High-Dimensional Uncertainty Quantification via Tensor Regression with Rank Determination and Adaptive Sampling

Zichang He and Zheng Zhang, *Member, IEEE*

(Invited Paper)

***Abstract*—Fabrication process variations can significantly influence the performance and yield of nano-scale electronic and photonic circuits. Stochastic spectral methods have achieved great success in quantifying the impact of process variations, but they suffer from the curse of dimensionality. Recently, low-rank tensor methods have been developed to mitigate this issue, but two fundamental challenges remain open: how to automatically determine the tensor rank and how to adaptively pick the informative simulation samples. This paper proposes a novel tensor regression method to address these two challenges. We use a ℓ_q/ℓ_2 group-sparsity regularization to determine the tensor rank. The resulting optimization problem can be efficiently solved via an alternating minimization solver. We also propose a two-stage adaptive sampling method to reduce the simulation cost. Our method considers both exploration and exploitation via the estimated Voronoi cell volume and nonlinearity measurement respectively. The proposed model is verified with synthetic and some realistic circuit benchmarks, on which our method can well capture the uncertainty caused by 19 to 100 random variables with only 100 to 600 simulation samples.**

***Index Terms*—Tensor regression, high dimensionality, uncertainty quantification, polynomial chaos, process variation, rank determination, adaptive sampling.**

I. Introduction

Fabrication process variations (e.g., surface roughness of interconnects and photonic waveguide, and random doping effects of transistors) have been a major concern in nano-scale chip design. They can significantly influence chip performance and decrease product yield [2]. Monte Carlo (MC) is one of the most popular methods o quantify the chip performance under uncertainty, but it requires a huge amount of computational cost [3]. Instead, stochastic spectral methods based on generalized polynomial chaos (gPC) [4] offer efficient solutions for fast uncertainty quantification by approximating a real uncertain circuit variable as a linear combination of some stochastic basis functions [5–7]. These techniques have been increasingly used in design automation [8–15]. The main challenge of the stochastic spectral method is the curse of dimensionality: the computational cost grows very fast as the number of random parameters increases. In order to address this fundamental challenge, many high-dimensional solvers have been developed. The representative techniques include (but are not limited to) compressive sensing [16, 17], hyperbolic regression [18], analysis of variance (ANOVA) [19, 20], model order reduction [21], and hierarchical modeling [22, 23], and tensor methods [23, 24].

The low-rank tensor approximation has shown promising performance in solving high-dimensional uncertainty quantification problems [24–29]. By low-rank tensor decomposition, one may reduce the number of unknown variables in uncertainty quantification to a linear function of the parameter dimensionality. However, there is a fundamental question: how can we determine the tensor rank and the associated model complexity? Because it is hard to exactly determine a tensor rank *a-priori* [30], existing methods often use a tensor rank pre-specified by the user or use a greedy method to update the tensor rank until convergence [24, 31, 32]. These methods often offer inaccurate rank estimation and are complicated in computation. Besides rank determination, another important question is: how can we adaptively add a few simulation samples to update the model with a low computation budget? This is very important in electronic and photonic design automation because obtaining each piece of data sample requires time-consuming device-level or circuit-level numerical simulations.

Paper contributions. We propose a novel tensor regression method for high-dimensional uncertainty quantification. Tensor regression has been studied in machine learning and image data analysis [33–35]. There are some existing works of automatic rank determination [36–38] and adaptive sampling [39, 40] for tensor decomposition and completion. The Bayesian frameworks [41, 42] can enable and guide the adaptive sampling procedure for tensor regression, but limit the model in the meanwhile. Focusing on uncertainty quantification, there are few works about tensor regression and its automatic rank determination and adaptive sampling. The main contributions of this paper include:

- We formulate high-dimensional uncertainty quantification as a tensor regression problem. We further propose a ℓ_q/ℓ_2 group-sparsity regularization method to determine rank automatically. Based on variation equality, the tensor-structured regression problem can be efficiently solved via a block coordinate descent algorithm with an analytical solution in each subproblem.
- We propose a two-stage adaptive sampling method to reduce the simulation cost. This method balances the exploration and exploitation via combining the estimation of Voronoi

The preliminary results of this work were published in EPEPS 2020 [1]. This work was partly supported by NSF grants #1763699 and #1846476.

Zichang He and Zheng Zhang are with Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, USA (e-mails: zichanghe@ucsb.edu; zhengzhang@ece.ucsb.edu).

cell volumes and the nonlinearity of an output function.
- We verify the proposed uncertainty quantification model on a 100-dim synthetic function, a 19-dim photonic band-pass filter, and a 57-dim CMOS ring oscillator. Our model can well capture the high-dimensional stochastic output with only 100-600 samples.

Compared with our conference paper [1], this manuscript presents the following additional results:
- The detailed implementations of the proposed method, including both the compact tensor regression solver and the adaptive sampling procedure (Section III and IV)
- The post-processing step of extracting statistical information from the obtained tensor regression model (Section V).
- The enriched experiments (Section VI), including a demonstrative synthetic example and detailed comparisons with other methods.

II. Notation and Preliminaries

Throughout this paper, a scalar is represented by a lowercase letter, e.g., $x \in \mathbb{R}$; a vector or matrix is represented by a boldface lowercase or capital letter respectively, e.g., $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{X} \in \mathbb{R}^{m\times n}$. A tensor, which describes a multidimensional data array, is represented by a bold calligraphic letter, e.g., $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\cdots\times n_d}$. The $(i_1, i_2, \cdots, i_d)$-th data element of a tensor $\mathcal{X}$ is denoted as $x_{i_1 i_2\cdots i_d}$. Obviously $\mathcal{X}$ reduces to a matrix $\mathbf{X}$ when $d = 2$, and its data element is $x_{i_1 i_2}$. In this section, we will briefly introduce the background of generalized polynomial chaos (gPC) and tensor computation.

A. Generalized Polynomial Chaos Expansion

Let $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_d] \in \mathbb{R}^d$ be a random vector describing fabrication process variations with mutually independent components. We aim to estimate the interested performance metric $y(\boldsymbol{\xi})$ (e.g., chip frequency or power) under such uncertainty. We assume that $y(\boldsymbol{\xi})$ has a finite variance under the process variations. A truncated gPC expansion approximates $y(\xi)$ as the summation of a series of orthornormal basis functions [4]:

$$y(\boldsymbol{\xi}) \approx \hat{y}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}\in\Theta} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \tag{1}$$

where $\boldsymbol{\alpha} \in \mathbb{N}^d$ is an index vector in the index set Θ, $c_{\boldsymbol{\alpha}}$ is the coefficient, and $\Psi_{\boldsymbol{\alpha}}$ is a polynomial basis function of degree $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \cdots + \alpha_d$. One of the most commonly used index set is the total degree one, which selects multivariate polynomials up to a total degree p, i.e.,

$$\Theta = \{\boldsymbol{\alpha} | \alpha_k \in \mathbb{N}, 0 \le \sum_{k=1}^{d} \alpha_k \le p\}, \tag{2}$$

leading to a total of $\frac{(d+p)!}{d!p!}$ terms of expansion. Let $\phi^{(k)}_{\alpha_k}(\xi_k)$ denote the order-α_k univariate basis of the k-th random parameter ξ_k, the multivariate basis is constructed via taking the product of univariate orthornormal polynomial basis:

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \prod_{k=1}^{d} \phi^{(k)}_{\alpha_k}(\xi_k). \tag{3}$$

Therefore, given the joint probability density function $\rho(\boldsymbol{\xi})$, the multivariate basis satisfies the orthornormal condition:

$$\langle \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})\rangle = \int_{\mathbb{R}^d} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})\rho(\boldsymbol{\xi})d\boldsymbol{\xi} = \delta_{\boldsymbol{\alpha},\boldsymbol{\beta}}. \tag{4}$$

The detailed formulation and construction of univariate basis functions can be found in [4, 43].

In order to estimate the unknown coefficients $c_{\boldsymbol{\alpha}}$'s, several popular methods can be used, including intrusive (i.e., non-sampling) methods (e.g., stochastic Galerkin [44] and stochastic testing [6]) and non-intrusive (i.e., sampling) methods (e.g., stochastic collocation based on pseudo-projection or regression [45]). It is well known that gPC expansion suffers the curse of dimensionality. The computational cost grows exponentially as the dimension of $\boldsymbol{\xi}$ increases.

B. Tensor and Tensor Decomposition

Given two tensors $\mathcal{X}$ and $\mathcal{Y} \in \mathbb{R}^{n_1\times n_2\cdots\times n_d}$, their inner product is defined as:

$$\langle \mathcal{X}, \mathcal{Y}\rangle := \sum_{i_1\cdots i_d} x_{i_1\cdots i_d} y_{i_1\cdots i_d}. \tag{5}$$

A tensor $\mathcal{X}$ can be unfolded into a matrix along the k-th mode/dimension, denoted as $\text{Unfold}_k(\mathcal{X}) := \mathbf{X}_{(k)} \in \mathbb{R}^{n_k\times n_1\cdots n_{k-1}n_{k+1}\cdots n_d}$. Conversely, folding the k-mode matrization back to the original tensor is denoted as $\text{Fold}_k(\mathbf{X}_{(k)}) := \mathcal{X}$.

