High-Dimensional Uncertainty Quantification via Active and Rank-Adaptive Tensor Regression

Zichang He and Zheng Zhang

Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106

Emails: zichanghe@ucsb.edu, zhengzhang@ece.ucsb.edu

Abstract—Uncertainty quantification based on stochastic spectral methods suffers from the curse of dimensionality. This issue was mitigated recently by low-rank tensor methods. However, there exist two fundamental challenges in low-rank tensor-based uncertainty quantification: how to automatically determine the tensor rank and how to pick the simulation samples. This paper proposes a novel tensor regression method to address these two challenges. Our method uses an $\ell_{2,p}$-norm regularization to determine the tensor rank and an estimated Voronoi diagram to pick informative samples for simulation. The proposed framework is verified by a 19-dim phonics bandpass filter and a 57-dim CMOS ring oscillator, capturing the high-dimensional uncertainty well with only 90 and 290 samples respectively.

I. INTRODUCTION

Fabrication process variations are a major concern in nano-scale chip design. To estimate and quantify the uncertainties caused by process variations, Monte Carlo (MC) is the mainstream uncertainty quantification (UQ) tool used in commercial EDA tools, but it requires a huge amount of simulation samples. Instead, stochastic spectral methods based on generalized polynomial chaos (gPC) [1] offer an efficient alternative by approximating a stochastic circuit performance metric as a linear combination of some basis functions [2–4]. However, stochastic spectral methods suffer from the curse of dimensionality: a huge amount of simulation samples are required when the number of random parameters is large.

Low-rank tensor methods are a promising technique to solve high-dimensional UQ problems [5–8]. In [6], a high-dimensional gPC expansion is obtained via a low-rank tensor recovery, which estimates massive unknown output samples from a few simulation results. However, the method [6] uses a fixed tensor rank, which is hard to estimate *a-priori* in practice. The most recent work [9] uses a greedy rank-1 update until a good accuracy is reached. However, greedy rank-1 tensor update does not provide optimal solutions and can cause over fitting. Besides, it is not clear how to adaptively pick the simulation samples to reduce the computation budget.

Contributions. This paper proposes a novel high-dimensional UQ solver based on tensor regression. In order to automatically determine the tensor rank, we employ a group-sparsity regularization in the training process. We also develop an adaptive sampling strategy to reduce the simulation cost. This method balances exploration and exploitation of our

model. Our method is used to quantify the uncertainties of a 19-dim phonic IC and a 57-dim electronic IC with 90 and 290 simulation samples respectively.

II. BACKGROUND

Generalized Polynomial Chaos. Let $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_d] \in \mathbb{R}^d$ be a random vector describing process variations. We aim to estimate the interested performance metric $y(\boldsymbol{\xi})$ (e.g., chip frequency or power) under such uncertainty. A truncated gPC expansion approximates $y(\xi)$ as

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

where $\boldsymbol{\alpha}$ is an index vector, and $\Psi_{\boldsymbol{\alpha}}$ is a polynomial basis function of degree $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \cdots + \alpha_d$. If the joint probability density function of $\boldsymbol{\xi}$ is $\rho(\boldsymbol{\xi})$, then the basis functions satisfy the orthornomal condition:

$$\langle \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi}) \rangle = \int_{\mathbb{R}^d} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) \Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi}) d\boldsymbol{\xi} = \delta_{\boldsymbol{\alpha},\boldsymbol{\beta}}. \qquad (2)$$

Once the index set Θ is chosen, we need to determine the unknown coefficient $c_{\boldsymbol{\alpha}}$ for each $\boldsymbol{\alpha} \in \Theta$. The gPC only requires a small number of basis functions and simulation samples when the parameter dimensionality d is small. However, a huge number of basis functions and simulation samples are required when d is large. For instance, in the classical stochastic collocation method [1], the number of simulation samples required to obtain $c_{\boldsymbol{\alpha}}$'s is an exponential or polynomial function of d.

Tensors. A promising tool to overcome the curse of dimensionality is tensors. A d-dim tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots n_d}$ represents a d-dimensional data array, and it becomes a matrix when $d = 2$. The $(i_1, \cdots, i_d)$-th element of $\mathcal{X}$ can be denoted as $x_{i_1 \cdots i_d}$. Given two tensors $\mathcal{X}$ and $\mathcal{Y}$ of same size, their inner product is defined as:

$$\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i_1 \cdots i_d} x_{i_1 \cdots i_d} y_{i_1 \cdots i_d}. \qquad (3)$$

A d-dim rank-R tensor can be written as the sum of R rank-1 tensors, known as a CP decomposition:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \cdots \circ \mathbf{u}_r^{(d)} = [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(d)}]\!], \quad (4)$$

where $\circ$ denotes an outer product. The last term is the Krusal form, where factor matrix $\mathbf{U}^{(k)} = \left[\mathbf{u}_1^{(k)}, \ldots, \mathbf{u}_R^{(k)}\right] \in \mathbb{R}^{n_k \times R}$ includes all vectors associated with the k-th dimension.

