University of California
Santa Barbara

Uncertainty-Aware and Efficient Design Automation from Classical Circuit to Quantum Computing

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Zichang He

Committee in charge:

Professor Zheng Zhang, Chair
Professor Kerem Camsari
Professor Joao Hespanha
Professor Luca Daniel

September 2023

The Dissertation of Zichang He is approved.

Professor Kerem Camsari

Professor Joao Hespanha

Professor Luca Daniel

Professor Zheng Zhang, Committee Chair

August 2023

Uncertainty-Aware and Efficient Design Automation from Classical Circuit to Quantum Computing

by

Zichang He

To my parents

Acknowledgements

I want to express my deepest gratitude to everyone who contributed to the successful completion of this dissertation. I apologize that not all of them can be listed here, but each of you has my heartfelt thanks.

Foremost, I owe a debt of gratitude to my advisor, Prof. Zheng Zhang for his invaluable guidance, patience, and unwavering support throughout this journey. Prof. Zhang introduced me to the field of design automation and gave me all the courage to explore the quantum computing field. His mentorship has indelibly shaped both my academic career and personal growth. I would also like to thank the members of my dissertation committee, Prof. Kerem Camsari, Prof. Joao Hespanha, and Prof. Luca Daniel, for their insightful suggestions and constructive comments. Their perspectives have deepened my understanding of the research and its broader impact.

My heartfelt thanks go to all the collaborators in academia and industry, without whom this research would not have been possible. In my early Ph.D. career, I was fortunate to have Dr. Chunfeng Cui, who patiently taught me everything from scratch. The collaborations and fruitful discussions with Dr. Weilong Cui, Prof. Timothy Sherwood, Prof. Bo Zhao, Mr. Ryan Solgi, Mr. Zhengqi Gao, Mr. Zhiding Liang, Mr. Zhixin Song, and Mr. Fan Liu also greatly broadened my research topics and strengthened my knowledge. Mr. Yifan Pan and Mr. Nanlin Guo gave me rewarding and valuable experience working with relatively junior researchers. When I started learning quantum computing, I was fortunate to work with Dr. Yuri Alexeev, Mr. Danylo Lykov, Mr. Cameron Ibrahim, Prof. Ilya Safro, Dr. Salvatore Mandrà, Dr. Sahil Gulania, Dr. Bo Peng, Dr. Niri Govind, Dr. Ruslan Shaydulin, Dr. Shouvanik Chakrabarti, Mr. Dylan Herman, Dr. Changhao Li, Dr. Yue Sun, Dr. Marco Pistoia, and all the colleagues from the JPMorgan Chase team.

I extend my thanks to all the labmates and friends along the way, including Zhuotong Chen, Cole Hawkins, Kaiqi Zhang, Jianyu Xu, Zi Yang, Xinling Yu, Yifan Yang, Ziyue Liu, Xinyu Chen, Junnan Shan, Rahul Singh, Hong Wang, Shiyang Li, Leilao Shao, Yuyang Wang, Hanbin Hu, Haoyang Zheng, Zhiming Huang, Yubo Huang, Yuqing Yang, Moxian Song, Ying Cao and many more.

Special thanks to my undergraduate advisors, Prof. Yong Deng and Prof. Wen Jiang, for introducing me to the academic world and significantly shaping my life.

Finally, I cannot end without thanking my family and my partner Jianing Chen, who provided emotional support and accompany throughout this challenging period.

Curriculum Vitæ

Zichang He

Education

2023	Ph.D. in Electrical and Computer Engineering, University of California, Santa Barbara.
2021	M.S. in Electrical and Computer Engineering, University of California, Santa Barbara.
2018	B.Eng. in Detection, Guidance and Control Technology, Northwestern Polytechnical University.

Publications

- **Z. He**, B. Peng, Y. Alexeev, and Z. Zhang, "Distributionally Robust Variational Quantum Algorithms with Shifted Noise", arXiv:2308.14935
- **Z. He**, Z. Zhang, and S. Mandra "One the Variational Method for QAOA Simulation", *in preparation*
- Y. Pan*, **Z. He***, N. Guo, and Z. Zhang, "Distributionally Robust Circuit Design Optimization under Variation Shifts," *International Conf. Computer Aided Design (ICCAD)*, San Francisco, CA, Oct. 2023 (* Equal contributions)
- **Z. He**, R. Shaydulin, S. Chakrabarti, D. Herman, C. Li, Y. Sun, M. Pistoia, "Alignment between Initial State and Mixer Improves QAOA Performance for Constrained Portfolio Optimization", arXiv: 2305.03857
- Z. Liang, Z. Song, J. Cheng, **Z. He**, J. Liu, H. Wang, R. Qin, Y. Wang, S. Han, X. Qian, and Y. Shi, "Hybrid Gate-Pulse Model for Variational Quantum Algorithms," *ACM/IEEE Design Automation Conference (DAC)*, San Francisco, CA, Jul. 2023.
- S. Gulania*, **Z. He***, B. Peng*, N. Govind and Y. Alexeev, "QuYBE - An Algebraic Compiler for Quantum Circuit Compression," *ACM/IEEE Symposium on Edge Computing (SEC) Workshop on Quantum Computing.* (* Equal contributions)
- C. Ibrahim, D. Lykov, **Z. He**, Y. Alexeev, I. Safro, "Constructing Optimal Contraction Trees for Tensor Network Quantum Circuit Simulation," *IEEE High Performance Extreme Computing Conference (HPEC)*, 2022. (Best Student Papar award)
- **Z. He** and Z. Zhang, "PoBO: A Polynomial Bounding Method for Chance-Constrained Yield-Aware Optimization of Photonic ICs," accepted by *IEEE Trans. Computer-Aided Design for Integrated Circuits and Systems (TCAD)*, vol. 41, no. 11, pp. 4915-4926, Nov. 2022.
- **Z. He**, B. Zhao and Z. Zhang, "Active Sampling for Accelerated MRI with Low-Rank Tensors," *the Annual International Conference on the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2022.