Given a d-dim tensor, it can be factorized as a summation some rank-1 vectors, which is called CANDECOMP/PARAFAC (CP) decomposition [46]:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \cdots \circ \mathbf{a}_r^{(d)} = [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(d)}]\!], \tag{6}$$

where $\circ$ denotes the outer product. The last term is the Krusal form, where factor matrix $\mathbf{A}^{(k)} = \left[\mathbf{a}_1^{(k)}, \ldots, \mathbf{a}_R^{(k)}\right] \in \mathbb{R}^{n_k\times R}$ includes all vectors associated with the k-th dimension. The smallest number of R that ensures the above equality is called a CP rank. The k-th mode unfolding matrix $\mathbf{X}_{(k)}$ can be written with CP factors as

$$\begin{aligned} \mathbf{X}_{(k)} =& \mathbf{A}^{(k)} \mathbf{A}^{(\backslash k)^T} \text{ with} \\ \mathbf{A}^{(\backslash k)} =& \mathbf{A}^{(d)} \odot \cdots \odot \mathbf{A}^{(k-1)} \odot \mathbf{A}^{(k+1)} \cdots \odot \mathbf{A}^{(1)}, \end{aligned} \tag{7}$$

where $\odot$ denotes the Khatri-Rao product, which performs column-wise Kronecker products [46]. More details of tensor operations can be found in [46].

III. Proposed Tensor Regression method

A. Low-Rank Tensor Regression Formulation

To approximate $y(\boldsymbol{\xi})$ as a tensor regression model, we choose a full tensor-product index set for the gPC expansion:

$$\Theta = \{\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_d] \mid 0 \le \alpha_k \le p, \forall k \in [1, d]\}. \tag{8}$$

This specifies a gPC expansion with $(p+1)^d$ basis functions. Let $i_k = \alpha_k + 1$, then we can define two d-dimensional tensors $\mathcal{X}$ and $\mathcal{B}(\boldsymbol{\xi})$ with their $(i_1, i_2, \cdots i_d)$-th elements as

$$x_{i_1 i_2\cdots i_d} = c_{\boldsymbol{\alpha}} \text{ and } b_{i_1 i_2\cdots i_d}(\boldsymbol{\xi}) = \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2021.3093432, IEEE Transactions on Components, Packaging and Manufacturing Technology

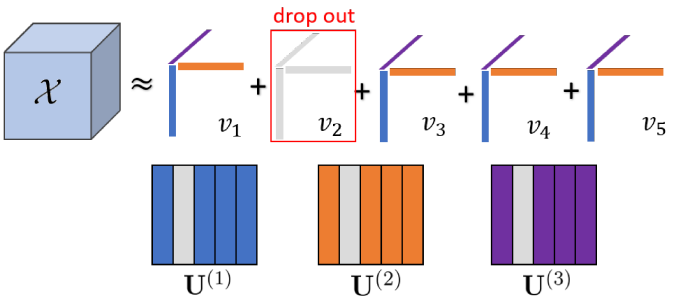


Fig. 1. Visualization of the tensor rank determination. Here the gray vectors denote some shrinking tensor factors that can be removed from a CP decomposition.

Combining Eqs (1), (8) and (9), the truncated gPC expansion can be written as a tensor inner product

$$y(\boldsymbol{\xi}) \approx \hat{y}(\boldsymbol{\xi}) = \langle \mathcal{X}, \mathcal{B}(\boldsymbol{\xi}) \rangle. \tag{10}$$

The tensor $\mathcal{B}(\boldsymbol{\xi}) \in \mathbb{R}^{(p+1)\times\cdots\times(p+1)}$ is a rank-1 tensor that can be exactly represented as:

$$\mathcal{B}(\boldsymbol{\xi}) = \boldsymbol{\phi}^{(1)}(\xi_1) \circ \boldsymbol{\phi}^{(2)}(\xi_2) \circ \cdots \circ \boldsymbol{\phi}^{(d)}(\xi_d), \tag{11}$$

where $\boldsymbol{\phi}^{(k)}(\xi_k) = [\phi_0^{(k)}(\xi_k), \cdots, \phi_p^{(k)}(\xi_k)]^T \in \mathbb{R}^{p+1}$ collects all univariate basis functions of random parameter ξ_k up to order-p.

The unknown coefficient tensor $\mathcal{X}$ has $(p+1)^d$ variables in total, but we can describe it via a rank-R CP approximation:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(d)} = [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(d)}]\!]. \tag{12}$$

It decreases the number of unknown variables to $(p+1)dR$, which only linearly depends on d and thus effectively overcomes the curse of dimensionality.

Our goal is to compute coefficient tensor $\mathcal{X}$ given a set of data samples $\{\boldsymbol{\xi}_n, y(\boldsymbol{\xi}_n)\}_{n=1}^N$ via solving the following optimization problem

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^d} h(\mathcal{X}) = \tfrac{1}{2} \sum_{n=1}^{N} \left(y_n - \langle [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(d)}]\!], \mathcal{B}^n \rangle\right)^2, \tag{13}$$

where $y_n = y(\boldsymbol{\xi}_n)$, $\mathcal{B}^n = \mathcal{B}(\boldsymbol{\xi}_n)$, and $\boldsymbol{\xi}_n$ denotes the n-th sample.

B. Automatic Rank Determination

The low-rank approximation (12) assumes that $\mathcal{X}$ can be well approximated by R rank-1 terms. In practice, it is hard to determine R in advance. In this work, we leverage a group-sparsity regularization function to shrink the tensor rank from an initial estimation. Specifically, define the following vector:

$$\mathbf{v} := [v_1, v_2, \cdots, v_R] \text{ with } v_r = \left(\sum_{k=1}^{d} \|\mathbf{u}_r^{(k)}\|_2^2\right)^{\frac{1}{2}} \forall r \in [1, R]. \tag{14}$$

We further use its ℓ_q norm with $q \in (0, 1]$ to measure the sparsity of $\mathbf{v}$:

$$g(\mathcal{X}) = \|\mathbf{v}\|_q, \quad q \in (0, 1]. \tag{15}$$

This function groups all rank-1 term factors together and enforces the sparsity among R groups. The rank is reduced when the r-th columns of all factor matrices are enforced to zero. When $q = 1$, this method degenerates to a group lasso, and a smaller q leads to a stronger shrinkage force.

Based on this rank-shrinkage function, we compute the tensor-structured gPC coefficients by solving a regularized tensor regression problem:

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^d} f(\mathcal{X}) = h(\mathcal{X}) + \lambda g(\mathcal{X}), \tag{16}$$

where $\lambda > 0$ is a regularization parameter. As shown in Fig. 1, after solving this optimization problem, some columns with the same column indices among all matrices $\mathbf{U}^{(k)}$'s are close to zero. These columns can be deleted and the actual rank of our obtained tensor becomes $\hat{R} \leq R$, where $\hat{R}$ is the number of remaining columns in each factor matrix.

C. A More Tractable Regularization

It is non-trivial to minimize $f(\mathcal{X})$ since $g(\mathcal{X})$ is non-differentiable and non-convex with respect to $\mathbf{U}^{(k)}$'s. Therefore, we replace the regularization function with a more tractable one based on the following variational equality.

Lemma 1 (Variational equality [47]). *Let $\alpha \in (0, 2]$, and $\beta = \frac{\alpha}{2-\alpha}$. For any vector $\mathbf{y} \in \mathbb{R}^p$, we have the following equality*

$$\|\mathbf{y}\|_\alpha = \min_{\boldsymbol{\eta} \in \mathbb{R}_+^p} \frac{1}{2} \sum_{r=1}^{p} \frac{y_r^2}{\eta_r} + \frac{1}{2} \|\boldsymbol{\eta}\|_\beta, \tag{17}$$

where the minimum is uniquely attained for $\eta_r = |y_r|^{2-\alpha} \|\mathbf{y}\|_\alpha^{\alpha-1}, r = 1, 2, \ldots, p$.

Proof. See Appendix A. □

If we take $p = R$, $\alpha = q$, and $y_r = v_r$ (defined in (14)), on the right-hand side of Eq. (17), then we have

$$\hat{g}(\mathcal{X}, \boldsymbol{\eta}) = \frac{1}{2} \sum_{r=1}^{R} \frac{v_r^2}{\eta_r} + \frac{1}{2} \|\boldsymbol{\eta}\|_{\frac{q}{2-q}}. \tag{18}$$

The original rank-shrinking function (15) is equivalent to

$$g(\mathcal{X}) = \min_{\boldsymbol{\eta} \in \mathbb{R}_+^R} \hat{g}(\mathcal{X}, \boldsymbol{\eta}). \tag{19}$$

As a result, we solve the following optimization problem as an alternative to (16):

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^d, \boldsymbol{\eta}} \hat{f}(\mathcal{X}) = h(\mathcal{X}) + \lambda \hat{g}(\mathcal{X}, \boldsymbol{\eta}). \tag{20}$$

D. A Block Coordinate Descent Solver for Problem (20)

Now we present an alternating minimization solver for Problem (20). Specifically, we decompose Problem (20) into $(d+1)$ sub-problems with respect to $\{\mathbf{U}^{(k)}\}_{k=1}^d$ and $\boldsymbol{\eta}$, and we can obtain the analytical solution to each sub-problem.

- **$\mathbf{U}^{(k)}$-subproblem:** By fixing $\boldsymbol{\eta}$ and all tensor factors except $\mathbf{U}^{(k)}$, we can see that the variational equality induces a convex subproblem. Based on the first-order optimality condition, $\mathbf{U}^{(k)}$ can be updated analytically:

$$\text{vec}(\mathbf{U}^{(k)}) = (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \tilde{\boldsymbol{\Lambda}})^{-1} \boldsymbol{\Phi}^T \boldsymbol{y}, \tag{21}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2021.3093432, IEEE Transactions on Components, Packaging and Manufacturing Technology

where $\mathbf{\Phi} = [\mathbf{\Phi}_1, \cdots, \mathbf{\Phi}_N]^T \in \mathbb{R}^{N \times R(p+1)}$ with rows $\mathbf{\Phi}_n^T = \text{vec}\left(\mathbf{B}_{(k)}^n \mathbf{U}^{(\backslash k)}\right)^T$ for any $n \in [1, N]$, $\tilde{\mathbf{\Lambda}} = \text{diag}(\frac{1}{\eta_1}, \ldots, \frac{1}{\eta_R}) \otimes \mathbf{I} \in \mathbb{R}^{R(p+1) \times R(p+1)}$, and $\mathbf{y} \in \mathbb{R}^N$ is a collection of output simulation samples. Here $\otimes$ denotes a Kronecker product, $\mathbf{U}^{(\backslash k)}$ is a series of Khatri-Rao product defined as Eq. (7), and $\mathbf{B}_{(k)}^n$ is the k-th mode matrization of the tensor $\mathcal{B}^n = \mathcal{B}(\boldsymbol{\xi}_n)$. For simplicity, we leave the derivation of Eq. (21) to Appendix B.