III. PROPOSED TENSOR REGRESSION METHOD

A. Tensor Regression Formulation

We choose the following index set for the gPC expansion:

$$\Theta = \left\{ \boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_d] \mid 0 \le \{\alpha_i\}_{i=1}^d \le p \right\}. \quad (5)$$

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 element specified as

$$x_{i_1 i_2 \cdots i_d} = c_{\boldsymbol{\alpha}} \text{ and } b_{i_1 i_2 \cdots i_d}(\boldsymbol{\xi}) = \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \quad (6)$$

Combining (1), (5) and (6), the truncated gPC expansion can be written as a tensor regression model

$$y(\boldsymbol{\xi}) \approx \hat{y}(\boldsymbol{\xi}) = \langle \mathcal{X}, \mathcal{B}(\boldsymbol{\xi}) \rangle. \quad (7)$$

It is worth noting that tensor $\mathcal{B}(\boldsymbol{\xi})$ depends on $\boldsymbol{\xi}$. When the random parameters $\boldsymbol{\xi}$ are mutually independent, $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi})$ can be written as the product of d uni-variable basis functions for each parameter ξ_k. In this case $\mathcal{B}(\boldsymbol{\xi})$ is a rank-1 tensor.

Our goal is to compute $\mathcal{X}$ given a set of data samples $\{\boldsymbol{\xi}_n, y(\boldsymbol{\xi}_n)\}_{n=1}^N$. Assume that $\mathcal{X}$ has the rank-R decomposition in (4), we can solve the following optimization problem

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^d} h(\mathcal{X}) = \frac{1}{2} \sum_{n=1}^N \left(y_n - \langle [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(d)}]\!], \mathcal{B}_n \rangle \right)^2, \quad (8)$$

where $y_n = y(\boldsymbol{\xi}_n)$ and $\mathcal{B}_n = \mathcal{B}(\boldsymbol{\xi}_n)$.

B. Automatic Rank Determination

The low-rank tensor regression (8) requires the rank of $\mathcal{X}$ to be determined *a-priori*, which is often infeasible in practice. In order to address this issue, we first choose a sufficiently large R such that it is above the actual rank, then we choose a proper rank-shrinking penalty function to regularize (8).

Specifically, we employ a group $\ell_{2,q}$-norm regularization function to shrink the rank of $\mathcal{X}$:

$$g(\mathcal{X}) = \sum_{r=1}^R \left[\left(\sum_{k=1}^d \|\mathbf{u}_r^{(k)}\|_2^2 \right)^{\frac{1}{2}} \right]^{\frac{1}{q}}, \quad q \in [0,1]. \quad (9)$$

This function puts $\mathbf{u}_r^{(k)}$, the r-th column of each $\mathbf{U}^{(k)}$, in the same group, and measures the $\ell_{2,p}$ norm of all groups. As a result, one can shrink some groups to zero by reducing $g(\mathcal{X})$, leading to an automatical rank reduction. A smaller q leads to a stronger shrinkage, and $q = 1$ corresponds to a group lasso.

By adding the penalty term (9), we have the following improved tensor regression model:

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^d} f(\mathcal{X}) = h(\mathcal{X}) + \lambda g(\mathcal{X}). \quad (10)$$

After solving this optimization problem, each obtained factor matrix $\mathbf{U}^{(k)}$ has a few common columns whose values are close to zero. These columns can be deleted and the actual rank of our obtained tensor becomes $\hat{R} \le R$, where $\hat{R}$ is the number of remaining columns that are not deleted.

