaches are, e.g., [40–42]. An intimately related area is random features, introduced by [43]. Random features originated as a randomized approach to kernel methods. Their connection to neural networks was noted early on [44,45]. For certain classes of kernels, random feature expansions and random weight neural networks are equivalent. Random feature expansions are seeing increasing use in applied mathematics; see [46,47].

Our motivation for considering neural network ELMs for GSA is twofold:

- Through a judicious choice of the activation function, analytic formulas for Sobol' indices can be derived; see Section 3.

- They are simple and inexpensive to train: only a linear least-squares problem needs to be solved.

Also, ELMs provide a flexible surrogate modeling framework; they do not make strong demands on the distribution of the inputs or the regularity of the input-output map. Additionally, ELMs are well suited to tackle nonlinear and high-dimensional models. The use of neural networks for surrogate-based uncertainty quantification has been the object of recent research efforts. Among many others, Refs. [48–53] have, for instance, used neural network surrogates for GSA. To our knowledge, the present work is the first of its kind to derive analytic formulas for Sobol' indices of neural network surrogates.

To be used reliably for a variance-based GSA, the surrogate $\hat{f}$ must capture some key structural properties of the exact model f. To illustrate this point, consider Eq. (1) with $\boldsymbol{x} \in \mathbb{R}^3$ and assume f has a mean of zero. The analysis of variance (ANOVA) decomposition of f is given by

$$f(x_1, x_2, x_3) = \mathbb{E}(f) + \sum_{i=1}^{3} f_i(x_i) + f_{12}(x_1, x_2) + f_{13}(x_1, x_3) + f_{23}(x_2, x_3) + f_{123}(x_1, x_2, x_3), \tag{2}$$

where $f_i = \mathbb{E}(f|x_i)$, $f_{ij} = \mathbb{E}(f|x_i, x_j) - f_i - f_j$, and $f_{123} = f - \sum_{i=1}^{3} f_i - \sum_{i<j} f_{ij}$. A key observation is that the variance of f can be decomposed as the sum of the variances of the individual terms in the ANOVA. Here, the terms $\{f_i\}_{i=1}^{d}$ represent contributions from main effects while the remaining terms capture contributions from variable interactions. This makes it possible to quantify the contributions of the individual inputs (or a group of inputs) to the total variance of f and leads to definition of Sobol' indices. To provide accurate estimates of Sobol' indices, a surrogate $\hat{f}$ must emulate the main effect terms as well as the variable interaction terms in Eq. (2). As discussed in Section 4, standard ELMs may fail to correctly capture the impact of variable interactions on output variance, leading to inaccurate Sobol' indices. We resolve this shortcoming by introducing sparsity to the hidden layer weights. The sparsification technique is related to DropConnect, a regularization method in machine learning [54]. DropConnect introduces sparsity to neural networks by randomly setting weights to zero before training. This has been used to enhance RVFLs [55] and ELMs [56]. Yet another closely related approach was proposed in [57] for random feature expansions. In that work, the purpose of adding sparsity is to efficiently emulate models with lower order variable interactions. Our implementation preserves the speed and simplicity of ELM while also allowing the surrogate to better incorporate model behavior.

We demonstrate the efficiency of the proposed approach on standard benchmark problems, an application from biochemistry, and a high-dimensional problem; see Section 5.

2. EXTREME LEARNING MACHINES

A SLNN with n hidden layer neurons and a scalar output has the following form:

$$g(\boldsymbol{x}) = \sum_{j=1}^{n} \beta_j \phi(\boldsymbol{w}_j^\top \boldsymbol{x} + b_j), \tag{3}$$

where $\boldsymbol{w}_j \in \mathbb{R}^d$, $b_j \in \mathbb{R}$, $j = 1, \ldots, n$ are the weights and biases of the hidden layer; β_j, $j = 1, \ldots, n$ are the weights of the output layer; and ϕ is the hidden layer activation function. For a given number of neurons n and a given activation function ϕ, we set

$$\mathcal{M}_n(\phi) = \left\{ \sum_{j=1}^{n} \beta_j \phi(\boldsymbol{w}_j^\top \boldsymbol{x} + b_j) \,:\, b_j, \beta_j \in \mathbb{R}, \boldsymbol{w}_j \in \mathbb{R}^d \right\}. \tag{4}$$

It is known that SLNNs are universal approximators since, when using non-polynomial continuous activation functions, they are dense in $\mathcal{C}(\mathbb{R}^d)$ [58]. More precisely, if $\phi \in \mathcal{C}(\mathbb{R})$ is not polynomial and $f \in \mathcal{C}(K)$, where K is a compact subset of $\mathbb{R}^d$. Then, for any $\epsilon > 0$, there exists n and $g_n \in \mathcal{M}_n(\phi)$, such that

$$\max_{\boldsymbol{x} \in K} |f(\boldsymbol{x}) - g_n(\boldsymbol{x})| < \epsilon. \tag{5}$$

In addition, SLNNs are universal approximators on $L^p(\mathbb{R}^d)$, for $p \in [1, \infty)$, when the activation function is (almost everywhere) not a polynomial [59]. The standard approach to train Eq. (3) is to determine all hidden layer weights and biases and output weights by solving a nonlinear least-squares problem.

In neural network ELMs, the weight vectors $\boldsymbol{w}_j$ and biases b_j, $j = 1, \ldots, n$, of the hidden layer are not determined as part of a regression process but rather are *chosen randomly*. Training only involves determining the output layer weights $\{\beta_j\}_{j=1}^n$ by solving a *linear* least-squares problem.

Remarkably, even though it bypasses training the hidden layer weights (and replaces a costly nonlinear least-squares problem by a linear one), ELMs are (almost surely) universal approximators on $L^2(\mathbb{R}^d)$. Namely, by Theorem 2.4 in [35], ELMs have a universal approximation property (5) on $L^2(\mathbb{R}^d)$ provided: (i) the hidden layer weights and biases are sampled from a continuous probability distribution; (ii) the activation function is piecewise continuous and not a polynomial (almost everywhere); and (iii) the output weights are determined by ordinary least-squares.

2.1 Computing ELM Surrogates

Let

$$\mathbf{W} = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \cdots & \boldsymbol{w}_n \end{bmatrix}^\top,$$
$$\boldsymbol{b} = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}^\top,$$
$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_n \end{bmatrix}^\top.$$

The ELM [Eq. (3)] takes the following form:

$$g(\boldsymbol{x}) = \boldsymbol{\beta}^\top \phi(\mathbf{W}\boldsymbol{x} + \boldsymbol{b}), \tag{6}$$

where ϕ is understood to act componentwise. The entries of the weight matrix, $\mathbf{W}$, and bias vector, $\boldsymbol{b}$, are sampled independently from a continuous probability distribution, $\mathcal{D}$. It is important to note that this is done before the training, and a fixed realization of $\mathbf{W}$ and $\boldsymbol{b}$ is used in all subsequent computations.

To construct a surrogate, the model f is sampled at $\{\boldsymbol{x}_i\}_{i=1}^m$, yielding $y_i = f(\boldsymbol{x}_i)$, $i = 1, \ldots, m$. We then find $\boldsymbol{\beta}$ by solving

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \|\mathbf{H}\boldsymbol{\beta} - \boldsymbol{y}\|_2, \tag{7}$$

where $\boldsymbol{y} = [y_1 \quad \cdots \quad y_m]^\top$ and $H_{ij} = \phi(\boldsymbol{w}_j^\top \boldsymbol{x}_i + b_j)$ for $i, j \in \{1, \ldots, m\} \times \{1, \ldots, n\}$.

In our computations, we implement a regularized least-squares problem to control the magnitude of the solution $\boldsymbol{\beta}$,

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{H}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \frac{\alpha}{2}\|\boldsymbol{\beta}\|_2^2, \tag{8}$$

with $\alpha > 0$, the solution of which is

$$\boldsymbol{\beta}^* = (\mathbf{H}^\top \mathbf{H} + \alpha \mathbf{I})^{-1} \mathbf{H}^\top \boldsymbol{y}.$$

The regularization parameter α can be selected using the L-curve method or generalized cross validation [60]. We construct the hidden layer weight matrix and bias vector by sampling individual values from the standard normal distribution and use Latin hypercube sampling (LHS) to sample points in the training set $\{\boldsymbol{x}_i\}_{i=1}^m$.

3. GLOBAL SENSITIVITY ANALYSIS USING ELMS

For each entry x_k, $k = 1, \ldots, d$, the first-order Sobol' index S_k and total Sobol' index S_k^{tot} of a surrogate $y = \hat{f}(\boldsymbol{x})$ are defined as follows:

$$S_k = \frac{\text{var}(\hat{f}_k)}{\text{var}(\hat{f})}, \quad S_k^{\text{tot}} = 1 - \frac{\text{var}(\mathbb{E}(\hat{f}|x_l,\ l \neq k))}{\text{var}(\hat{f})}, \tag{9}$$

where $\hat{f}_k(x_k) = \mathbb{E}(\hat{f}|x_k) - \mathbb{E}(\hat{f})$; see, e.g., [6]. Henceforth, we assume that the entries of the input vector $\boldsymbol{x}$ are independent and uniformly distributed on the interval $[0,1]$ and, therefore, the input domain is $[0,1]^d$. It is straightforward to extend the proposed approach to the case where entries of $\boldsymbol{x}$ are independent uniformly distributed random variables on arbitrary closed and bounded intervals.

To obtain analytic formulas for the Sobol' indices while avoiding Monte Carlo approximations, we must be able to easily compute the mean, variance, and partial variances of the surrogate. By carefully choosing the activation function, we can design $\hat{f}$ as an ELM that (i) suits the abovementioned requirement and (ii) preserves the universal approximation property. Activation functions traditionally used in machine learning are not suited for our purpose as the corresponding calculations in Eq. (9) become impractical or impossible. Instead, we choose an exponential activation function

$$\phi(t) = e^t,$$

which results in

$$\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{n} \beta_j e^{\boldsymbol{w}_j^\top \boldsymbol{x} + b_j} = \sum_{j=1}^{n} \left(\beta_j e^{b_j} \prod_{l=1}^{d} e^{w_{j,l} x_l} \right). \tag{10}$$

Since ϕ is a continuous non-polynomial function, the results cited in Section 2 apply and ELMs constructed this way still have the universal approximation property. ELMs with exponential activation function can be viewed as an approximation by sums of exponential functions, which is well-studied [61,62]. Still, one could ask whether other choices of activation function would provide more accurate ELM surrogates. For the numerical examples considered in this paper, we do not observe a clear disadvantage in terms of accuracy for ELM surrogates with exponential activation function compared to other activation functions. Moreover, our primary concern is whether the ELM surrogate yields accurate Sobol' index approximations. Still, the impact of the choice of activation function on the surrogate accuracy merits further study.

We now compute the first and second moments of Eq. (10).