- **$\boldsymbol{\eta}$-subproblem:** Suppose that $\{\mathbf{U}^{(k)}\}_{k=1}^d$ are fixed, the formulation of the $\boldsymbol{\eta}$-subproblem is shown in (19). According to lemma 1, we update $\boldsymbol{\eta}$ as

$$\eta_r = (v_r)^{2-q} \|\mathbf{v}\|_q^{q-1} + \epsilon, \tag{22}$$

where $\epsilon > 0$ is a small scalar to avoid numerical issues. Suppose that the tensor rank is reduced in the optimization, i.e., $\mathbf{u}_r^{(k)} = \mathbf{0}$, $\forall k \in [1, d]$, then we can see that η_r will become zero without ϵ.

Algorithm 1: Overall Adaptive Tensor Regression

Input: Initial sample pairs $\{\boldsymbol{\xi}_n, y(\boldsymbol{\xi}_n)\}_{n=1}^N$, unitary polynomial order p, initial tensor rank R
Output: Constructed surrogate model [Eq. (25)]
while *Adaptive sampling does not stop* **do**
  Construct the basis tensor $\mathcal{B}(\boldsymbol{\xi})$
  if *No additional samples* **then**
    Initialize with Eq. (23)
  else
    Initialize $\{\mathbf{U}^{(k)}\}_{k=1}^d$ with the last solution
  while *Tensor regression does not stop* **do**
    for $k = 1, 2, \ldots, d$ **do**
      update $\mathbf{U}^{(k)}$ via Eq. (21)
    Update $\boldsymbol{\eta}$ via Eq. (22)
    Update regularization parameter λ via Eq. (24)
  Shrink the tensor rank to $\hat{R}$ if possible
  Select new sample pairs based on Alg. 2

E. Discussions

We would like to highlight a few key points in practical implementations.

- The solution depends on the initialization process. In the first iteration of updating the k-th factor matrices, we suggest the following initialization

$$\begin{aligned} &\mathbf{\Phi} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, \ldots \mathbf{\Phi}_N]^T \text{ with} \\ &\mathbf{\Phi}_n = \text{vec}(\mathbf{O}_{n,k})^T, \forall n \in [1, N], \\ &\mathbf{O}_{n,k} = \left[\boldsymbol{\phi}^{(k)}(\xi_k^n), \ldots, \boldsymbol{\phi}^{(k)}(\xi_k^n)\right] \in \mathbb{R}^{(p+1) \times R}, \end{aligned} \tag{23}$$

where ξ_k^n is the k-th variable of sample $\boldsymbol{\xi}_n$, $\boldsymbol{\phi}^{(k)}(\xi_k^n) \in \mathbb{R}^{p+1}$ collects all univariate basis functions of ξ_k^n up to degree p, and $\mathbf{O}_{n,k}$ stores R copies of $\boldsymbol{\phi}^{(k)}(\xi_k^n)$. Besides, in the first iteration without adaptive sampling, we set $\boldsymbol{\eta}$ as an all-ones vector multiplied by a scalar factor since we do not have a good initial guess for $\{\mathbf{U}^{(k)}\}_{k=1}^d$. The value of the scalar factor does not influence a lot once it makes Eq. (21) numerically stable. In an adaptive sampling setting (see Section IV), we need to solve (20) after adding new samples. In this case, we use a warm-up initialization by setting the initial guess of $\{\mathbf{U}^{(k)}\}_{k=1}^d$ as the solution obtained based on the last-round sampling, and therefore can initialize $\boldsymbol{\eta}$ via Eq. (22).
- The regularization parameter λ is highly related to the force of rank shrinkage. To adaptively balance the empirical loss and the rank shrinkage term, we suggest an iterative update of the parameter

$$\lambda = \lambda_0 \max(\boldsymbol{\eta}), \tag{24}$$

where λ_0 is chosen via cross validation.
- We stop the block coordinate descent solver for problem (20) when the update of factor matrices $\{\mathbf{U}^{(k)}\}_{k=1}^d$ is below a predefined threshold, or the algorithm reaches a predefined maximal number of iterations.

The overall algorithm, including an adaptive sampling which will be introduced in Section IV, is summarized in Alg. 1. After solving the factor matrices $\{\mathbf{U}^{(k)}\}_{k=1}^d$, i.e. the coefficient tensor $\mathcal{X}$, the surrogate on a sample $\boldsymbol{\xi}$ can be efficiently calculated as

$$\hat{y}(\boldsymbol{\xi}) = \langle \mathcal{X}, \mathcal{B}(\boldsymbol{\xi}) \rangle = \sum_{r=1}^R \prod_{k=1}^d \left[\boldsymbol{\phi}^{(k)}(\xi_k)\right]^T \mathbf{u}_r^{(k)}. \tag{25}$$

In this work, tensor $\mathcal{X}$ is approximated by a low-rank CP decomposition. It is also possible to use other kinds of tensor decompositions. In those cases, although the tensor ranks are defined in different ways, the idea of enforcing group-sparsity over tensor factors still works. It is also worth noting that (20) can be seen as a generalization of weighted group lasso. To further exploit the sparsity structure of the gPC coefficients, many variants can be developed from the statistic regression perspective, including the sparse group lasso, tensor-structured Elastic-Net regression, and so forth [48].

IV. Adaptive Sampling Approach

Another fundamental question in uncertainty quantification is how to select the parameter samples $\boldsymbol{\xi}$ for simulation. We aim to reduce the simulation cost by selecting only a few informative samples for the detailed device- or circuit-level simulations.

Given a set of initial samples Θ, we design a two-stage method to balance the exploration and exploitation in our active sampling process. In the first stage, we estimate the volume of some Voronoi cells via a Monte Carlo method to measure the sampling density in each region. In the second stage, we roughly measure the nonlinearity of $y(\boldsymbol{\xi})$ at some candidate samples via a Taylor expansion. We choose new samples that are located in a low-density region and make $y(\boldsymbol{\xi})$ highly nonlinear. In our implementation, the initial samples $\Theta = \{\boldsymbol{\xi}_n, y(\boldsymbol{\xi}_n)\}_{n=1}^N$ are generated by the Latin Hypercube (LH) sampling method [49]. Specifically, we first generate some standard LH samples $\{\boldsymbol{\zeta}_n^{\text{LH}}\}_{n=1}^N$ in a hyper cube $[0, 1]^d$, then we transform them to the practical parameter space Ω via the inverse transforms of the cumulative distribution function.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2021.3093432, IEEE Transactions on Components, Packaging and Manufacturing Technology

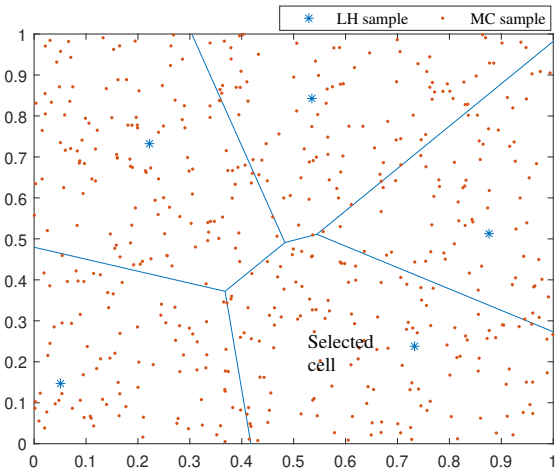


Fig. 2. An example of Voronoi diagram on $[0,1]^2$. Each LH sample is a Voronoi cell center. The lower right cell should be selected in the first-stage since it has the largest estimated area (volume).

Generally, the initial sample size of Θ depends on the number of unknowns in the model. Since problem (20) is regularized and solved via an alternating solver, given a limited simulation budget, we set the initial size N to be smaller than the number of unknowns in our examples.

A. Exploration: Volume Estimation of Voronoi Cells

Firstly, we employ an exploration step via a space-filling sequential design. Given the existing sample set Θ, the sample density in Ω can be estimated via a Voronoi diagram [50]. Specifically, each sample $\boldsymbol{\xi}_n$ corresponds to a Voronoi cell $C_n \in \Omega$ that contains all the samples that lie closely to $\boldsymbol{\xi}_n$ than other samples in Ω. The Voronoi diagram is a complete set of cells that tesselate the whole sampling space. The volume of a cell reflects its sample density: a larger volume means that the cell region is less sampled.

Here we provide a formal description of the Voronoi cell. Given two distinct samples $\boldsymbol{\xi}_i, \boldsymbol{\xi}_j \in \Omega$, there always exist a half-plane $\mathrm{hp}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)$ that contains all samples that are at least as close to $\boldsymbol{\xi}_i$ as to $\boldsymbol{\xi}_j$

$$\mathrm{hp}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) = \{\boldsymbol{\xi} \in \mathbb{R}^d | \ \|\boldsymbol{\xi} - \boldsymbol{\xi}_i\| \leq \|\boldsymbol{\xi} - \boldsymbol{\xi}_j\|\}. \tag{26}$$

The Voronoi cell C_i is defined as the space that lie in the intersection of all half-plane $\mathrm{hp}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j), \forall \boldsymbol{\xi}_j \in \Omega \setminus \boldsymbol{\xi}_i$:

$$C_i = \bigcap_{\boldsymbol{\xi}_j \in \Omega \setminus \boldsymbol{\xi}_i} \mathrm{hp}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j). \tag{27}$$

It is intractable to construct a precise Voronoi diagram and calculate the volume exactly in a high-dimensional space. Fortunately, we do not need to construct the exact Voronoi diagram. Instead, we only need to estimate the volume in order to measure the sample density in that cell. This can be done via a Monte Carlo method.