It is non-trivial to minimize $f(\mathcal{X})$ since the regularization function $g(\mathcal{X})$ is usually non-differentiable and non-convex

with respect to $\mathbf{U}^{(k)}$'s. Instead, we solve the following optimization problem in practice:

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^d} \hat{f}(\mathcal{X}) = h(\mathcal{X}) + \lambda \hat{g}(\mathcal{X}). \quad (11)$$

Here $\hat{g}(\mathcal{X})$ is an upper bound of $g(\mathcal{X})$ obtained via the variational inequality [10]:

$$g(\mathcal{X}) \le \hat{g}(\mathcal{X}) = \min_{\boldsymbol{\eta} \in \mathbb{R}^R} \frac{\lambda}{2} \sum_{r=1}^R \frac{\sum_{k=1}^d \|\mathbf{u}_r^{(k)}\|_2^2}{\eta_r} + \frac{\lambda}{2} \|\boldsymbol{\eta}\|_q. \quad (12)$$

Once $\{\mathbf{U}^{(k)}\}_{k=1}^d$ is given, the values of $\hat{g}(\mathcal{X})$ and $\hat{f}(\mathcal{X})$ can be estimated by setting the elements of $\boldsymbol{\eta}$ as

$$\eta_r = (z_r)^{\frac{2}{q+1}} \|\mathbf{z}\|_{q_1}^{q_2}, \quad \forall \, r = 1, \ldots, R, \quad (13)$$

where $z_r = \left(\sum_{k=1}^d \|\mathbf{u}_r^{(k)}\|_2^2 \right)^{\frac{1}{2}}$, $q_1 = \frac{2q}{q+1}$ and $q_2 = \frac{q-1}{q+1}$.

Problem (11) can be solved via an alternating algorithm such as a block coordinate descent solver or alternating direction method of multipliers. Due to the page limitation, we omit the details in this paper and will explain the detailed optimization algorithm in an extended journal paper.

C. Adaptive Sampling Strategy

Another fundamental question is how to select the parameter samples $\{\boldsymbol{\xi}_n\}_{n=1}^N$ for simulation. The method in [7] uses some Monte Carlo random samples. Instead, this paper reduces the simulation cost by selecting only a few informative samples for detailed device- or circuit-level simulations.

We first use the Latin Hybercube (LH) sampling to generate an initial sample set Ω. Then we employ an exploration step via the Voronoi diagram to measure the sample density in Ω. Given two distinct samples $\boldsymbol{\xi}_i, \boldsymbol{\xi}_j \in \Omega$, a Voronoi cell $C_i(\boldsymbol{\xi}_i)$ covers the region that are closest to $\boldsymbol{\xi}_i$. It is defined as the intersection of a set of half-planes (hp):

$$C_i(\boldsymbol{\xi}_i) = \bigcap_{\boldsymbol{\xi}_j \in \Omega \backslash \boldsymbol{\xi}_i} \mathrm{hp}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) \quad (14)$$

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

It is intractable to calculate an Voronoi cell exactly in a high-dimensional space. However, we can easily estimate it via Monte Carlo [11]. The sample density of C_i is approximated by counting the number of samples that are closest to $\boldsymbol{\xi}_i$. Each sample in Ω determines one Voronoi cell with itself as the center, and we can select a new sample from the cell region with the lowest density.

If the performance metric $y(\boldsymbol{\xi})$ is known to be highly nonlinear, we can further exploit its non-linearity. Given $\boldsymbol{\xi}$ and a Voronoi cell center $\mathbf{a}$, we measure the nonlineary of $y(\boldsymbol{\xi})$ as

$$\gamma(\boldsymbol{\xi}) = |\hat{y}(\boldsymbol{\xi}) - \hat{y}(\mathbf{a}) - \nabla \hat{y}(\mathbf{a})^T (\boldsymbol{\xi} - \mathbf{a})|. \quad (15)$$

We select a new sample as the one with largest $\gamma(\boldsymbol{\xi})$ in a Voronoi cell with the lowest sample density. This method can be easily extended to a batch version by searching the top-K least-sampled regions.
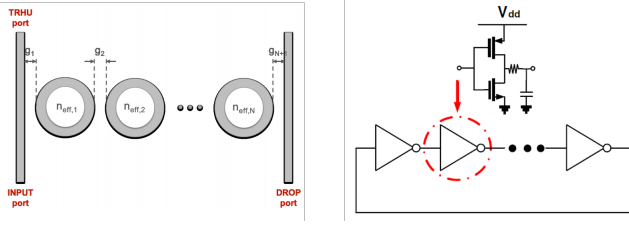
Fig. 1. Left: a photonic band-pass filter with 9 micro-ring resonators. Right: Schematic of a CMOS ring oscillator.
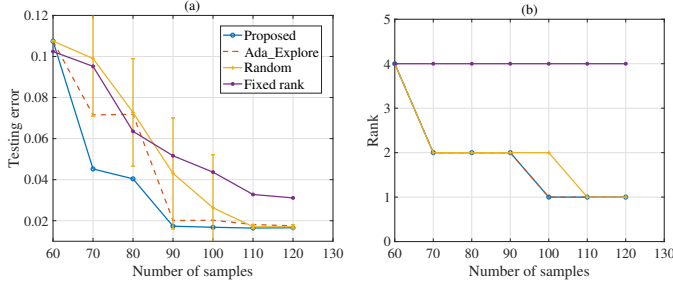

Fig. 2. Result of the photonic filter. (a) Testing error on 10^5 MC samples, achieving 98.3% accuracy. (b) The estimated tensor ranks.