Lemma 1. *The mean and variance of the ELM [Eq. (10)] are*

$$\mathbb{E}(\hat{f}) = \sum_{j=1}^{n} \left(\beta_j e^{b_j} \prod_{l=1}^{d} \epsilon_l(w_{j,l}) \right),$$

and

$$\operatorname{var}(\hat{f}) = \sum_{j,i=1}^{n} \beta_j \beta_i e^{b_j + b_i} \left(\prod_{l=1}^{d} \epsilon_l(w_{j,l} + w_{i,l}) - \prod_{r=1}^{d} \epsilon_r(w_{j,r}) \epsilon(w_{i,r}) \right),$$

where

$$\epsilon_k(t) = \int_{\mathbb{R}} e^{t x_k} \pi_k(x_k)\, dx_k. \tag{11}$$

Here, π_k denotes the probability density function of x_k.

Proof. See Appendix A. □

Using Lemma 1, we obtain analytic expressions for the Sobol' indices of Eq. (10).

Proposition 1. *Let ϵ_k be the function defined in Eq. (11). The first-order and total Sobol' indices of the ELM [Eq. (10)] are given by*

$$S_k = \frac{1}{\operatorname{var}(\hat{f})} \sum_{j,i=1}^{n} \beta_j \beta_i e^{b_j + b_i} (\epsilon_l(w_{j,k} + w_{i,k}) - \epsilon_l(w_{j,k}) \epsilon_l(w_{i,k})) \prod_{l \neq k} \epsilon_l(w_{j,l}) \epsilon_l(w_{i,l}), \tag{12}$$

and

$$S_k^{\text{tot}} = 1 - \frac{1}{\text{var}(\hat{f})} \sum_{j,i=1}^{n} \beta_j \beta_i e^{b_j + b_i} \epsilon_k(w_{j,k}) \epsilon_k(w_{i,k}) \left(\prod_{r \neq k} \epsilon_r(w_{j,r} + w_{i,r}) - \prod_{l \neq k} \epsilon_r(w_{j,r}) \epsilon_l(w_{i,l}) \right), \tag{13}$$

respectively, for $k = 1, \ldots, d$.

Proof. See Appendix A. □

We have access to explicit formulas whenever we know the expressions of ϵ_k, for $k = 1, \ldots, d$. Example distributions that admit expressions include the following:

- Uniform distribution: $x_k \sim \mathcal{U}([a, b])$, then $\epsilon_k(t) = \begin{cases} (e^{bt} - e^{at})/t, & t \neq 0 \\ 1, & t = 0 \end{cases}$.
- Gaussian distribution: $x_k \sim \mathcal{N}(\mu, \sigma^2)$, then $\epsilon_k(t) = e^{\mu t + \sigma^2 t^2/2}$.
- Gamma distribution: $x_k \sim \Gamma(\alpha, \beta)$, then $\epsilon_k(t) = \begin{cases} \beta^{\alpha}(t - \beta)^{\alpha - 1}, & t \neq 0 \\ 1, & t = 0 \end{cases}$.

The inputs may follow different distributions and can be a mix of Gaussian and uniform random variables, for example.

4. SPARSE WEIGHT ELMS

ELM surrogates [Eq. (6)] are constructed by randomly sampling the hidden layer weight matrix $\mathbf{W}$ and bias vector $\boldsymbol{b}$ from the standard normal distribution. As pointed out in Section 2, an ELM surrogate $\hat{f}$ can be found to satisfy

$$\hat{f} \approx f \tag{14}$$

as accurately as desired. Our goal is however not so much to construct $\hat{f}$ satisfying Eq. (14) within a given tolerance but rather to construct $\hat{f}$ such that

$$S(\hat{f}) \approx S(f), \tag{15}$$

where S stands here for any of the Sobol' indices from Proposition 1 and $S(\hat{f})$ can be computed at low cost. To ensure that the Sobol' indices computed from $\hat{f}$ yield a reliable description of the variable interactions, we thus have to construct a surrogate $\hat{f}$ that not only satisfies Eq. (14) but also,

$$\hat{f}_{\boldsymbol{u}} \approx f_{\boldsymbol{u}}, \quad \text{for any } \boldsymbol{u} \subset \{1, \ldots, d\}, \tag{16}$$

were $f_{\boldsymbol{u}}$ and $\hat{f}_{\boldsymbol{u}}$ in the ANOVA decomposition of f and $\hat{f}$, respectively, are given by

$$f = \sum_{\boldsymbol{u} \subseteq \{1, \ldots, d\}} f_{\boldsymbol{u}}, \quad f_{\boldsymbol{u}} = \sum_{\boldsymbol{v} \subseteq \boldsymbol{u}} (-1)^{|\boldsymbol{u}| - |\boldsymbol{v}|} \mathbb{E}\Big(\hat{f} | x_l,\ l \in \boldsymbol{v} \Big), \tag{17}$$

corresponding to the subset $\boldsymbol{u}$.

4.1 ELMs and Variable Interactions

Simple examples show that the amount of variable interactions in f plays a key role in either achieving or failing to achieve Eq. (15). As an illustration, consider the following parametrized function:

$$f_\delta(\boldsymbol{x}) = \sum_{i=1}^{d} x_i + \delta \prod_{j=1}^{d} (1 + x_j), \quad \boldsymbol{x} \in [0, 1]^d, \tag{18}$$

where δ controls the amount of variable interactions. When $\delta = 0$, f_δ is fully additive; there are no variable interactions and thus the first-order and corresponding total Sobol indices are equal. For $\delta > 0$, there are interactions between the variables that increase in importance as δ increases. By construction, we also observe that the individual entries x_i, $i = 1, \ldots, d$, all share the same Sobol' indices. This feature is not present when considering the ELM surrogate $\hat{f}$ instead of f because of approximation errors. The first-order and total Sobol' indices of Eq. (18) can be found analytically and are given in Appendix B. Therefore, by varying δ, we can explore how variable interactions affect accuracy in both Eqs. (14) and (15).

To approximate the Sobol' indices of Eq. (18) with $d = 15$, we construct ELMs of the form Eq. (10), as follows:

1. Collect $m = 900$ training points sampled by LHS.
2. Construct ELMs with $n = 300$ neurons.
3. Compute the surrogate relative error

$$E_{\text{surr}} = \frac{1}{\sqrt{\sum_{j=1}^{s} y_j^2}} \sqrt{\sum_{j=1}^{s} (\hat{f}(\boldsymbol{x}_j) - y_j)^2}, \quad y_j = f(\boldsymbol{x}_j) \quad j = 1, \ldots, s, \tag{19}$$

   using $s = 1000$ validation points sampled by LHS.
4. Select a regularization parameter by the L-curve method.
5. Compute Sobol' indices of the ELM surrogate using Eqs. (12) and (13).

For each choice of δ, the L-curve method selects $\alpha = 10^{-3}$ as the regularization parameter.

Figure 1 (top left) illustrates that, unsurprisingly, the surrogate relative error, which measures the discrepancy in Eq. (14), increases as the amount of interactions increases with δ. Interestingly, the situation is reversed when looking at the accuracy of the Sobol' indices through Eq. (15). Indeed, Fig. 1 (bottom left) shows that, while $S_i(\hat{f})$ is a close approximation of $S_i(f)$, for any $i = 1, \ldots, d$, at the larger values of δ, the relative accuracy of the corresponding approximations decreases for smaller levels of interaction (i.e., smaller values of δ). The variance of the $S_i(\hat{f})$'s, which again should ideally all have the value $S_1(f) = \cdots = S_d(f)$, also increases with smaller values of δ. The total Sobol' indices and their approximations largely behave in similar fashion, see Fig. 1 (bottom right). This example shows that ELM-based surrogates have a tendency to overestimate variable interactions, a point made clear in Fig. 1 (top right), where the interaction indices $I_i := S_i^{\text{tot}} - S_i$, $i = 1, \ldots, d$ are considered.

These results demonstrate that, for variance-based GSA, standard ELM may fail to effectively adapt when applied to models featuring different degrees of contribution from interaction terms to the output variance.

4.2 Hidden Layer Weight Sparsification

We propose sparsifying the hidden layer weight matrix $\mathbf{W}$ before training the output weights as a means of controlling the influence of variable interactions. Specifically, we replace $\mathbf{W}$ by a sparse weight matrix $\widetilde{\mathbf{W}}$ defined as

$$\widetilde{\mathbf{W}} = \mathbf{B} \circ \mathbf{W}, \tag{20}$$

where $\mathbf{B}$ is a $(n \times d)$-matrix with independent Bernoulli entries, i.e.,

$$B_{ij} = \begin{cases} 0 & \text{with probability } p \\ 1 & \text{with probability } 1 - p \end{cases}, \tag{21}$$

and where $\circ$ stands for elementwise matrix multiplication. The sparsification parameter $p \in [0, 1)$ determines how sparse the hidden layer weight matrix is. If $p = 0$, then $\widetilde{\mathbf{W}} = \mathbf{W}$ and the method reverts to standard ELM; when p