Observation 1. *In order to detect the least-density region in Ω, we can either estimate the density of Ω directly or estimate the density of hyper-cube $[0,1]^d$ and then transform it to Ω. In the Monte-Carlo-based density estimation, it is fairer to choose the latter one.*

Example. *One simple example is given in Appendix C.*

Based on the above observation, we estimate the volume of Voronoi cell in the hyper cube. Let the existing LH samples $\{\boldsymbol{\zeta}_n^{\mathrm{LH}}\}_{n=1}^N$ be the cell centers $\{C_n\}_{n=1}^N$. We first randomly generate M Monte Carlo samples $\{\boldsymbol{\psi}_m\}_{m=1}^M \in [0,1]^d$. For each random sample, we calculate its Euclidean distance towards the cell centers and assign it to the closest one. Then the volume of the cell $\mathrm{vol}(C_n)$ is estimated by counting the number of assigned random samples. The cell with the largest estimated volume is least-sampled. The MC samples assigned to this cell are denoted as set Γ. A simple example that illustrates the first-round search is shown in Fig. 2. After transforming all Monte Carlo samples in set Γ back to the actual parameter space Ω via the inverse transform sampling method, we obtain a set of candidates for the next-stage selection, denoted as set Γ_Ω.

The accuracy of volume estimation depends on the number of random samples. Clearly, more Monte Carlo samples can estimate the volume more accurately, but they also induce more computational burden. As suggested by [51], to achieve a good estimation accuracy, we use M Monte Carlo samples with $M = 100N$.

B. Exploitation: Nonlinearity Measurement

In the second stage, we aim to do an exploitation search based on the obtained candidate sample set Γ_Ω. Based on the assumption that the region with a more nonlinear response is harder to capture, we choose the criterion in the second stage as the nonlinearity measure of the target function. We know that the first-order Taylor expansion of a function becomes more inaccurate if that function is more nonlinear. Therefore, given a sample $\boldsymbol{\xi}$, we measure the non-linearity of $y(\boldsymbol{\xi})$ via the difference of $y(\boldsymbol{\xi})$ and its first-order Taylor expansion around the closest Voronoi cell center $\mathbf{a} \in \Omega$ [52]. We do not know exactly the expression of $y(\boldsymbol{\xi})$, but we have already built a surrogate model $\hat{y}(\boldsymbol{\xi})$ based on previous simulation samples. Therefore, the nonlinearity of $y(\boldsymbol{\xi})$ can be roughly estimate as

$$\gamma(\boldsymbol{\xi}) = |\hat{y}(\boldsymbol{\xi}) - \hat{y}(\mathbf{a}) - \nabla\hat{y}(\mathbf{a})^T(\boldsymbol{\xi} - \mathbf{a})|. \tag{28}$$

Notice that the nonlinear measure does not imply the accuracy of the surrogate model since we do not use the simulation value here. In the second stage, we will choose the sample $\boldsymbol{\xi}^\star$ that has the largest $\gamma(\boldsymbol{\xi})$ from the candidate set of Γ_Ω:

$$\boldsymbol{\xi}^\star = \underset{\boldsymbol{\xi} \in \Gamma_\Omega}{\mathrm{argmax}} \left(\gamma\left(\boldsymbol{\xi}\right)\right). \tag{29}$$

To summarize, we select the most nonlinear sample from the least-sampled cell space, which is a good trade-off between exploration and exploitation. Based on the above, we summarize the adaptive sampling procedure in Alg. 2.

C. Discussion

The proposed adaptive sampling method can be easily extended to a batch version by searching for the top-K least-sampled regions in the first stage. We can stop sampling

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2021.3093432, IEEE Transactions on Components, Packaging and Manufacturing Technology

Algorithm 2: Adaptive sampling procedure

Input: Initial samples pairs $\Theta = \{\boldsymbol{\xi}_n, y(\boldsymbol{\xi}_n)\}_{n=1}^N$
Output: Sample pairs $\Theta^\star$ with the additional sample
Uniformly generate $M = 100N$ Monte Carlo samples $\{\boldsymbol{\psi}_m\}_{m=1}^M \in [0,1]^d$
for $m = 1, 2, \ldots, M$ **do**
  Find the closest cell C_n center to $\boldsymbol{\psi}_m$
  $\mathrm{vol}(C_n) \leftarrow \mathrm{vol}(C_n) + 1$
Find the cell with the biggest estimated volume vol and the sample set Γ assigned to this cell
$\Gamma_\Omega \leftarrow$ Inverse_transform_sampling(Γ)
Calculate the nonlinearity measure $\gamma(\Gamma_\Omega)$ via Eq. (28)
Select $\boldsymbol{\xi}^\star$ according to Eq. (29)
$\Theta^\star \leftarrow \Theta \bigcup \{\boldsymbol{\xi}^\star, y(\boldsymbol{\xi}^\star)\}$

when we exceed a sampling budget or when the constructed surrogate model achieves the desired accuracy.

The sampling criteria do not rely on the structure of the targeted surrogate model. Therefore, the proposed sampling method is very flexible and generic. The proposed method is very suitable for constructing a high-dimensional polynomial model due to two reasons. Firstly, the number of samples required in estimating the Voronoi cell does not rely on the parameter dimensionality but on the number of existing samples. Secondly, the derivative and the nonlinearity of the surrogate model are easy to compute.

Some variants of the proposed sampling methods may be further developed. For instance, we may define a score function as the combination of the estimated volume and the nonlinearity measure, and then calculate the score for each Monte Carlo sample and select the best one. In the batch version, we may also select several top nonlinear samples from the same Voronoi cell.

V. Statistical Information Extraction

Based on the obtained tensor regression model $\hat{y}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}\in\Theta} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \langle \mathcal{X}, \mathcal{B}(\boldsymbol{\xi}) \rangle$, we can easily extract important statistical information such as moments and Sobol' indices.

- **Moment information.** The mean μ of the constructed $\hat{y}(\cdot)$ is the coefficient of the zero-order basis $\Psi_{\mathbf{0}}(\boldsymbol{\xi})$:

$$\mu = c_{\mathbf{0}} = x_{11\cdots1} = \sum_{r=1}^{R} \boldsymbol{u}_r^{(1)}(1)\boldsymbol{u}_r^{(2)}(1)\cdots\boldsymbol{u}_r^{(d)}(1). \quad (30)$$

  $x_{11\cdots1}$ is the $(1,1,\cdots,1)$-th element of tensor $\mathcal{X}$, and $\mathbf{u}_r^{(k)}(1)$ denotes the first element of vector $\mathbf{u}_r^{(k)}$ The variance of $y(\boldsymbol{\xi})$ can be estimated as:

$$\begin{aligned}\sigma^2 &= \sum_{\boldsymbol{\alpha}\in\Theta, \boldsymbol{\alpha}\neq 0} c_{\boldsymbol{\alpha}} = \langle \mathcal{X}, \mathcal{X} \rangle - x_{11\cdots1}^2 \\ &= \sum_{r_1=1}^{R}\sum_{r_2=1}^{R}\prod_{k=1}^{d} {\mathbf{u}_{r_1}^{(k)}}^T \mathbf{u}_{r_2}^{(k)} - \mu^2.\end{aligned} \quad (31)$$

- **Sobol' indices.** Based on the obtained model, we can also extract the Sobol' indices [53, 54] for global sensitivity analysis. The main sensitivity index S_j measures the contribution by random parameter ξ_j along to the variance $y(\boldsymbol{\xi})$:

$$S_j = \frac{\mathrm{Var}\left[\mathbb{E}\left[y(\boldsymbol{\xi})|\xi_j\right]\right]}{\sigma^2} \quad (32)$$

  where $\mathbb{E}\left[y(\boldsymbol{\xi})|\xi_j\right]$ denotes the conditional expectation of $y(\boldsymbol{\xi})$ over all random variables except ξ_j. The variance of this conditional expectation can be estimated as

$$\begin{aligned}\mathrm{Var}\left[\mathbb{E}\left[y(\boldsymbol{\xi})|\xi_j\right]\right] &= \sum_{i_j=2}^{p+1} x_{1\cdots i_j 1\cdots 1}^2 \\ &= \sum_{i_j=2}^{p+1}\left[\sum_{r=1}^{R} \boldsymbol{u}_r^{(j)}(i_j)\prod_{k\neq j}\boldsymbol{u}_r^{(k)}(1)\right]^2.\end{aligned} \quad (33)$$

  The total sensitivity index T_j measures the contribution to the variance of $y(\boldsymbol{\xi})$ by variable $\boldsymbol{\xi}_j$ and its interactions with all other variables:

$$T_j = 1 - \frac{\mathrm{Var}\left[\mathbb{E}\left[y(\boldsymbol{\xi})|\boldsymbol{\xi}_{\setminus j}\right]\right]}{\sigma^2}. \quad (34)$$

  Here $\boldsymbol{\xi}_{\setminus j}$ includes all elements of $\boldsymbol{\xi}$ except ξ_j. The involved variance of a conditional expectation is estimated as

$$\begin{aligned}&\mathrm{Var}\left[\mathbb{E}\left[y(\boldsymbol{\xi})|\boldsymbol{\xi}_{\setminus j}\right]\right] \\ &= \sum_{(i_1,i_2,\cdots i_d),\, i_j=1} x_{i_1\cdots i_d}^2 - x_{11\cdots1}^2 \\ &= \sum_{r_1=1}^{R}\sum_{r_2=1}^{R}\mathbf{u}_{r_1}^{(j)}(1)\mathbf{u}_{r_2}^{(j)}(1)\prod_{k\neq j}{\mathbf{u}_{r_1}^{(k)}}^T\mathbf{u}_{r_2}^{(k)} - \mu^2.\end{aligned} \quad (35)$$

  Similarly, we can also express any higher-order index representing the effect from the interaction between a set of variables with an analytical form.