Fig. 3. Results of the CMOS ring oscillator. (a) Testing error on 10^5 MC samples (b) Probability density functions of the oscillator frequency.

TABLE I
COMPARISON WITH THE GPC MODEL WITH A TOTAL DEGREE SCHEME.

	Proposed	Total-degree gPC	MC
# of variables	855	1711	N/A
# of samples	290	1711	10^5
Mean	106.28	106.58	106.53
Stdvar	4.616	6.810	4.641
Error	1%	4.84%	N/A

IV. NUMERICAL EXPERIMENT

We verify our algorithm by a photonic band-pass filter and a CMOS ring oscillator as shown in Fig. 1.

Photonic band-pass filter: This benchmark has 19 Gaussian random parameters describing the variations of the effective phase index (n_{neff}) of each ring, and the gap (g) between adjacent rings and between the end ring and the bus waveguides. We set the highest polynomial order $p = 2$ for each dimension, and initialize $\mathcal{X}$ as rank-4, and use $q = 0.5$ in the regularizer. We initialize 60 samples with a standard LH experimental design, and further select 6 batches with 60 additional samples. The proposed tensor regression framework is compared with a fixed rank method, a random and an adaptive exploration sampling method (shown in Fig. 2).

CMOS ring oscillator: This circuit has 57 Gaussian random parameters describing the variations of threshold voltages, gate-oxide thickness, and effective gate length/width. The simulation results are obtained by calling a periodic steady-state simulator repeatedly. We set basis as order-2 in each dimension, initialize $\mathcal{X}$ as rank-5, and use $q = 0.5$ in the regularizer. We start from 140 standard LH samples, and adaptively select additional 7 batches with 210 samples. The results and comparison are summarized in Fig. 3 and Table I.

V. CONCLUSION

This paper has proposed a tensor regression framework for high-dimensional uncertainty quantification. Our method has addressed two fundamental challenges: automatic tensor rank determination and adaptive sampling. The numerical result has demonstrated the excellent capability of automatic rank determination of our method, and the simulation cost reduction by our adaptive sampling method.
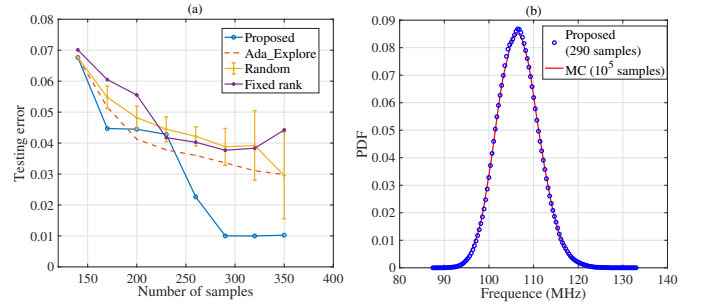
REFERENCES

[1] D. Xiu, *Numerical methods for stochastic computations: a spectral method approach*. Princeton university press, 2010.

[2] 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. CAD Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1533–1545, 2013.

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

[4] D. V. Ginste, D. De Zutter, D. Deschrijver, T. Dhaene, P. Manfredi, and F. Canavero, "Stochastic modeling-based variability analysis of on-chip interconnects," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 2, no. 7, pp. 1182–1192, 2012.

[5] 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.

[6] 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. CAD Integr. Circ. Syst.*, vol. 34, no. 1, pp. 63–76, 2015.

[7] 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.

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

[9] X. Shi, H. Yan, Q. Huang, J. Zhang, L. Shi, and L. He, "Metamodel based high-dimensional yield analysis using low-rank tensor approximation," in *Proc. DAC*, 2019, pp. 1–6.

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

[11] K. Crombecq, I. Couckuyt, D. Gorissen, and T. Dhaene, "Space-filling sequential design strategies for adaptive surrogate modelling," in *Int. Conf. Soft Comput. Techn. Civil, Struct. Environ. Eng.*, 2009, pp. 1–20.