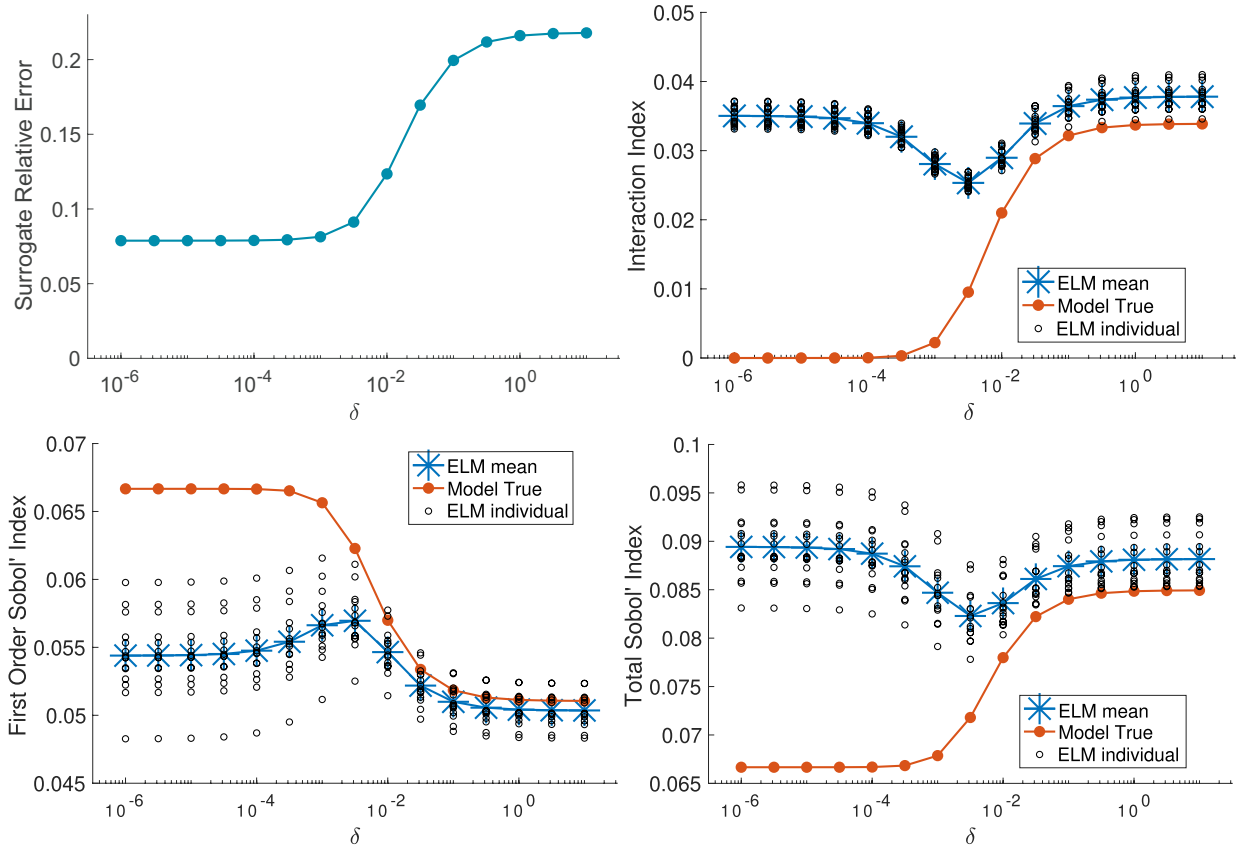

FIG. 1: Example (18) with $d = 15$. Top left: Surrogate relative error; bottom left: first-order Sobol' indices $S_i(f)$ and $S_i(\hat{f})$ for x_i, $i = 1, \ldots, 15$ and corresponding mean; bottom right: total Sobol' indices $S_i^{\text{tot}}(f)$ and $S_i^{\text{tot}}(\hat{f})$ of x_i, $i = 1, \ldots, 15$, and corresponding mean; top right: interaction indices $I_i := S_i^{\text{tot}} - S_i$, $i = 1, \ldots, 15$, for both f and $\hat{f}$. In the true model with fixed δ, indices $S_i(f)$ are equal for all $i = 1, \ldots, 15$. The "ELM individual" values provide the distribution of estimates $S_i(\hat{f})$, and "ELM mean" is the mean of these estimates at a given δ.

is selected to be near 1, the weight matrix $\widetilde{\mathbf{W}}$ is sparse. This technique strays from the ELM theory, which assumes sampling weights from a continuous probability distribution to guarantee universal approximation.

We implement the sparse-weight ELM (SW-ELM) approach, i.e., ELM with $\widetilde{\mathbf{W}}$ instead of $\mathbf{W}$ as a hidden layer weight matrix, on example (18). Figure 2 illustrates the results for a range of values of the sparsification parameter p and for three values of δ corresponding to low variable interactions ($\delta = 10^{-8}$), limited variable interactions ($\delta = 10^{-3}$), and strong variable interactions ($\delta = 10^{8}$). In the case of low variable interactions ($\delta = 10^{-8}$), Fig. 2 shows that using a sparse weight matrix can dramatically increase the accuracy of both the ELM surrogate and of the Sobol' indices computed using that surrogate. Sparsifying the weight matrix may thus lead to significant improvements in GSA results when dealing with models with few interaction terms. On the other hand, in cases where interaction terms are prominent in the model, sparsifying the hidden layer weight matrix may offer no improvement or may even result in loss of accuracy.

4.3 Selection of Sparsification Parameter

Figure 2 suggests a framework for how to implement SW-ELM for a given model. Indeed, for fixed values of δ and varying values of p, we observe similar trends in the surrogate error and in the total Sobol' index approximation error. This indicates that we may use the surrogate error as a guideline when searching for the best sparsification parameter.

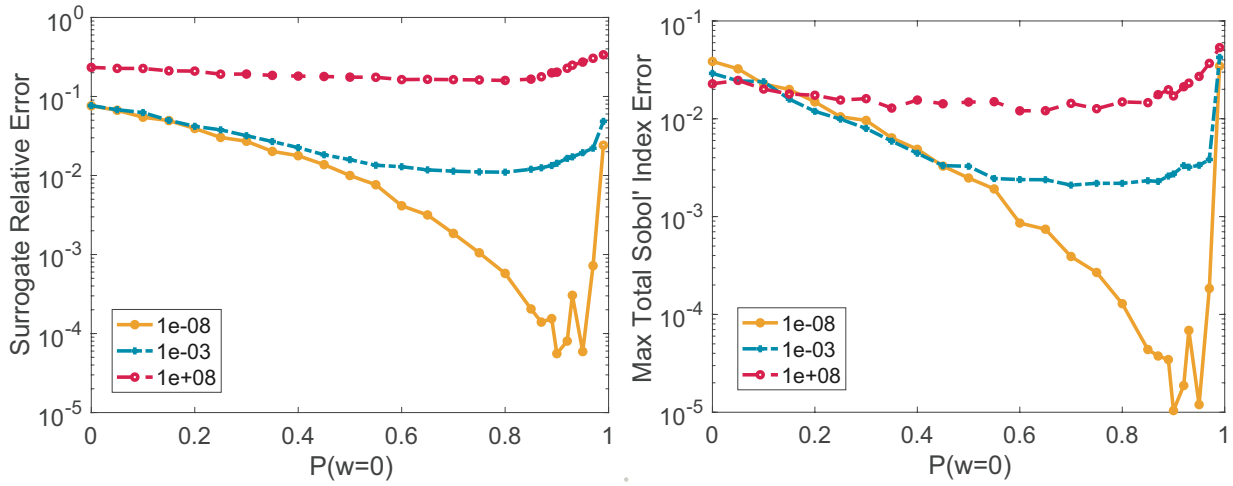

FIG. 2: Example (18) with $d = 15$, with low variable interactions $\delta = 10^{-8}$, limited variable interactions 10^{-3}, and strong variable interactions 10^{8}; surrogate relative error (left) and largest absolute error in total Sobol' index approximation (right)

We propose creating a validation set alongside the training set when using SW-ELM surrogates for GSA. Subsequently, we select r values of the sparsification parameter p and construct corresponding sparse weight matrices. Then, ELMs are trained using each weight matrix and the validation set is used to compute the surrogate error for each ELM. This serves as a diagnostic tool to aid in deciding whether sparsification may improve Sobol' index approximation. If sparsification yields an improved surrogate error, then we use the SW-ELM with the lowest surrogate error to approximate Sobol' indices. If, on the other hand, sparsification does not yield a notable improvement then we may default to using standard ELM. We summarize the method in Algorithm 1.

We illustrate the power of this method by repeating the experiment in Fig. 1 for Eq. (18) using SW-ELM instead of ELM. Figure 3 displays the total Sobol' indices for both surrogates as well as the exact indices. SW-ELM displays increased accuracy and reduced variance across all tested ranges of interaction strength.

In the context of GSA, SW-ELM displays better accuracy and flexibility than standard ELM. The proposed implementation is only marginally more costly than standard ELM because there is no need to resample training or validation points for each ELM.

Algorithm 1: GSA with sparse weight ELM

Input: (i) Model f; (ii) training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^m$; (iii) validation set $\{\boldsymbol{x}'_j, y'_j\}_{j=1}^s$; (iv) number of neurons n; and (v) candidate sparsification values $\{p_l\}_{l=1}^r$ (with $p_1 = 0$).

Output: (i) First-order Sobol' indices $\{S_k\}_{k=1}^n$ and (ii) total Sobol' indices $\{S_k^{\text{tot}}\}_{k=1}^n$.

Generate weight matrix $\mathbf{W}_0$ and bias vector $\boldsymbol{b}$ using a standard normal distribution

for $l = 1, \ldots, r$ **do**

  Construct $\mathbf{W}_l = \mathbf{B} \circ \mathbf{W}_0$ with $\mathbf{B}$ from Eq. (21) with $p = p_l$.

  Determine regularization parameter α_l by the L-curve method.

  Find output weights $\boldsymbol{\beta}_l$ by training ELM using $\mathbf{W}_l, \boldsymbol{b}, \alpha_l$ [see Eq. (8)].

  Compute relative surrogate error $E_l = E_{\text{surr}}$ [see Eq. (19)].

end for

Select hidden layer weight matrix $\mathbf{W}$ and output weights $\boldsymbol{\beta}$ corresponding to sparsification parameter that gives smallest relative surrogate error.

With $\mathbf{W}$, $\boldsymbol{\beta}$, and $\boldsymbol{b}$, compute first-order Sobol' indices $\{S_k\}_{k=1}^n$ and total Sobol' indices $\{S_k^{\text{tot}}\}_{k=1}^n$ using Eqs. (12) and (13), respectively.

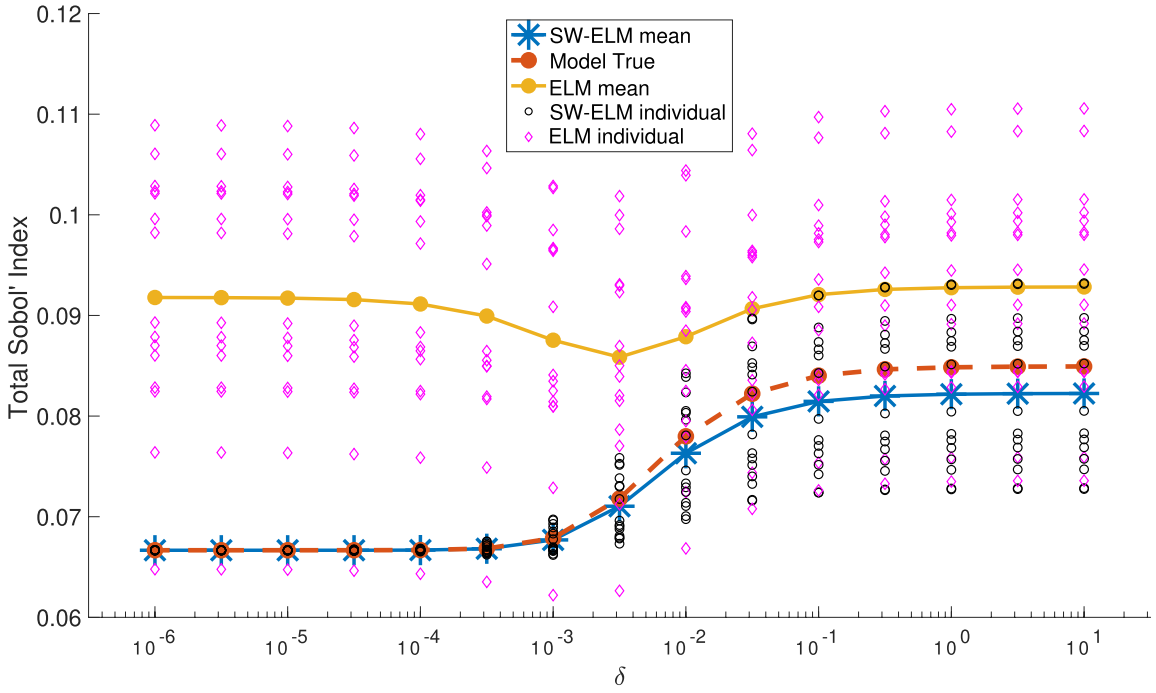

FIG. 3: Example (18) true analytic total Sobol' indices compared to those approximated by standard ELM and sparse weight ELM (SW-ELM). All approximated indices are plotted with means for each δ. In the true model with fixed δ, indices $S_i^{\text{tot}}(f)$ are equal for all $i = 1, \ldots, 15$. The "ELM individual" values show all ELM surrogate estimates of total indices when standard normal sampling is used; "ELM mean" gives the mean of these estimates. "SW-ELM individual" values show ELM surrogate estimates of total indices when the sparse weight sampling of Algorithm 1 is used; "SW-ELM mean" is the mean of these estimates.