VI. Numerical Results

In this section, we will verify the proposed tensor-regression uncertainty quantification method in one synthetic function and two photonic/ electronic IC benchmarks.

A. Baseline Methods for Comparison

We compare our proposed method with the following approaches.

- Tensor regression with adaptive sampling based on space exploration only introduced in Section IV-A (denoted as Space).
- Tensor regression with adaptive sampling based on exploiting nonlinearity only introduced in model with only Section IV-B (denoted as Nonlinear).
- Tensor regression model with random sampling (denoted as Rand). In each iteration of adding samples, new samples are simply randomly selected.
- Fixed-rank tensor regression (denoted as Fixed rank). This method uses a tensor ridge regularization in the regression objective function [35]:

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^d} f(\mathcal{X}) = h(\mathcal{X}) + \lambda\sum_{k=1}^{d}\|\mathbf{U}^{(k)}\|_{\mathrm{F}}^2. \quad (36)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2021.3093432, IEEE Transactions on Components, Packaging and Manufacturing Technology

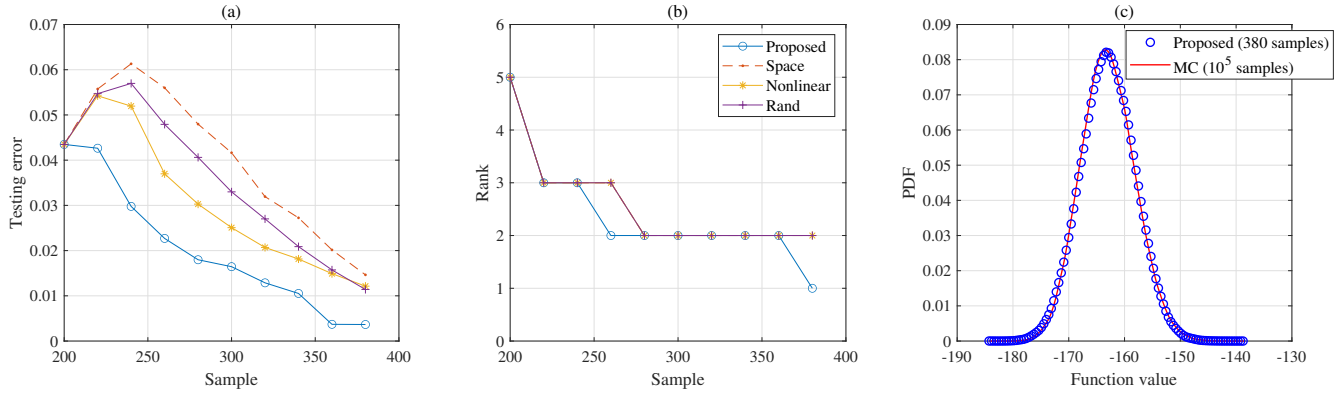


Fig. 3. Results of approximating the synthetic function. (a) Testing error on 10^5 MC samples. (b) The estimated rank. (c) Probability density functions of the function value.

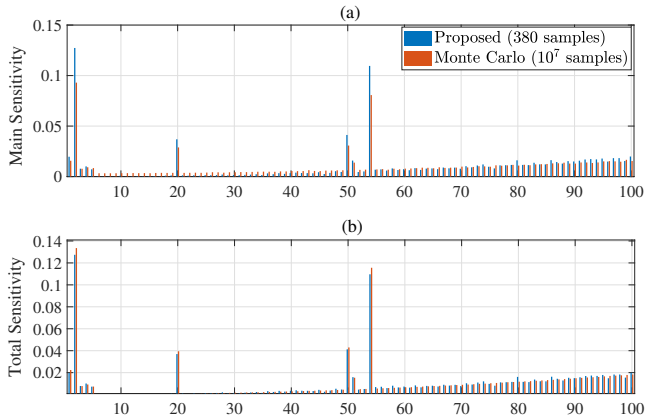


Fig. 4. Sensitivity analysis of the synthetic function in (38). The proposed method fits the results from Monte Carlo [53] with 10^7 simulations very well.

TABLE I
MODEL COMPARISONS ON THE SYNTHETIC FUNCTION

	Sample #	Variable #	Mean	Std	Testing
Monte Carlo	10^5	N/A	-162.95	4.80	N/A
Sparse gPC	380	5151	-163.04	2.27	2.3%
Fixed rank	380	15x100	-163.24	5.02	1.01%
Proposed	380	3x100*	-162.93	4.86	0.37%

* In the alternating solver, there are 100 subproblems with 3 unknown variables in each one (the rank has been shrunk).

The standard ridge regression does not induce a sparse structure. We will keep the tensor rank fixed in solving Eq. (36).

- Sparse gPC expansion with a total degree truncation [55] (denoted as Sparse gPC). With the truncation scheme in Eq. (2), we compute the gPC coefficients by solving the following problem:

$$\min_{\hat{\boldsymbol{c}}} \frac{1}{2}\sum_{n=1}\left(y_n - \sum_{\boldsymbol{\alpha}\in\Theta}\hat{c}_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_n)\right)^2 + \lambda\|\hat{\boldsymbol{c}}\|_1. \quad (37)$$

B. Synthetic Function (100-dim)

We first consider the following high-dimensional analytical function [56]:

$$y(\boldsymbol{\xi}) = 3 - \frac{5}{d}\sum_{k=1}^{d} k\xi_k + \frac{1}{d}\sum_{k=1}^{d} k\xi_k^3 + \xi_1\xi_2^2 + \xi_2\xi_4 - \xi_3\xi_5 + \xi_{51} + \xi_{50}\xi_{54}^2 + \ln\left(\frac{1}{3d}\sum_{k=1}^{d} k(\xi_k^2 + \xi_k^4)\right) \quad (38)$$

where dimension $d = 100$, $\xi_{20} \sim \mathcal{U}([1,3])$, and $\xi_k \sim \mathcal{U}([1,2]), k \neq 20$. We aim to approximate $f(\boldsymbol{\xi})$ by a tensor-regression gPC model and perform sensitivity analysis.

Assume that we use 2nd-order univariate basis functions for each random variable, then we will need 3^{100} multivariate basis functions in total. To approximate the coefficient tensor, we initialize it with a rank-5 CP decomposition and use $q = 0.5$ in regularization. We initialize the training with 200 Latin-Hypercube samples and adaptively select 9 batches of additional samples, with each batch having 20 new samples. We test the accuracy of different models on additional 10^5 samples. Fig. 3 (a) shows the relative ℓ_2 testing errors (i.e., $\frac{\|\boldsymbol{y}(\boldsymbol{\xi})-\hat{\boldsymbol{y}}(\boldsymbol{\xi})\|_2}{\|\boldsymbol{y}(\boldsymbol{\xi})\|_2}$) of different methods. The testing errors may not monotonically decrease since more samples can not strictly guarantee the convergence of the surrogate model. However, see from the figure, we can generally conclude that more training samples lead to a better model approximation and the proposed sampling method outperforms the others. Fig. 3 (b) shows the estimated tensor rank as the number of training samples increases. The proposed method shrinks the tensor rank differently from other methods while achieving the best performance. It shows that a correct determination of the tensor rank helps the function approximation. Fig. 3 (c) plots the predicted probability density function of our obtained model which is estimated via a kernel density estimator. It matches the Monte Carlo simulation result of the original function very well.

We compare the complexity and accuracy of different methods in Table I. We treat the result from 10^5 Monte Carlo simulations as the ground truth. For the other models, the mean and standard deviation are both extracted from the polynomial coefficients. Given the same amount of (limited)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2021.3093432, IEEE Transactions on Components, Packaging and Manufacturing Technology

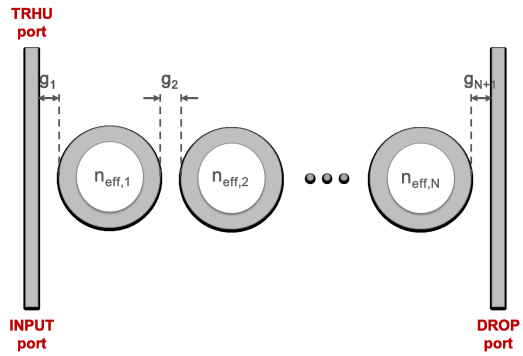


Fig. 5. Schematic of a band-pass filter with 9 micro-ring resonators.

TABLE II
MODEL COMPARISONS ON THE PHOTONIC BAND-PASS FILTER

	Sample #	Variable #	Mean	Std	Error
Monte Carlo	10^5	N/A	21.6511	0.0988	N/A
Sparse gPC	100	210	21.6537	0.0735	0.39%
Fixed rank	100	12x19	21.6677	0.1906	0.52%
Proposed	100	3x19	21.6567	0.0955	0.16%

training samples, the proposed method achieves the highest approximation accuracy.

Now we perform sensitivity analysis to identify the random variables that are most influential to the output. Fig. 4 plots the main and total sensitivity metrics from the proposed method and a Monte Carlo estimation [53] with 10^7 simulations. With much fewer function evaluations, our proposed method can precisely identify the indices of some most dominant random variables that contribute to the output variance.

C. Photonic Band-pass Filter (19-dim)

Now we consider the photonic band-pass filter in Fig. 5. This photonic IC has 9 micro-ring resonators, and it was originally designed to have a 3-dB bandwidth of 20 GHz, a 400-GHz free spectral range, and a 1.55-nm operation wavelength. A total of 19 independent Gaussian random parameters are used to describe the variations of the effective phase index (n_{neff}) of each ring, as well as the gap (g) between adjacent rings and between the first/last ring and the bus waveguides. We aim to approximate the 3-dB bandwidth $f_{3\text{dB}}$ at the DROP port as a tensor-regression gPC model.

We use 2nd order univariate polynomial basis functions for each random parameter and have 3^{19} multivariate basis functions in total in the tensor regression gPC model. We initialize the gPC coefficients as a rank-4 CP tensor decomposition and set $q = 0.5$ in our regularization. We initialize the training with 60 Latin-Hypercube samples and adaptively select 9 batches of additional samples, with each batch have 10 new samples. We test the obtained model with additional 10^5 samples. Fig. 6 (a) shows the relative ℓ_2 testing errors. The proposed method outperforms the others in the first few adaptive sampling rounds. All the models perform similarly when the ranks are all shrunk to 1. Fig. 6 (b) shows the estimated tensor rank as the number of training samples increases. The tensor ranks are shrunk gradually in all cases, but our proposed method finds the best rank with minimal samples. Fig. 6 (c) plots the predicted probability density function of our obtained result. Since the benchmark has a relatively small standard deviation, the limited approximation error is revealed as the discrepancy around the peak.

In order to see the influence of the tensor rank initialization, we do the one-shot approximation with different initial tensor ranks R and different regularization parameters λ as illustrated in Fig. 7. For a specific benchmark, the best-estimated tensor rank highly depends on the number of training samples. Given the limited number of simulation samples, the rank-1 initialization works the best in this example. It coincides with the results shown in Fig. 6, where the predicted rank is 1. We also compare the complexity and accuracy of all methods in Table II. The proposed method achieves the best accuracy with limited simulation samples.

TABLE III
MODEL COMPARISONS ON THE CMOS RING OSCILLATOR

	Sample #	Variable #	Mean	Std	Error
Monte Carlo	3×10^4	N/A	12.7920	0.3829	N/A
Sparse gPC	600	1711	12.7931	0.3777	0.11%
Fixed rank	600	12x57	12.7929	0.3822	0.10%
Proposed	600	6x57	12.7918	0.3830	0.04%

D. CMOS Ring Oscillator (57-dim)

We continue to consider the 7-stage CMOS ring oscillator in Fig. 8. This circuit has 57 random variation parameters, including Gaussian parameters describing the temperature, variations of threshold voltages and gate-oxide thickness, and uniform-distribution parameters describing the effective gate length/width. We aim to approximate the oscillator frequency with tensor-regression gPC under the process variations.

We use 2nd-order univariate basis functions for each random parameter, leading to 3^{57} multivariate basis functions in total. We initialize the gPC coefficients as a rank-4 tensor and set $q = 0.5$ in the regularization term. We initialize the training with 500 Latin-Hypercube samples and adaptively select 300 additional samples in total by 6 batches. We test the obtained model with 3×10^4 additional samples. Fig. 10 (a) shows the relative ℓ_2 testing errors of all methods. The proposed method outperforms other methods significantly when the number of samples is small. Fig. 10 (b) shows that the estimated tensor rank reduces to 2 in all methods. Fig. 10 (c) plots the predicted probability density function of the obtained tensor regression model, which is indistinguishable from the result of Monte Carlo simulations.

We do the one-shot approximation with different initial tensor ranks R and different regularization parameters λ as illustrated in Fig. 9. Conforming with the results shown in Fig. 10, a rank-2 model is more suitable in this example. We compare the proposed method with the fixed rank model and the 2nd-order sparse gPC in Table III, where the proposed compact tensor model is shown to have the best approximation accuracy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2021.3093432, IEEE Transactions on Components, Packaging and Manufacturing Technology

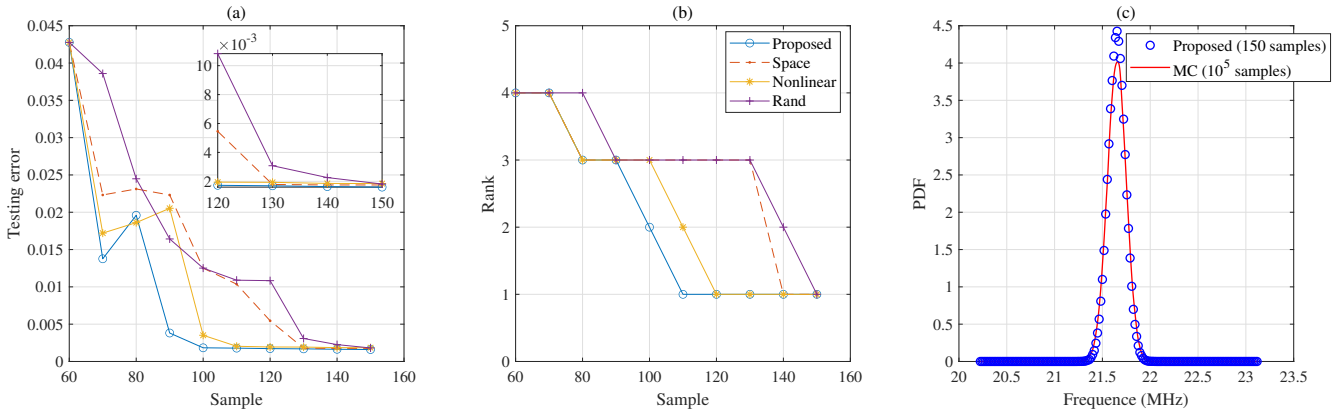


Fig. 6. Result of the photonic filter. (a) Testing error on 10^5 MC samples. (b) The estimated rank. (c) Probability density functions of the 3-dB bandwidth $f_{3\text{dB}}$ at the DROP port.

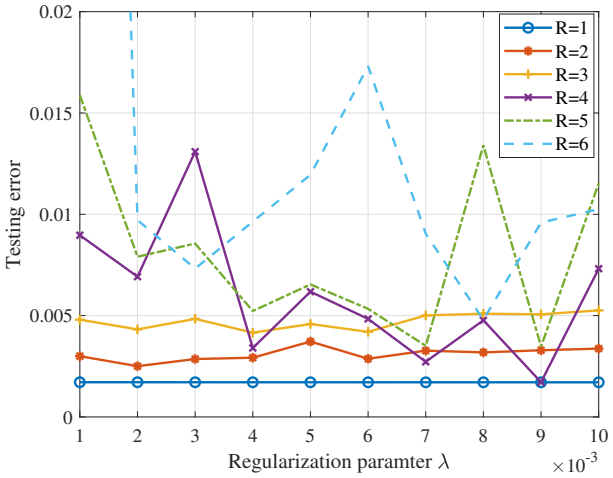


Fig. 7. One-shot approximations for the photonic band-pass filter with 800 training samples under different ranks and λ. The rank-1 initialization works the best in this example.

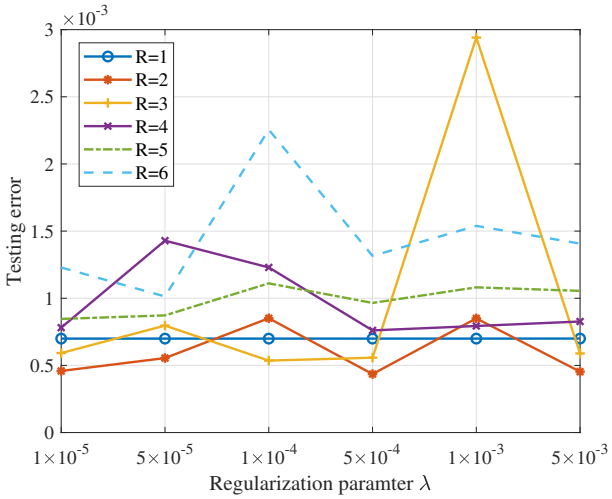


Fig. 9. One-shot approximations for the CMOS ring oscillator with 150 training samples under different ranks and λ. In this example, the rank-2 model works the best in most cases.

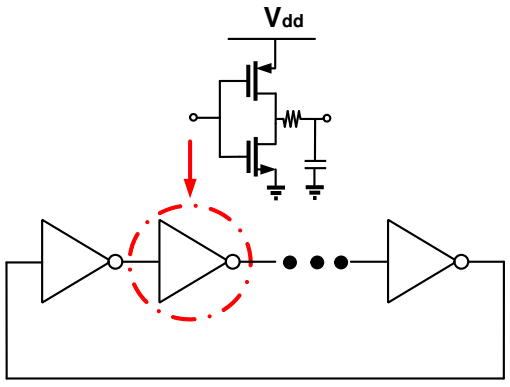


Fig. 8. Schematic of a CMOS ring oscillator.

VII. Conclusion

This paper has proposed a tensor regression framework for quantifying the impact of high-dimensional process variations. By low-rank tensor representation, this formulation can reduce the number of unknown variables from an exponential function of parameter dimensionality to only a linear one. Therefore it works well with a limited simulation budget. We have addressed two fundamental challenges: automatic tensor rank determination and adaptive sampling. The tensor rank is estimated via a ℓ_q/ℓ_2-norm regularization. The simulation samples are chosen based on a two-stage adaptive sampling method, which utilizes the Voronoi cell volume estimation and the nonlinearity measure of the quantity of interest. Our model has been verified by both synthetic and realistic examples with 19 to 100 random parameters. The numerical experiments have shown that our method can well capture the high-dimensional stochastic performance with much fewer simulation data.

Appendix A
Proof of Lemma 1

We consider two cases $\alpha \in (0, 2)$ [47] and $\alpha = 2$ [35].