5. COMPUTATIONAL RESULTS

5.1 Analytic Example: Sobol' g-Function

The Sobol' g-function [4]

$$f(\boldsymbol{x}) = \prod_{i=1}^{d} g_i(x_i) \quad \boldsymbol{x} \in [0,1]^d, \quad g_i(x_i) = \frac{|4x_i - 2| + a_i}{1 + a_i}, \quad i = 1, \ldots, d, \tag{22}$$

is commonly used as a benchmark to test new methods; its nonlinearity and lack of smoothness make approximating its Sobol' indices a significant challenge. Moreover, we can compare to the true Sobol' indices. Formulas for these can be found in the appendix to [7]. The constants a_i, chosen from the interval $(-1, \infty)$, can be tuned to determine which input variables are important. The closer a_i is to -1, the more important x_i becomes. On the other hand, the relative importance of x_i diminishes with larger values of a_i. For our test, we take the input dimension $d = 8$ and let $\boldsymbol{a} = [1, 2, 5, 10, 20, 50, 100, 500]$ as in [11].

We train our SW-ELM surrogates with 160 hidden layer neurons using 400 training points and 100 validation points, all sampled by LHS. The regularization parameter $\alpha = 10^{-3}$ is selected by the L-curve method. The sparsification parameter p is determined through Algorithm 1. While Fig. 4 does not show drastic improvements in error, we nevertheless conclude from this test that sparsifying may improve our approximations when performing GSA; we select the sparsification parameter $p = 0.85$. Sobol' indices estimated by SW-ELM are compared to their true analytic values.

As can seen from Fig. 5, SW-ELM correctly ranks the Sobol' indices and successfully identifies the most influential input variables. For important input variables, each approximated first-order Sobol' index is within 5% relative

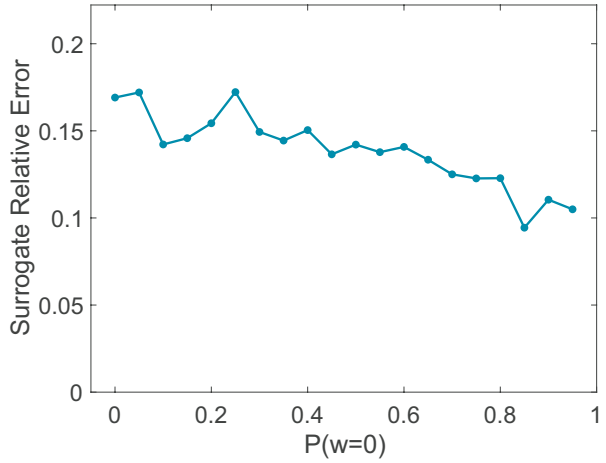

FIG. 4: Sparsification test for eight-dimensional g-function (22) with $\boldsymbol{a} = [1, 2, 5, 10, 20, 50, 100, 500]$. Relative surrogate error is estimated using validation set with 100 points.

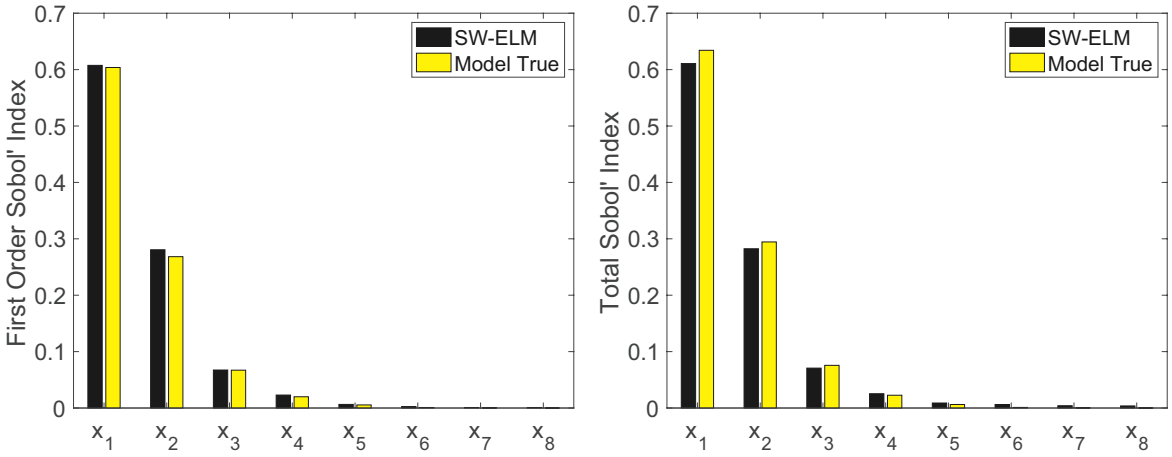

FIG. 5: Eight-dimensional g-function (22) with $\boldsymbol{a} = [1, 2, 5, 10, 20, 50, 100, 500]$; first-order (left) and total (right) Sobol' indices approximated via SW-ELM and computed analytically

error of the respective true first-order Sobol' index while each approximated total Sobol' index is within 7% relative error of the respective true total Sobol' index.

We use this benchmark example to study how numerical accuracy of SW-ELM affects the accuracy of GSA. In the experiment, we train SW-ELMs to a certain accuracy level by adding neurons and training points until the surrogate reaches the accuracy level. Each surrogate uses two training points for every neuron. We decrease the surrogate relative error from 15% to 5% and record the relative error of the first-order and total Sobol' index estimates for the inputs x_1, x_2, x_3. The relations between numerical error and GSA error are shown in Fig. 6. The experiment shows that SW-ELM can provide accurate GSA estimates, especially for the more influential inputs, despite 11% numerical error. Still, reducing the numerical error has a demonstrative effect on GSA. As we decrease the error to 5%, the Sobol' indices become much more accurate.

5.2 Genetic Oscillator

We examine the performance of the proposed approach on a challenging application problem from biochemistry, specifically, the genetic oscillator system that describes the time evolution of molecular species involved in the

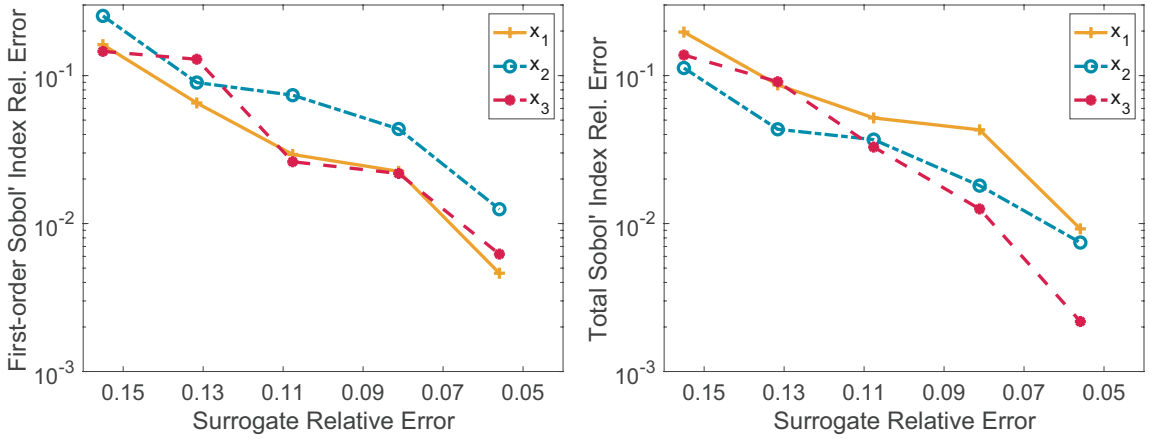

FIG. 6: Eight-dimensional g-function (22) with $\boldsymbol{a} = [1, 2, 5, 10, 20, 50, 100, 500]$; effect of reducing numerical error on the relative error of first-order (left) and total (right) Sobol' indices estimated by SW-ELM

regulation of circadian rhythm [65]. The reaction network consists of 16 reactions and involves nine species. The reactions, the corresponding propensity functions, and the nominal values of the rate parameters are given in Table 1. We consider the reaction rate equations (RREs), described by a nonlinear system of ordinary differential equations (ODEs), for this reaction network; see e.g., [65], [63], or [64], which we follow in the specific problem formulation used in the present study. We focus on the uncertainty in the reaction rate parameters. A uniform distribution is attached to each rate parameter on an interval given by a $\pm$ 5% perturbation from the corresponding nominal value. We use a random vector $\boldsymbol{x} \in \mathbb{R}^{16}$, whose entries are uniformly distributed on $[0, 1]$, to parametrize the uncertainty in the reaction rates. The vector of reaction rates is obtained by applying a linear transformation to $\boldsymbol{x}$ that maps the entries of $\boldsymbol{x}$ to the respective physical ranges. In the present study, we consider the quantity of interest (QoI) given by

$$f(\boldsymbol{x}) = \frac{1}{T}\int_0^T R(t;\boldsymbol{x})\,dt, \tag{23}$$

TABLE 1: Genetic oscillator reactions, propensity functions, parameters, and nominal values of the parameters [63,64]

Reaction	Propensity function	Parameter	Value
$P_a \to P_a + \text{mRNA}_a$	$\alpha_A P_a$	α_A	50.0
$P_r \to P_r + \text{mRNA}_r$	$\alpha_R P_r$	α_R	0.01
$\text{mRNA}_a \to \text{mRNA}_a + A$	$\beta_A \text{mRNA}_a$	β_A	50.0
$\text{mRNA}_r \to \text{mRNA}_r + R$	$\beta_R \text{mRNA}_r$	β_R	5.0
$A + R \to C$	$\gamma_C AR$	γ_C	20.0
$P_a + A \to P_a\_A$	$\gamma_A P_a A$	γ_A	1.0
$P_a\_A \to P_a + A$	$\theta_A P_a\_A$	θ_A	50.0
$P_r + A \to P_r\_A$	$\gamma_R P_r A$	γ_R	1.0
$P_r\_A \to P_r + A$	$\theta_R P_r\_A$	θ_R	1.0
$A \to \emptyset$	$\delta_A A$	δ_A	1.0
$R \to \emptyset$	$\delta_R R$	δ_R	0.2
$\text{mRNA}_a \to \emptyset$	$\delta_{MA} \text{mRNA}_a$	δ_{MA}	10.0
$\text{mRNA}_r \to \emptyset$	$\delta_{MR} \text{mRNA}_r$	δ_{MR}	0.5
$C \to R$	$\delta'_A C$	δ'_A	1.0
$P_a\_A \to P_a\_A + \text{mRNA}_a$	$\alpha_a \alpha_A P_a\_A$	α_a	10.0
$P_r\_A \to P_r\_A + \text{mRNA}_r$	$\alpha_r \alpha_R P_r\_A$	α_r	5000

where $R(t;\boldsymbol{x})$ is the concentration of the species R present in the system at time t, and T is the final simulation time. Notice that computing $R(t;\boldsymbol{x})$, for $t \in [0,T]$, requires solving the system of RREs with the reaction rates set according to $\boldsymbol{x}$. This system, for the present application, is given by a stiff system of nonlinear ODEs. Evaluations of the QoI in Eq. (23) are computationally expensive. This sensitivity analysis problem was considered in [64], where Sobol' indices were estimated by the pick-freeze method. We replicate their results using 1.5×10^6 samples and use these as reference values to measure the accuracy of our method.

Figure 7 (top row) presents approximations for first-order and total Sobol' indices when using standard ELM. The experiment uses 3000 training points and 1000 neurons. The regularization parameter $\alpha = 10^{-4}$ is selected

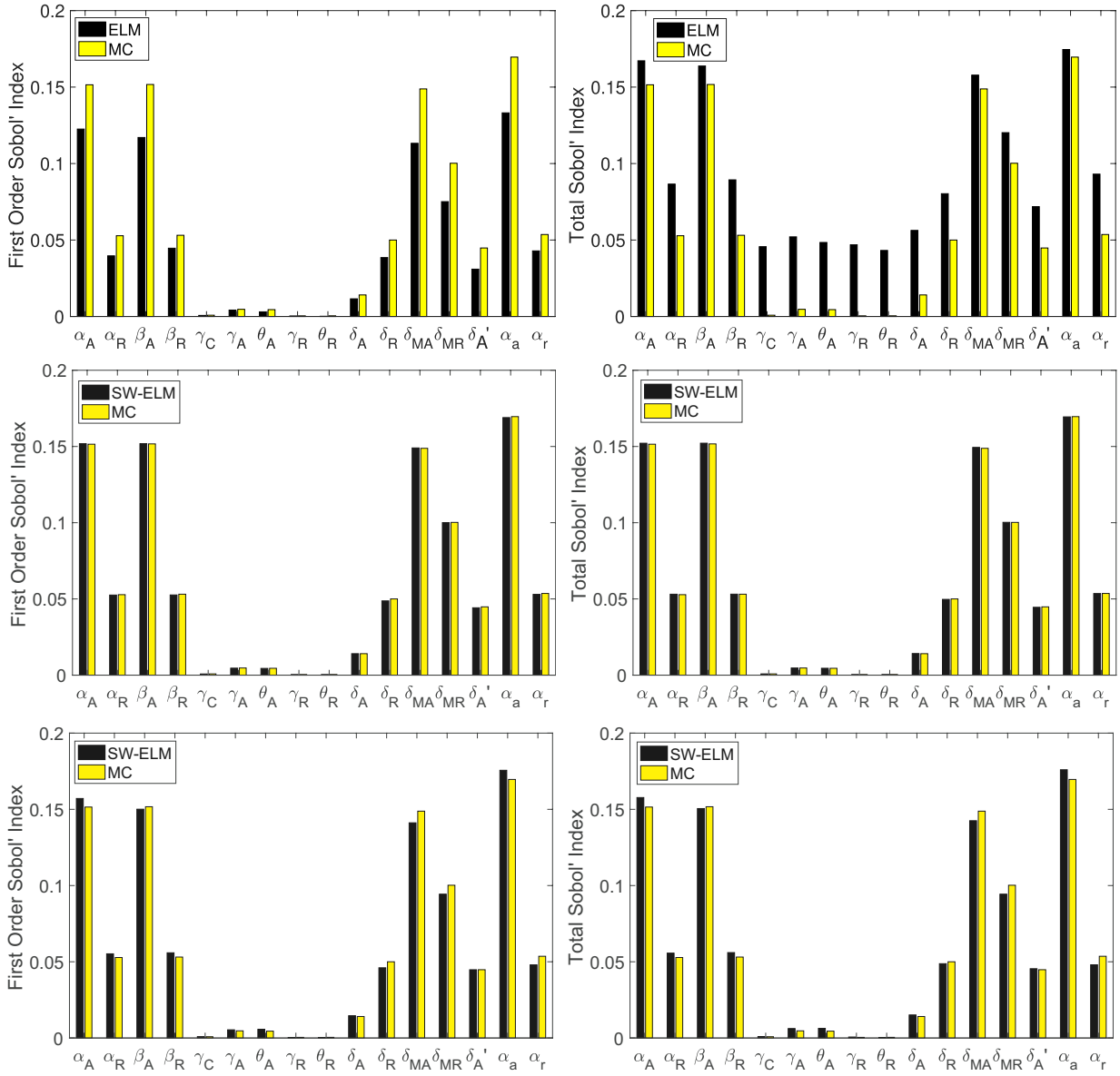

FIG. 7: Top row: first-order (left) and total Sobol's (right) indices corresponding to the QoI (23) approximated via ELM with weights sampled from standard normal distribution, using a training set with 3000 points. MC values are pick-freeze estimates. Middle row: same experiment with SW-ELM, with a 100-point validation set, instead of standard ELM. Bottom row: same experiment with SW-ELM instead of ELM and training size of 150 instead of 3000.

by the L-curve method. We observe that approximation using standard ELM leads to overestimates, particularly for unimportant input variables, of the total Sobol' indices and underestimates of the first-order Sobol' indices. This is a clear case of standard ELM overestimating contributions to the output variance by higher order interactions.

Let us instead tackle this problem using SW-ELM. We train our surrogates with again 3000 training points, using 1000 hidden layer neurons, and 100 validation points, all sampled by LHS. Regarding sparsification, we follow Algorithm 1. The results of these tests are displayed in Fig. 8 (left), and show that as the hidden layer weight matrix becomes more sparse, the error in the surrogate improves dramatically. We select the sparsification parameter $p = 0.8$. Figure 7 (middle row) displays the resulting approximated Sobol' indices and compares them to the reference values. Given that the Sobol' indices computed via pick-freeze methods are a close representation of the true Sobol' indices, SW-ELM provides an accurate approximation of the indices.

We can further challenge the performance of SW-ELM. We train surrogates with only 150 training points, using 50 hidden layer neurons and 100 validation points. The results of sparsification tests are displayed in Fig. 8 (right). We select the sparsification parameter $p = 0.9$. Figure 7 (bottom row) displays the resulting approximated Sobol' indices. SW-ELM with 150 training points or 3000 training points gives very similar results. This indicates that SW-ELM is extremely efficient for this problem—a modest number of training samples are sufficient for obtaining accurate sensitivity analysis results.

As shown in Fig. 7, the first-order Sobol' indices and the total Sobol' indices are very close to each other for this appplication. This suggests that there are few, if any, higher order variable interactions present in the QoI function (23). The fact that the sparsity of the weight matrix had a significant influence over the quality of the ELM surrogate, as seen in Fig. 8, further supports this. Here, sparsification is not simply helpful; it is essential and allows us to drastically decrease the number of sample points needed to obtain accurate approximations.

5.3 High-Dimensional Example

In this section, we use our proposed approach to peform GSA in a model governed by a system of ODEs which has 50 uncertain inputs. Specifically, we consider a linear homogeneous ODE system

$$\dot{\boldsymbol{x}} = -\mathbf{A}\boldsymbol{x}, \quad \boldsymbol{x}_0 = \boldsymbol{x}(0), \tag{24}$$

where $\boldsymbol{x}(t) = [x_1(t), \ldots, x_{50}(t)]^\top$ and $\mathbf{A} \in \mathbb{R}^{50\times 50}$. Note that the solution of this is system is given by $\boldsymbol{x}(t) = e^{-t\mathbf{A}}\boldsymbol{x}_0$.

In the present example, we assume $\mathbf{A}$ is symmetric. Thus, $\mathbf{A}$ has spectral decomposition $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$, where $\mathbf{Q}$ is orthogonal and Λ is diagonal with the eigenvalues $\lambda_1, \ldots, \lambda_{50}$ of $\mathbf{A}$ as its diagonal entries. This allows us to express the solution as follows:

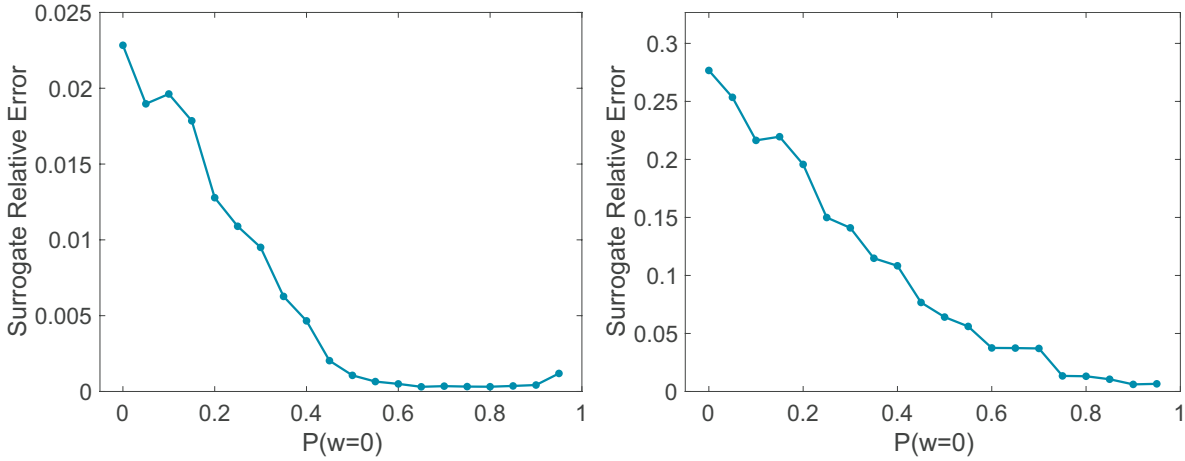

FIG. 8: Sparsification test for 16-dimensional QoI (23). Left: ELMs trained with training size 3000. Right: ELMs trained with training size 150. Relative surrogate error is estimating using a validation set with 100 points.

$$\boldsymbol{x}(t) = \mathbf{Q}e^{-t\boldsymbol{\Lambda}}\mathbf{Q}^\top \boldsymbol{x}_0, \tag{25}$$

where $e^{-t\boldsymbol{\Lambda}}$ is the diagonal matrix taking $e^{-t\lambda_1}, \ldots, e^{-t\lambda_{50}}$ as its diagonal entries. In this example, we consider a symmetric matrix $\mathbf{A}$ with a known orthogonal matrix $\mathbf{Q}$ of eigenvectors and uncertain eigenvalues. That is, the diagonal entries of $\boldsymbol{\Lambda}$ in Eq. (25), are uncertain. We assume these eigenvalues take nominal values $\lambda_k = 1/k$, $k = 1, \ldots, 50$. Each eigenvalue is assumed to follow a uniform distribution on an interval given by a $\pm$ 5% perturbation from the respective nominal value. We consider the quantity of interest,

$$f(\boldsymbol{\theta}) = x_{50}(10), \tag{26}$$

which is the last entry in the solution vector $\boldsymbol{x}(t) = \mathbf{Q}e^{-t\boldsymbol{\Lambda}}\mathbf{Q}^\top \boldsymbol{x}_0$ at time $t = 10$. For further details on the setup of the problem, see Appendix C. Reference Sobol' indices are computed by the pick-freeze method using 10^5 samples.