When $\alpha \in (0, 2)$, $\kappa(\boldsymbol{\eta}) := \frac{1}{2}\sum_{r=1}^{p} \frac{y_r^2}{\eta_r} + \frac{1}{2}\|\boldsymbol{\eta}\|_\beta$ is a continuously differentiable function for any $\eta_i \in (0, \infty)$. When

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2021.3093432, IEEE Transactions on Components, Packaging and Manufacturing Technology

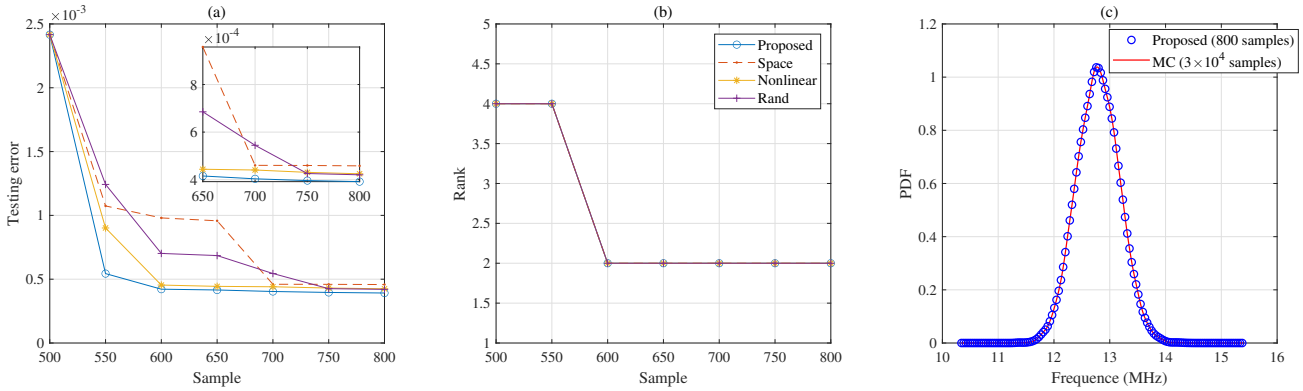


Fig. 10. Results of the CMOS ring oscillator. (a) Testing error on 3×10^4 MC samples. (b) The estimated rank. (c) Probability density functions of the oscillator frequency.

$y_r \neq 0$, $\lim_{\eta_r \to \infty} \kappa(\boldsymbol{\eta}) = \infty$ and $\lim_{\eta_r \to 0} \kappa(\boldsymbol{\eta}) = \infty$. Therefore, the infimum of $\kappa(\boldsymbol{\eta})$ exists and it is attained. According to the first-order optimality and enforcing the derivative w.r.t. η_r $(\eta_r > 0)$ to be zero, we can obtain

$$\eta_r = |y_r|^{2-\alpha} \|\boldsymbol{\eta}\|_{\frac{\alpha}{2-\alpha}}^{\alpha-1}. \tag{39}$$

With $y_r = \|\boldsymbol{\eta}\|_{\frac{\alpha}{2-\alpha}}^{\frac{1-\alpha}{2-\alpha}} \eta_r^{\frac{1}{2-\alpha}}$, we have $\|\boldsymbol{\eta}\|_{\frac{\alpha}{2-\alpha}} = \|\boldsymbol{\eta}\|_{\frac{\alpha}{2-\alpha}}^{\frac{1-\alpha}{2-\alpha}} (\sum_{r=1}^{p} \eta_r^{\frac{\alpha}{2-\alpha}})^{\frac{1}{\alpha}} = \|\mathbf{y}\|_\alpha$, therefore we obtain the optimal solution $\eta_r = |y_r|^{2-\alpha} \|\mathbf{y}\|_\alpha^{\alpha-1}$ in Lemma 1. If $y_r = 0$, the solution to $\min_{\boldsymbol{\eta} \geq 0} \kappa(\boldsymbol{\eta})$ is $\eta_r = 0$, which is also consistent with Lemma 1.

When $\alpha = 2$, $\|\boldsymbol{\eta}\|_1$ is non-differentiable. Given a scalar y_r, we have $y_r = \frac{y_r^2}{2\eta_r} + \frac{1}{2}\eta_r$ only when $\eta_r = y_r$ (we let $\frac{y_r^2}{2\eta_r} = 0$ when $y_r = \eta_r = 0$). Similarly, given a vector $\mathbf{y} \in \mathbb{R}^p$, we have $\|\mathbf{y}\|_1 = \frac{1}{2}\sum_{r=1}^{p} \frac{y_r^2}{\eta_r} + \frac{1}{2}\|\boldsymbol{\eta}\|_1$ only when $\boldsymbol{\eta} = |\mathbf{y}|$, which is also consistent with Lemma 1.

Appendix B
Detailed Derivation of Eq. (21)

Let $\boldsymbol{\Lambda} = \text{diag}(\frac{1}{\eta_1}, \ldots, \frac{1}{\eta_R})$ and $*$ denote a Hadamard product, we can rewrite the objective function of an $\mathbf{U}^{(k)}$-subproblem as

$$\begin{aligned} & f_k(\mathbf{U}^{(k)}) \\ =& \frac{1}{2}\sum_{n=1}^{N} \left[y_n - \langle \mathbf{U}^{(k)} \mathbf{U}^{(\backslash k)^T}, \mathbf{B}_{(k)}^n \rangle \right]^2 + \frac{\lambda}{2} \sum_{r=1}^{R} \frac{\|\mathbf{u}_r^{(k)}\|_2^2}{\eta_r} \\ =& \frac{1}{2}\sum_{n=1}^{N} \left[y_n - \text{Tr}\left(\mathbf{U}^{(k)} \left(\tilde{\mathbf{B}}_{(k)}^n \right)^T \right) \right]^2 + \frac{\lambda}{2} \text{Tr}(\mathbf{U}^{(k)} \boldsymbol{\Lambda} \mathbf{U}^{(k)^T}) \end{aligned}$$

with $\tilde{\mathbf{B}}_{(k)}^n = \mathbf{B}_{(k)}^n \mathbf{U}^{(\backslash k)}$. When the dimension d is large, it is intractable to store and compute $\mathbf{B}_{(k)}^n \in \mathbb{R}^{(p+1)\times(p+1)^{(d-1)}}$ or $\mathbf{U}^{(\backslash k)} \in \mathbb{R}^{(p+1)^{(d-1)}\times R}$. Fortunately, based on the property of Khatri-Rao product, we can compute $\tilde{\mathbf{B}}_{(k)}^n$ as

$$\begin{aligned} & \tilde{\mathbf{B}}_{(k)}^n = \mathbf{B}_{(k)}^n \mathbf{U}^{(\backslash k)} \\ & = \boldsymbol{\phi}^{(k)}(\xi_k^n)[\boldsymbol{\phi}^{(d)}(\xi_d^n)^T \mathbf{U}^{(d)} * \cdots \boldsymbol{\phi}^{(k+1)}(\xi_{k+1}^n)^T \mathbf{U}^{(k+1)} * \\ & \boldsymbol{\phi}^{(k-1)}(\xi_{k-1}^n)^T \mathbf{U}^{(k-1)} * \cdots \boldsymbol{\phi}^{(1)}(\xi_1^n)^T \mathbf{U}^{(1)}]. \end{aligned}$$

Enforcing the following 1st-order optimality condition

$$\begin{aligned} & \frac{\partial f_k(\mathbf{U}^{(k)})}{\partial \mathbf{U}^{(k)}} = \\ & -\frac{1}{2}\sum_{n=1}^{N} \left[y_n - \text{Tr}(\mathbf{U}^{(k)} (\tilde{\mathbf{B}}_{(k)}^n)^T) \right] \tilde{\mathbf{B}}_{(k)}^n + \lambda \mathbf{U}^{(k)} \boldsymbol{\Lambda} = \mathbf{0}, \end{aligned}$$

we can obtain the analytical solution in Eq. (21).

Appendix C
An Example to Show Observation 1

Suppose we already have two samples $[0.2, 0.6]$, and we consider a candidate sample 0.4 in the interval $[0, 1]$ equipped with a uniform distribution. Then, based on Box–Muller transform, their corresponding Gaussian-distributed samples are $[-0.8416, 0.2533]$ and -0.2533, respectively. It is easy to know that the PDF value of sample 0.2533 is larger than sample -0.8416 in a standard Gaussian distribution. Apparently, the candidate sample is equally close to the two examples in a uniform-sampled space, but it is closer to the one with a higher probability density in the Gaussian-sampled space.

References

[1] Z. He and Z. Zhang, "High-dimensional uncertainty quantification via active and rank-adaptive tensor regression," in *Proc. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, 2020, pp. 1–3.

[2] D. S. Boning, K. Balakrishnan, H. Cai, N. Drego, A. Farahanchi, K. M. Gettings, D. Lim, A. Somani, H. Taylor, D. Truque *et al.*, "Variation," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 1, pp. 63–71, 2008.

[3] M. H. Kalos and P. A. Whitlock, *Monte Carlo methods*. John Wiley & Sons, 2009.

[4] D. Xiu and G. E. Karniadakis, "Modeling uncertainty in flow simulations via generalized polynomial chaos," *J. Comput. Phys.*, vol. 187, no. 1, pp. 137–167, 2003.

[5] K. Strunz and Q. Su, "Stochastic formulation of SPICE-type electronic circuit simulation with polynomial chaos," *ACM Trans. Model. Comput. Simul.*, vol. 18, no. 4, pp. 1–23, 2008.

[6] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1533–1545, 2013.

[7] P. Manfredi, D. V. Ginste, D. De Zutter, and F. G. Canavero, "Stochastic modeling of nonlinear circuits via SPICE-compatible spectral equivalents," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 7, pp. 2057–2065, 2014.

[8] Z. He, W. Cui, C. Cui, T. Sherwood, and Z. Zhang, "Efficient uncertainty modeling for system design via mixed integer programming," in *Proc. Intl. Conf. Computer Aided Design*, 2019, pp. 1–8.

[9] M. Ahadi and S. Roy, "Sparse linear regression (SPLINER) approach for efficient multidimensional uncertainty quantification of high-speed circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 10, pp. 1640–1652, 2016.

[10] P. Manfredi and S. Grivet-Talocia, "Rational polynomial chaos expansions for the stochastic macromodeling of network responses," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 1, pp. 225–234, 2019.

[11] P. Manfredi, D. V. Ginste, D. De Zutter, and F. G. Canavero, "Generalized decoupled polynomial chaos for nonlinear circuits with many random parameters," *IEEE Microw. Wireless Compon. Lett.*, vol. 25, no. 8, pp. 505–507, 2015.

[12] A. C. Yücel, H. Bağcı, and E. Michielssen, "An ME-PC enhanced HDMR method for efficient statistical analysis of multiconductor transmission line networks," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 5, no. 5, pp. 685–696, 2015.

[13] F. Wang, P. Cachecho, W. Zhang, S. Sun, X. Li, R. Kanj, and C. Gu, "Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 8, pp. 1255–1268, 2015.

[14] S. Zhang, W. Lyu, F. Yang, C. Yan, D. Zhou, X. Zeng, and X. Hu, "An efficient multi-fidelity Bayesian optimization approach for analog circuit synthesis," in *Proc. Design Autom. Conf*, 2019, pp. 1–6.

[15] C. Cui and Z. Zhang, "Stochastic collocation with non-Gaussian correlated process variations: Theory, algorithms, and applications," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 9, no. 7, pp. 1362–1375, 2018.

[16] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance variability modeling of analog/RF circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 11, pp. 1661–1668, 2010.

[17] C. Cui and Z. Zhang, "High-dimensional uncertainty quantification of electronic and photonic IC with non-Gaussian correlated process variations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 8, pp. 1649–1661, 2019.

[18] M. Ahadi, A. K. Prasad, and S. Roy, "Hyperbolic polynomial chaos expansion (HPCE) and its application to statistical analysis of nonlinear circuits," in *Proc. IEEE Workshop Signal Power Integr.*, 2016, pp. 1–4.

[19] X. Ma and N. Zabaras, "An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations," *J. Comput. Phys.*, vol. 229, no. 10, pp. 3884–3915, 2010.

[20] X. Yang, M. Choi, G. Lin, and G. E. Karniadakis, "Adaptive ANOVA decomposition of stochastic incompressible and compressible flows," *J. Comput. Phys.*, vol. 231, no. 4, pp. 1587–1614, 2012.

[21] T. El-Moselhy and L. Daniel, "Variation-aware interconnect extraction using statistical moment preserving model order reduction," in *Proc. Design, Autom. Test Eur. Conf. Exhibit.*, 2010, pp. 453–458.

[22] Z. Zhang, X. Yang, G. Marucci, P. Maffezzoni, I. A. M. Elfadel, G. Karniadakis, and L. Daniel, "Stochastic testing simulator for integrated circuits and MEMS: Hierarchical and sparse techniques," in *Proc. IEEE Custom Integrated Circuits Conference*, 2014, pp. 1–8.

[23] Z. Zhang, X. Yang, I. V. Oseledets, G. E. Karniadakis, and L. Daniel, "Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 1, pp. 63–76, 2014.

[24] Z. Zhang, T.-W. Weng, and L. Daniel, "Big-data tensor recovery for high-dimensional uncertainty quantification of process variations," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 7, no. 5, pp. 687–697, 2017.

[25] ——, "A big-data approach to handle process variations: Uncertainty quantification by tensor recovery," in *Proc. IEEE Workshop Signal Power Integr.*, 2016, pp. 1–4.

[26] P. Rai, "Sparse low rank approximation of multivariate functions–applications in uncertainty quantification," Ph.D. dissertation, Ecole Centrale de Nantes (ECN), 2014.

[27] K. Konakli and B. Sudret, "Polynomial meta-models with canonical low-rank approximations: Numerical insights and comparison to sparse polynomial chaos expansions," *J. Comput. Phys.*, vol. 321, pp. 1144–1169, 2016.

[28] M. Chevreuil, R. Lebrun, A. Nouy, and P. Rai, "A least-squares method for sparse low rank approximation of multivariate functions," *SIAM/ASA Int. J. Uncertain. Quantif.*, vol. 3, no. 1, pp. 897–921, 2015.

[29] A. Nouy, *Low-rank methods for high-dimensional approximation and model order reduction*. Philadelphia: SIAM, 2017, pp. 171–226.

[30] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *Journal of the ACM (JACM)*, vol. 60, no. 6, pp. 1–39, 2013.

[31] X. Shi, H. Yan, Q. Huang, J. Zhang, L. Shi, and L. He, "Meta-model based high-dimensional yield analysis using low-rank tensor approximation," in *Proc. Design Autom. Conf*, 2019, pp. 1–6.

[32] K. Tang and Q. Liao, "Rank adaptive tensor recovery based model reduction for partial differential equations with high-dimensional random inputs," *J. Comput. Phys.*, vol. 409, p. 109326, 2020.

[33] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *J. Amer. Statist. Assoc.*, vol. 108, no. 502, pp. 540–552, 2013.

[34] J. Kossaifi, Z. C. Lipton, A. Kolbeinsson, A. Khanna, T. Furlanello, and A. Anandkumar, "Tensor regression networks," *J. Mach. Learn. Res.*, vol. 21, pp. 1–21, 2020.

[35] W. Guo, I. Kotsia, and I. Patras, "Tensor learning for regression," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 816–827, 2011.

[36] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, 2015.

[37] J. Luan and Z. Zhang, "Prediction of multidimensional spatial variation data via Bayesian tensor completion," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 2, pp. 547–551, 2019.

[38] C. Li and Z. Sun, "Evolutionary topology search for tensor network decomposition," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5947–5957.

[39] A. Krishnamurthy and A. Singh, "Low-rank matrix and tensor completion via adaptive sampling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.

[40] Z. He, B. Zhao, and Z. Zhang, "Active sampling for accelerated MRI with low-rank tensors," *arXiv preprint arXiv:2012.12496*, 2020.

[41] R. Guhaniyogi, S. Qamar, and D. B. Dunson, "Bayesian tensor regression," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2733–2763, 2017.

[42] R. Yu, G. Li, and Y. Liu, "Tensor regression meets Gaussian processes," in *Proc. Int. Conf. Artif. Intell. Stat.* PMLR, 2018, pp. 482–490.

[43] W. Gautschi, "On generating orthogonal polynomials," *SIAM J. Sci. Stat. Comp.*, vol. 3, no. 3, pp. 289–317, 1982.

[44] R. G. Ghanem and P. D. Spanos, "Stochastic finite element method: Response statistics," in *Stochastic finite elements: a*

spectral approach, 1991, pp. 101–119.

[45] D. Xiu, *Stochastic collocation methods: A survey*. Cham: Springer International Publishing, 2016, pp. 1–18.

[46] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[47] R. Jenatton, G. Obozinski, and F. Bach, "Structured sparse principal component analysis," in *Proc. Intl. Conf. Artif. Intell. Stat.*, 2010, pp. 366–373.

[48] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[49] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[50] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.

[51] K. Crombecq, I. Couckuyt, D. Gorissen, and T. Dhaene, "Space-filling sequential design strategies for adaptive surrogate modelling," in *Int. Conf. Soft Computing Techn. in Civil, Structural and Environmental Engineering*, vol. 38, 2009.

[52] S. Mo, D. Lu, X. Shi, G. Zhang, M. Ye, J. Wu, and J. Wu, "A Taylor expansion-based adaptive design strategy for global surrogate modeling with applications in groundwater modeling," *Water Resour. Res.*, vol. 53, no. 12, pp. 10 802–10 823, 2017.

[53] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, "Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index," *Comput. Phys. Commun.*, vol. 181, no. 2, pp. 259–270, 2010.

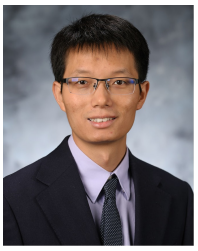
[54] I. M. Sobol, "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates," *Math. Comput. Simulation*, vol. 55, no. 1-3, pp. 271–280, 2001.

[55] N. Luthen, S. Marelli, and B. Sudret, "Sparse polynomial chaos expansions: Literature survey and benchmark," *SIAM/ASA J. Uncertain. Quantif.*, vol. 9, no. 2, pp. 593–649, 2021.

[56] S. Marelli and B. Sudret, "UQLab: A framework for uncertainty quantification in Matlab," in *Vulnerability, uncertainty, and risk: quantification, mitigation, and management*, 2014, pp. 2554–2563.

Zichang He received the B.E. degree in Detection, Guidance and Control Technology in 2018 from Northwestern Polytechnical University, Xi'an, China. In 2018 he joined the Department of Electrical and Computer Engineering at University of California, Santa Barbara as a Ph.D. student.

Zichang's research activities are mainly focused on uncertainty quantification and tensor related topics with applications on design automation, machine learning, and quantum computing. He is the recipient of best student paper award in IEEE Electrical Performance of Electronic Packaging and Systems (EPEPS) conference in 2020 and the Outstanding Teaching Assistant award in the department of ECE at UCSB in 2020 and 2021.



Zheng Zhang (M'15) received his Ph.D degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 2015. He is an Assistant Professor of Electrical and Computer Engineering with the University of California at Santa Barbara (UCSB), CA. His research interests include uncertainty quantification and tensor computational methods with applications to multi-domain design automation, robust/safe and high-dimensional machine learning and its algorithm/hardware co-design.

Dr. Zhang received the Best Paper Award of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems in 2014, two Best Paper Awards of IEEE Transactions on Components, Packaging and Manufacturing Technology in 2018 and 2020, and three Best Conference Paper Awards (IEEE EPEPS 2018 and 2020, IEEE SPI 2016). His Ph.D. dissertation was recognized by the ACM SIGDA Outstanding Ph.D. Dissertation Award in Electronic Design Automation in 2016, and by the Doctoral Dissertation Seminar Award (i.e., Best Thesis Award) from the Microsystems Technology Laboratory of MIT in 2015. He received the NSF CAREER Award in 2019 and Facebook Research Award in 2020.