As show in Fig. 9, the surrogate error reaches a minimum when $p = 0.95$, indicating that a very sparse weight matrix should be used. In Fig. 10, we see the results of the sparsification test. The experiment uses 700 training points, 350 neurons, and 100 validation points to compute surrogate error. Figure 10 demonstrates the accuracy of the total

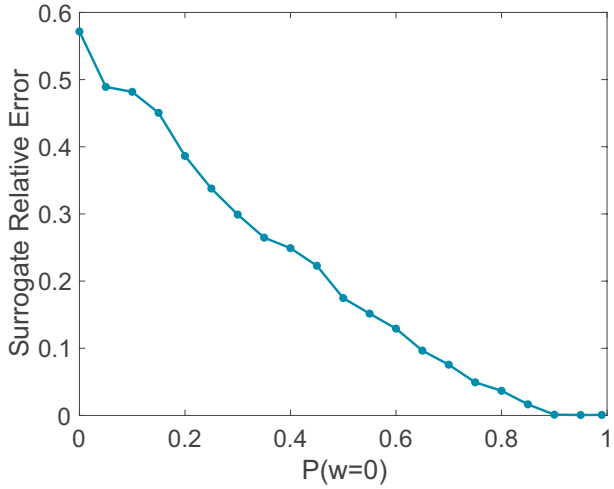

FIG. 9: Sparsification test for Eq. (26) on a 100-point validation set. ELMs trained with 700 training points and 350 neurons. Surrogate relative error reaches a minimum of 2.73×10^{-4} on the validation set at $p = 0.95$.

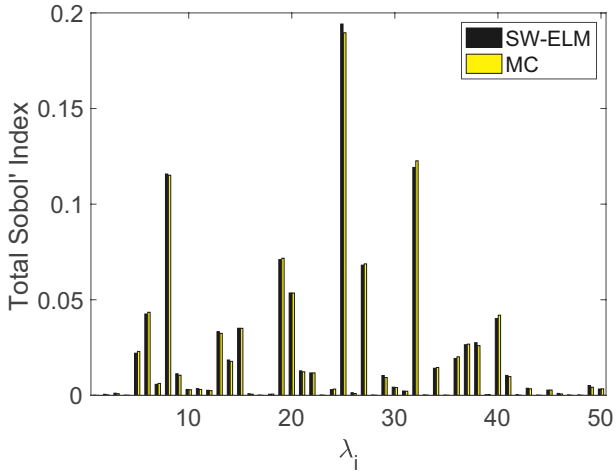

FIG. 10: Total Sobol' indices are computed for SW-ELM; MC values are pick-freeze estimates; and SW-ELM with $p = 0.95$ are trained on 700 training points with 350 neurons

Sobol' indices estimated using SW-ELM. These are compared to the reference values. The Sobol' indices reveal the absence of variable interactions, justifying the use of a very sparse weight matrix.

6. CONCLUSION

The use of SW-ELM as a surrogate in the context of variance-based GSA allows the user to completely eschew Monte Carlo integration, which is a perennial bottleneck in this field. The previously presented numerical results make a strong case for further study of the approach. In particular, there is, in general, very little known theoretically regarding the link from Eq. (14) (accurate surrogate) to Eq. (15) (accurate GSA).

In our particular case, we note further that SW-ELM is not strictly covered by the existing theory [34] because the distribution we use for $\mathbf{W}$ does not correspond to a continuous random variable. The ELM theory of [34] is also limited in not providing convergence rates or generalization bounds. In this respect, theory for RVFLs [39,66] and random features [45,57] is more developed. Extending existing theory to the setting in this paper is an interesting direction for future study.

A key point in our approach is the ability to analytically compute the Sobol' indices of $\hat{f}$. This is where assumptions about the parameter distribution are crucial. Further work is need to understand how these assumptions can be generalized. More activation functions could be identified that facilitate formulas for Sobol' indices with respect to specific distributions of interest. Application of isoprobablistic transformations, as is commonly done in the context of PCE [9,67], can expand the classes of input distributions that can make use of the analytic Sobol' index formulas of SW-ELM.

For certain applications, our method can be enhanced by building on sparse feature methods [28,68], including Bayesian compressive sensing [69] and data-driven [31,70] approaches geared to high-dimensional models. Application to ELM is an area for future work. A deeper understanding of the proposed sparsification process could lead to improvement of the proposed approach but also have broader impact. We interpret the weight sparsification step as a data-driven means of increasing the probability of choosing basis functions with appropriate ANOVA components. The algorithm, which bears similarity to ours, proposed in [57] for random feature expansions also enjoys enhanced accuracy from weight sparsification. Future directions should seek to develop sparse sampling approaches that create a random basis from more sophisticated information about the ANOVA decomposition of the target model.

ACKNOWLEDGMENTS

The work of J.E. Darges, A. Alexanderian, and P.A. Gremaud was supported in part by US National Science Foundation under Grant Nos. DMS-1745654 and DMS-1953271. Additionally, the work of A. Alexanderian was also supported in part by the US National Science Foundation under Grant No. DMS-2111044.

REFERENCES

1. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., *Global Sensitivity Analysis: The Primer*, Hoboken, NJ: Wiley, 2008.
2. Iooss, B. and Le Maître, P., *A Review on Global Sensitivity Analysis Methods*, Boston: Springer, pp. 101–122, 2015.
3. Iooss, B. and Saltelli, A., Introduction to Sensitivity Analysis, in *Handbook of Uncertainty Quantification*, R. Ghanem, D. Higdon, and H. Owhadi, Eds., New York: Springer, pp. 1103–1122, 2017.
4. Saltelli, A. and Sobol', I., Sensitivity Analysis for Nonlinear Mathematical Models: Numerical Experience, *Mat. Model.*, **7**(11):16–28, 1995.
5. Sobol', I., Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates, *Math. Comput. Simul.*, **55**(1-3):271–280, 2001.
6. Prieur, C. and Tarantola, S., Variance-Based Sensitivity Analysis: Theory and Estimation Algorithms, in *Handbook of Uncertainty Quantification*, R. Ghanem, D. Higdon, and H. Owhadi, Eds., New York: Springer, pp. 1217–1239, 2017.
7. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S., Variance Based Sensitivity Analysis of Model Output: Design and Estimator for the Total Sensitivity Index, *Comput. Phys. Commun.*, **181**(2):259–270, 2010.

8. Hart, J., Gremaud, P., and David, T., Global Sensitivity Analysis of High-Dimensional Neuroscience Models: An Example of Neurovascular Coupling, *Bull. Math. Biol.*, **81**(6):1805–1828, 2019.
9. Le Gratiet, L., Marelli, S., and Sudret, B., Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes, in *Handbook of Uncertainty Quantification*, R. Ghanem, D. Higdon, and H. Owhadi, Eds., New York: Springer, 2017.
10. Sargsyan, K., Surrogate Models for Uncertainty Propagation and Sensitivity Analysis, in *Handbook of Uncertainty Quantification*, R. Ghanem, D. Higdon, and H. Owhadi, Eds., New York: Springer, 2017.
11. Sudret, B., Global Sensitivity Analysis Using Polynomial Chaos Expansions, *Reliab. Eng. Syst. Safety*, **93**(7):964–979, 2008.
12. Crestaux, T., Le Maître, O., and Martinez, J.M., Polynomial Chaos Expansion for Sensitivity Analysis, *Reliab. Eng. Syst. Saf.*, **94**(7):1161–1172, 2009.
13. Friedman, J.H., Multivariate Adaptive Regression Splines, *Ann. Stat.*, **19**(1):1–67, 1991.
14. Hart, J., Alexanderian, A., and Gremaud, P., Efficient Computation of Sobol' Indices for Stochastic Models, *SIAM J. Sci. Comput.*, **39**(4):A1514–A1530, 2017.
15. Marrel, A., Iooss, B., Laurent, B., and Roustant, O., Calculations of Sobol' Indices for the Gaussian Process Metamodel, *Reliab. Eng. Syst. Safety*, **94**(3):742–751, 2009.
16. Oakley, J.E. and O'Hagan, A., Probabilistic Sensitivity Analysis of Complex Models: A Bayesian Approach, *J. Royal Stat. Society: Ser. B*, **66**(3):751–769, 2004.
17. Jin, R., Chen, W., and Sudjianto, A., Analytical Metamodel-Based Global Sensitivity Analysis and Uncertainty Propagation for Robust Design, SAE Tech. Paper 2004-01-0429, 2004.
18. Horiguchi, A., Pratola, M.T., and Santner, T.J., Assessing Variable Activity for Bayesian Regression Trees, *Reliab. Eng. Syst. Safety*, **207**:107391, 2021.
19. Antoniadis, A., Lambert-Lacroix, S., and Poggi, J.M., Random Forests for Global Sensitivity Analysis: A Selective Review, *Reliab. Eng. Syst. Safety*, **206**:107312, 2021.
20. Steiner, M., Bourinet, J.M., and Lahmer, T., An Adaptive Sampling Method for Global Sensitivity Analysis Based on Least-Squares Support Vector Regression, *Reliab. Eng. Syst. Safety*, **183**:323–340, 2019.
21. Fock, E., Global Sensitivity Analysis Approach for Input Selection and System Identification Purposes: A New Framework for Feedforward Neural Networks, *IEEE Trans. Neural Netw. Learn. Syst.*, **25**(8):1484–1495, 2014.
22. Datteo, A., Busca, G., Quattromani, G., and Cigada, A., On the Use of AR Models for SHM: A Global Sensitivity and Uncertainty Analysis Framework, *Reliab. Eng. Syst. Safety*, **170**:99–115, 2018.
23. Todri, E., Amenaghawon, A., Del Val, I., Leak, D., Kontoravdi, C., Kucherenko, S., and Shah, N., Global Sensitivity Analysis and Meta-Modeling of an Ethanol Production Process, *Chem. Eng. Sci.*, **114**:114–127, 2014.
24. Cheng, K., Zhenzhou, L., Ling, C., and Zhou, S., Surrogate-Assisted Global Sensitivity Analysis: An Overview, *Struct. Multidiscip. Optim.*, **61**:1187–1213, 2020.
25. Wu, Z., Wang, D., Okolo, P.N., Hu, F., and Zhang, W., Global Sensitivity Analysis Using a Gaussian Radial Basis Function Metamodel, *Reliab. Eng. Syst. Safety*, **154**:171–179, 2016.
26. Wu, Z., Wang, W., Wang, D., Zhao, K., and Zhang, W., Global Sensitivity Analysis Using Orthogonal Augmented Radial Basis Function, *Reliab. Eng. Syst. Safety*, **185**:291–302, 2019.
27. Blatman, G. and Sudret, B., Efficient Computation of Global Sensitivity Indices Using Sparse Polynomial Chaos Expansions, *Reliab. Eng. Syst. Safety*, **95**(11):1216–1229, 2010.
28. Lüthen, N., Marelli, S., and Sudret, B., Sparse Polynomial Chaos Expansions: Literature Survey and Benchmark, *SIAM/ASA J. Uncertain. Quantif.*, **9**(2):593–649, 2021.
29. Alexanderian, A., Gremaud, P.A., and Smith, R.C., Variance-Based Sensitivity Analysis for Time-Dependent Processes, *Reliab. Eng. Syst. Safety*, **196**:106722, 2020.
30. Ehre, M., Papaioannou, I., and Straub, D., Global Sensitivity Analysis in High Dimensions with PLS-PCE, *Reliab. Eng. Syst. Safety*, **198**:106861, 2020.
31. Zhou, Y., Lu, Z., Hu, J., and Hu, Y., Surrogate Modeling of High-Dimensional Problems via Data-Driven Polynomial Chaos Expansions and Sparse Partial Least Square, *Comput. Methods Appl. Mech. Eng.*, **364**:112906, 2020.
32. Lüthen, N., Marelli, S., and Sudret, B., Automatic Selection of Basis-Adaptive Sparse Polynomial Chaos Expansions for Engineering Applications, *Int. J. Uncertain. Quantif.*, **12**(3):49–74, 2022.

33. Almohammadi, S.M., Le Maître, O.P., and Knio, O.M., Computational Challenges in Sampling and Representation of Uncertain Reaction Kinetics in Large Dimensions, *Int. J. Uncertain. Quantif.*, **12**(1):1–24, 2022.
34. Huang, G.B., Zhu, Q.Y., and Siew, C.K., Extreme Learning Machine: Theory and Applications, *Neurocomputing*, **70**(1):489–501, 2006.
35. Huang, G.B., Wang, D., and Lan, Y., Extreme Learning Machines: A Survey, *Int. J. Mach. Learn. Cybern.*, **2**(2):107–122, 2011.
36. Zhang, Q., Zhao, Y.G., Kolozvari, K., and Xu, L., Reliability Analysis of Reinforced Concrete Structure against Progressive Collapse, *Reliab. Eng. Syst. Safety*, **228**:108831, 2022.
37. Schmidt, W., Kraaijveld, M., and Duin, R., Feedforward Neural Networks with Random Weights, in *Proc. of 11th IAPR International Conf. on Pattern Recognition, Vol. II, Conf. B: Pattern Recognition Methodology and Systems*, Los Alamitos, CA: IEEE, pp. 1–4, 1992.
38. Pao, Y.H., Park, G.H., and Sobajic, D.J., Learning and Generalization Characteristics of the Random Vector Functional-Link Net, *Neurocomputing*, **6**(2):163–180, 1994.
39. Igelnik, B. and Pao, Y.H., Stochastic Choice of Basis Functions in Adaptive Function Approximation and the Functional-Link Net, *IEEE Trans. Neural Netw.*, **6**(6):1320–1329, 1995.
40. Scardapane, S. and Wang, D., Randomness in Neural Networks: An Overview, *Wiley Interdiscip. Rev.: Data Mining Knowl. Discov.*, **7**(2):e1200, 2017.
41. Cao, W., Wang, X., Ming, Z., and Gao, J., A Review on Neural Networks with Random Weights, *Neurocomputing*, **275**:278–287, 2018.
42. Suganthan, P.N. and Katuwal, R., On the Origins of Randomization-Based Feedforward Neural Networks, *Appl. Soft Comput.*, **105**:107239, 2021.
43. Rahimi, A. and Recht, B., Random Features for Large-Scale Kernel Machines, in *Advances in Neural Information Processing Systems*, Vol. 20, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., Red Hook, NY: Curran Associates, Inc., 2007.
44. Rahimi, A. and Recht, B., Weighted Sums of Random Kitchen Sinks: Replacing Minimization with Randomization in Learning, in *Advances in Neural Information Processing Systems*, Vol. 21, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., Red Hook, NY: Curran Associates, Inc., 2008.
45. Rahimi, A. and Recht, B., Uniform Approximation of Functions with Random Bases, in *Proc. of 46th Annual Allerton Conference on Communication, Control, and Computing*, Los Alamitos, CA: IEEE, pp. 555–561, 2008.
46. Nelsen, N.H. and Stuart, A.M., The Random Feature Model for Input-Output Maps between Banach Spaces, *SIAM J. Sci. Comput.*, **43**(5):A3212–A3243, 2021.
47. Liu, F., Huang, X., Chen, Y., and Suykens, J.K., Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44**(10):7128–7148, 2022.
48. Nagawkar, J. and Leifsson, L., Efficient Global Sensitivity Analysis of Model-Based Ultrasonic Nondestructive Testing Systems Using Machine Learning and Sobol' Indices, *J. Nondestruct. Eval.*, **4**(4):041008 , 2021.
49. Walzberg, J., Carpenter, A., and Heath, G.A., Role of the Social Factors in Success of Solar Photovoltaic Reuse and Recycle Programmes, *Nat. Energy*, **6**(9):913–924, 2021.
50. Li, S., Yang, B., and Qi, F., Accelerate Global Sensitivity Analysis Using Artificial Neural Network Algorithm: Case Studies for Combustion Kinetic Model, *Combust. Flame*, **168**:53–64, 2016.
51. Kapusuzoglu, B. and Mahadevan, S., Information Fusion and Machine Learning for Sensitivity Analysis Using Physics Knowledge and Experimental Data, *Reliab. Eng. Syst. Safety*, **214**:107712, 2021.
52. Ye, D., Nikishova, A., Veen, L., Zun, P., and Hoekstra, A.G., Non-Intrusive and Semi-Intrusive Uncertainty Quantification of a Multiscale In-Stent Restenosis Model, *Reliab. Eng. Syst. Safety*, **214**:107734, 2021.
53. Zhao, Y., Cheng, X., Zhang, T., Wang, L., Shao, W., and Wiart, J., A Global-Local Attention Network for Uncertainty Analysis of Ground Penetrating Radar Modeling, *Reliab. Eng. Syst. Safety*, **234**:109176, 2023.
54. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R., Regularization of Neural Networks Using DropConnect, in *Proc. of 30th International Conf. on Machine Learning*, Vol. 28, Stroudsburg, PA: International Machine Learning Society, pp. 1058–1066, 2013.
55. Katuwal, R. and Suganthan, P.N., Dropout and DropConnect Based Ensemble of Random Vector Functional Link Neural Network, in *Proc. of 2018 IEEE Symp. Series on Computational Intelligence (SSCI)*, Piscataway, NJ: IEEE, pp. 1772–1778, 2018.

56. Iosifidis, A., Tefas, A., and Pitas, I., DropELM: Fast Neural Network Regularization with Dropout and DropConnect, *Neurocomputing*, **162**:57–66, 2015.

57. Hashemi, A., Schaeffer, H., Shi, R., Topcu, U., Tran, G., and Ward, R., Generalization Bounds for Sparse Random Feature Expansions, *Appl. Comput. Harmon. Anal.*, **62**:310–330, 2023.

58. Pinkus, A., Approximation Theory of the MLP Model in Neural Networks, *Acta Numer.*, **8**:143–195, 1999.

59. Leshno, M., Lin, V.Y., Pinkus, A., and Schocken, S., Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function, *Neural Netw.*, **6**(6):861–867, 1993.

60. Hansen, P.C., *Getting Serious: Choosing the Regularization Parameter*, Chapter 5, Philadelphia: SIAM, pp. 85–107, 2010.

61. Evans, J.W., Gragg, W.B., and LeVeque, R.J., On Least Squares Exponential Sum Approximation with Positive Coefficients, *Math. Comput.*, **34**(149):203–211, 1980.

62. Braess, D., *Nonlinear Approximation Theory*, New York: Springer, 1986.

63. Sheppard, P.W., Rathinam, M., and Khammash, M., A Pathwise Derivative Approach to the Computation of Parameter Sensitivities in Discrete Stochastic Chemical Systems, *J. Chem. Phys.*, **136**(3):034115, 2012.

64. Merritt, M., Alexanderian, A., and Gremaud, P.A., Multiscale Global Sensitivity Analysis for Stochastic Chemical Systems, *Multiscale Model. Simul.*, **19**(1):440–459, 2021.

65. Vilar, J.M., Kueh, H.Y., Barkai, N., and Leibler, S., Mechanisms of Noise-Resistance in Genetic Oscillators, *Proc. Nat. Acad. Sci.*, **99**(9):5988–5992, 2002.

66. Needell, D., Nelson, A.A., Saab, R., and Salanevich, P., Random Vector Functional Link Networks for Function Approximation on Manifolds, *Stat. Mach. Learn.*, arXiv:2007.15776, 2022.

67. Blatman, G. and Sudret, B., An Adaptive Algorithm to Build Up Sparse Polynomial Chaos Expansions for Stochastic Finite Element Analysis, *Probab. Eng. Mech.*, **25**(2):183–197, 2010.

68. Adcock, B., Brugiapaglia, S., and Webster, C.G., *Sparse Polynomial Approximation of High-Dimensional Functions*, Philadelphia: SIAM, 2022.

69. Sargsyan, K., Safta, C., Najm, H.N., Debusschere, B.J., Ricciuto, D., and Thornton, P., Dimensionality Reduction for Complex Models via Bayesian Compressive Sensing, *Int. J. Uncertain. Quantif.*, **4**(1):63–93, 2014.

70. Lataniotis, C., Marelli, S., and Sudret, B., Extending Classical Surrogate Modelling to High Dimensions through Supervised Dimensionality Reduction: A Data-Driven Approach, *Int. J. Uncertain. Quantif.*, **10**(1):55–82, 2020.

71. Owen, A.B., *Monte Carlo Theory, Methods and Examples*, Thousand Oaks, CA: SAGE Publications, 2013.

APPENDIX A. DERIVATION OF ANALYTIC FORMULAS FOR ELM SURROGATE

APPENDIX A.1 Proof of Lemma 1

Proof. For the surrogate presented in this paper, we derive expressions for the mean and variance. Recall the expression for the ELM (10) and the definition of the function $\epsilon_k(t)$, given in the statement of the lemma,

$$\epsilon_k(t) = \int_0^1 e^{tx_k} \pi_k(x_k) dx_k.$$

We first find the expression for the mean,

$$\mathbb{E}(\hat{f}) = \int_{[0,1]^d} \sum_{j=1}^{n} \left(\beta_j e^{b_j} \prod_{l=1}^{d} e^{w_{j,l} x_l} \right) d\boldsymbol{x} = \sum_{j=1}^{n} \left(\beta_j e^{b_j} \prod_{l=1}^{d} \int_0^1 e^{w_{j,l} x_l} dx_l \right).$$

Therefore, the mean can be expressed as follows:

$$\mathbb{E}(\hat{f}) = \sum_{j=1}^{n} \left(\beta_j e^{b_j} \prod_{l=1}^{d} \epsilon_l(w_{j,l}) \right).$$

Next, we find the expression for the variance:

$$\begin{aligned}\text{var}(\hat{f}) = \mathbb{E}(\hat{f}^2) - \mathbb{E}(\hat{f})^2 &= \int_{[0,1]^d} \left(\sum_{j=1}^{n} \left(\beta_j e^{b_j} \prod_{l=1}^{d} e^{w_{j,l} x_l} \right) \right)^2 d\boldsymbol{x} - \mathbb{E}(\hat{f})^2 \\ &= \sum_{j,i=1}^{n} \left(\beta_j \beta_i e^{b_j + b_i} \prod_{l=1}^{d} \epsilon_l(w_{j,l} + w_{i,l}) \right) - \mathbb{E}(\hat{f})^2 \\ &= \sum_{j,i=1}^{n} \beta_j \beta_i e^{b_j + b_i} \left(\prod_{l=1}^{d} \epsilon_l(w_{j,l} + w_{i,l}) - \prod_{r=1}^{d} \epsilon_r(w_{j,r}) \epsilon_r(w_{i,r}) \right).\end{aligned}$$

□

APPENDIX A.2 Proof of Proposition 1

Proof. For the surrogate presented in this paper, we derive expressions for the general formulas for regular and total Sobol' indices and offer simplified expressions for indices corresponding to single variables. Recall the expression for the ELM [Eq. (10)] and the function $\epsilon_k(t)$, and let $\mathbb{E}(\hat{f})$ and $\text{var}(\hat{f})$ be as presented in Lemma 1.

Given a subset of variables of $\{x_1, \ldots, x_d\}$, we first derive the expressions for Sobol' indices,

$$S_{\boldsymbol{u}} = \frac{\text{var}(\hat{f}_{\boldsymbol{u}})}{\text{var}(\hat{f})}, \quad \hat{f}_{\boldsymbol{u}} := \sum_{\boldsymbol{v} \subseteq \boldsymbol{u}} (-1)^{|\boldsymbol{u}| - |\boldsymbol{v}|} \mathbb{E}\Big(\hat{f} | x_l,\ l \in \boldsymbol{v} \Big),$$

where $\boldsymbol{u} \subset \{1, \ldots, d\}$ [6]. Since terms in the ANOVA decomposition have the property that $\text{var}(\hat{f}_{\boldsymbol{u}}) = \mathbb{E}(\hat{f}^2_{\boldsymbol{u}})$ [71] we can we can express the Sobol' index $S_{\boldsymbol{u}}$ for Eq. (10) as follows:

$$\begin{aligned}S_{\boldsymbol{u}} &= \frac{\text{var}(\hat{f}_{\boldsymbol{u}})}{\text{var}(\hat{f})} = \frac{\mathbb{E}(\hat{f}^2_{\boldsymbol{u}})}{\text{var}(\hat{f})} \\ &= \frac{1}{\text{var}(\hat{f})} \int_{[0,1]^d} \left(\sum_{\boldsymbol{v} \subseteq \boldsymbol{u}} \sum_{j=1}^{n} (-1)^{|\boldsymbol{u}| - |\boldsymbol{v}|} \beta_j e^{b_j} \left(\prod_{l \in \boldsymbol{v}} e^{w_{j,l} x_l} \right) \left(\prod_{r \notin \boldsymbol{v}} \epsilon_r(w_{j,r}) \right) \right)^2 d\boldsymbol{x} \\ &= \frac{1}{\text{var}(\hat{f})} \sum_{\boldsymbol{v}, \tilde{\boldsymbol{v}} \subseteq \boldsymbol{u}} \sum_{j,i=1}^{n} (-1)^{2|\boldsymbol{u}| - |\boldsymbol{v}| - |\tilde{\boldsymbol{v}}|} \int_{[0,1]^d} \left(\beta_j \beta_i e^{b_j + b_i} \prod_{l \in \boldsymbol{v}} e^{w_{j,l} x_l} \prod_{r \notin \boldsymbol{v}} \epsilon_r(w_{j,r}) \prod_{s \in \tilde{\boldsymbol{v}}} e^{w_{i,s} x_l} \prod_{q \notin \tilde{\boldsymbol{v}}} \epsilon_q(w_{i,q}) \right) d\boldsymbol{x} \\ &= \frac{1}{\text{var}(\hat{f})} \sum_{\boldsymbol{v}, \tilde{\boldsymbol{v}} \subseteq \boldsymbol{u}} \sum_{j,i=1}^{n} (-1)^{2|\boldsymbol{u}| - |\boldsymbol{v}| - |\tilde{\boldsymbol{v}}|} \beta_j \beta_i e^{b_j + b_i} \left(\prod_{l \in \boldsymbol{v} \cap \tilde{\boldsymbol{v}}} \epsilon_l(w_{j,l} + w_{i,l}) \right) \left(\prod_{r \notin \boldsymbol{v} \cap \tilde{\boldsymbol{v}}} \epsilon_r(w_{j,r}) \epsilon_r(w_{i,r}) \right).\end{aligned}$$

When we only consider the single variable case or, in other words, when $\boldsymbol{u} = \{k\}$, then we arrive at Eq. (12) for the first-order Sobol' indices:

$$\begin{aligned}S_k &= \frac{1}{\text{var}(\hat{f})} \sum_{j,i=1}^{n} \left(\beta_j \beta_i e^{b_j + b_i} \epsilon_k(w_{j,k} + w_{i,k}) \prod_{l \neq k} \epsilon_l(w_{j,l}) \epsilon_l(w_{i,l}) \right) - \frac{\mathbb{E}(\hat{f})^2}{\text{var}(\hat{f})} \\ &= \frac{1}{\text{var}(\hat{f})} \sum_{j,i=1}^{n} \beta_j \beta_i e^{b_j + b_i} (\epsilon_k(w_{j,k} + w_{i,k}) - \epsilon_k(w_{j,k}) \epsilon_k(w_{i,k})) \prod_{l \neq k} \epsilon_l(w_{j,l}) \epsilon_l(w_{i,l}).\end{aligned}$$

Now we derive the expressions for the total Sobol' indices for a given subset of variables of $\{x_1, \ldots, x_d\}$,

$$S^{\text{tot}}_{\boldsymbol{u}} = 1 - \frac{\text{var}(\mathbb{E}(\hat{f} | x_r,\ r \notin \boldsymbol{u}))}{\text{var}(\hat{f})},$$

where $\boldsymbol{u} \subset \{1, \ldots, d\}$ [6]. We find the expression for the total index $S_{\boldsymbol{u}}^{\text{tot}}$ for Eq. (10), as follows:

$$\begin{aligned} S_{\boldsymbol{u}}^{\text{tot}} &= 1 - \frac{1}{\text{var}(\hat{f})}\left(\int_{[0,1]^d}\left(\sum_{j=1}^{n}\beta_j e^{b_j}\left(\prod_{l\in\boldsymbol{u}}\epsilon_l(w_{j,l})\right)\left(\prod_{r\notin\boldsymbol{u}}e^{w_{j,r}x_r}\right)\right)^2 d\boldsymbol{x} - \mathbb{E}(\hat{f})^2\right) \\ &= 1 - \frac{1}{\text{var}(\hat{f})}\left(\sum_{j,i=1}^{n}\beta_j\beta_i e^{b_j+b_i}\left(\prod_{l\in\boldsymbol{u}}\epsilon_l(w_{j,l})\epsilon_l(w_{i,l})\right)\left(\prod_{r\notin\boldsymbol{u}}\epsilon_r(w_{j,r}+w_{i,r})\right) - \mathbb{E}(\hat{f})^2\right). \end{aligned}$$

When we consider the total Sobol' index corresponding to a single variable x_k, we have expression Eq. (13) based on:

$$\begin{aligned} S_k^{\text{tot}} &= 1 - \frac{1}{\text{var}(\hat{f})}\left(\sum_{j,i=1}^{n}\beta_j\beta_i e^{b_j+b_i}\epsilon_k(w_{j,k})\epsilon_k(w_{i,k})\prod_{l\neq k}\epsilon_l(w_{j,l}+w_{i,l}) - \mathbb{E}(\hat{f})^2\right) \\ &= 1 - \frac{1}{\text{var}(\hat{f})}\sum_{j,i=1}^{n}\beta_j\beta_i e^{b_j+b_i}\epsilon_k(w_{j,k})\epsilon_k(w_{i,k})\left(\prod_{l\neq k}\epsilon_l(w_{j,l}+w_{i,l}) - \prod_{r\neq k}\epsilon_r(w_{j,r})\epsilon_r(w_{i,r})\right). \end{aligned}$$

□

APPENDIX B. SOBOL' INDICES OF TUNABLE INTERACTIONS EXAMPLE

Here we present formulas for the analytic first-order and total Sobol' indices for Eq. (18), given an input dimension d and parameter δ. The first-order Sobol indices are the same for all $k = 1, \ldots, d$. The total Sobol' indices are also the same for $k = 1, \ldots, d$. The first-order and total Sobol' indices are given by

$$S_k = \frac{\text{var}(\mathbb{E}(f_\delta|x_k))}{\text{var}(f_\delta)} \quad \text{and} \quad S_k^{\text{tot}} = 1 - \frac{\text{var}(\mathbb{E}(f_\delta|x_l,\ l\neq k))}{\text{var}(f_\delta)},$$

where

$$\begin{aligned} \text{var}(f_\delta) &= \frac{d\delta}{9}\left(\frac{3}{2}\right)^d + \delta^2\left(\left(\frac{7}{3}\right)^d - \left(\frac{9}{4}\right)^d\right) + \frac{d}{12}, \\ \text{var}(\mathbb{E}(f_\delta|x_k)) &= \frac{\delta^2}{27}\left(\frac{9}{4}\right)^d + \frac{\delta}{9}\left(\frac{3}{2}\right)^d + \frac{1}{12}, \\ \text{var}(\mathbb{E}(f_\delta|x_l,\ l\neq k)) &= \frac{\delta(d-1)}{9}\left(\frac{3}{2}\right)^d + \delta^2\left(\frac{27}{28}\left(\frac{7}{3}\right)^d - \left(\frac{9}{4}\right)^d\right) + \frac{d-1}{12}. \end{aligned}$$

APPENDIX C. SETUP OF HIGH-DIMENSIONAL EXAMPLE

Here we describe our choices for $\boldsymbol{x}_0$ and $\mathbf{Q}$ in Eq. (25), as implemented in MATLAB. For the initial state, $\boldsymbol{x}_0$, we take $\boldsymbol{x}_0 = [1, \ldots, 1]^\top$. We chose $\mathbf{Q}$ by generating a random orthogonal matrix as follows. First, we generated a matrix $\mathbf{P}$ whose entries are independent draws from standard normal distribution. This was done using MATLAB's `randn` command. The random seed was fixed using `rng(1)`. Subsequently, the matrix $\mathbf{Q}$ was obtained by performing a QR factorization $\mathbf{P} = \mathbf{QR}$, where $\mathbf{Q}$ is orthogonal and $\mathbf{R}$ is upper triangular, and retaining $\mathbf{Q}$. The QR factorization was computed using MATLAB's `qr` command.