- R. Solgi, **Z. He**, W. J. Liang and Z. Zhang, "Evolutionary Tensor Shape Search for Optimum Data Compression with Tensor Train Decomposition," arXiv: 2205.10651
- **Z. He** and Z. Zhang, "Progress of Tensor-based high-Dimensional Uncertainty Quantification of Process Variations," *International Applied Computational Electromagnetics Society (ACES) Symposium*, 3 pages, Aug. 2021. (Invited paper)
- **Z. He** and Z. Zhang, "High-dimensional Uncertainty Quantification via Rank- and Sample-Adaptive Tensor Regression," *IEEE Trans. Components, Packaging and Manufacturing Technology (T-CMPT)*, vol. 11, no. 9, pp. 1317-1328, Sept. 2021. (Invited paper)
- **Z. He** and Z. Zhang, "High-Dimensional Uncertainty Quantification via Active and Rank-Adaptive Tensor Regression," *IEEE Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 3 pages, San Jose, CA, Oct. 2020. (Best Student Paper Award)
- **Z. He**, W. Cui, C. Cui, T. Sherwood and Z. Zhang, "Efficient Uncertainty Modeling for System Design via Mixed Integer Programming," *International Conf. Computer Aided Design (ICCAD)*, 8 pages, Westminster, CO, Nov. 2019.

Professional Experience

2022 Summer	Research associate at JPMorgan Chase
2021 Fall	Research intern at KBR with NASA Quantum AI Lab
2021 Summer	Visiting researcher at Argonne National Lab

Awards

2022	Best student paper award at IEEE HPEC Conference
2021	DAC Young Student Fellow, Best Research Video Award
2021	IEE Excellent in Research Fellowship
2021,2020	Outstanding Teaching Assistant award in the department of ECE, UCSB
2020	Best student paper award at IEEE EPEPS Conference
2018	Graduate Fellowship in the department of ECE, UCSB

Abstract

Uncertainty-Aware and Efficient Design Automation from Classical Circuit to Quantum Computing

by

Zichang He

In the twilight of Moore's Law, the semiconductor industry grapples with escalating challenges driven by growing process variations and inherent uncertainties in the manufacturing of increasingly miniaturized devices. Traditional electronic design automation (EDA) tools attempt to address this through rigorous simulation, modeling, and optimization. Concurrently, the need for alternative computational paradigms, notably quantum computing, has surged, introducing fresh complexities and challenges. This dissertation primarily addresses the high cost and low efficiency of modeling and simulation, and the difficulties in design optimization under conditions of noisy and expensive simulations in the context of classical circuits and quantum computing. It proposes uncertainty-aware and efficient design automation methodologies in the EDA field and quantum computing.

The first part of this dissertation focuses on the circuit-level uncertainty quantification and optimization of classical circuits. We first propose a tensor regression model as the stochastic circuit performance model under the high-dimensional process variations. Then we propose a yield-aware stochastic programming approach to optimize the circuit design, balancing the circuit yield and performance. To tackle optimization in the face of shifted variations, we introduce a distributionally robust optimization formulation, which can be efficiently solved using a Bayesian optimization solver.

The second part addresses block-level simulation and optimization of quantum algo-

rithms. Highlighting the difficulty of the simulation, we begin with an exploration of a variational method employed in the classical simulation of quantum algorithms. Given the substantial obstacle noise poses to the practical applications of quantum computing, we extend the idea of distributionally robust optimization to optimize the parameters of variational quantum algorithms, aiming to enhance the reliability of a quantum algorithm under real-time noise.

Contents

Chapter 1

Introduction

1.1 Background and Motivation

In the era dominated by Moore's Law, the semiconductor industry experienced exponential growth in computational power [1]. However, as we approach the physical limits of semiconductor technologies, we find ourselves not only facing a slowdown in this progress, but also dealing with increasing levels of process variation and uncertainties in the manufacturing of these miniaturized devices [2, 3]. The smaller the devices become, the more significant the impact of these variations. This variability can arise due to several factors, including imperfections in the manufacturing process, environmental conditions, changes over time, etc [4]. Such process variations could significantly impact the performance, power consumption, and reliability of circuits [4–6].

In the conventional realm of electronic design automation (EDA), handling these uncertainties has become an integral part of the design process. Simulation, modeling, and optimization tools have been essential in addressing this issue [7–10]. They enable designers to predict and understand the variations in system behavior, and to optimize design trade-offs under these uncertain conditions. This reduces the time, cost, and risk

associated with the development of new hardware while ensuring the robustness and reliability of the final product.

As we are approaching the end of Moore's era [11], the scientific community has been motivated to explore new computational paradigms, such as quantum computing [12], probabilistic computing [13], optical computing [14], etc. Among them, quantum computing, which holds the promise of significant computational speedups for specific problems, has been heavily investigated in recent years [12,15]. However, with quantum computing, we face an entirely new set of challenges. The quantum circuit is, in principle, hard to simulate in a classical computer due to its exponentially growing memory requirements. Additionally, a quantum system is highly sensitive to noise, introducing complexity and challenges to the design optimization process. Although quantum computing has its unique challenges, we hope to borrow the lessons from EDA to this emerging computing framework.

As quantum computing continues to mature and the scale of quantum systems increases, there is an urgent need for robust tools for the simulation, modeling, and optimization of quantum circuits and systems. This emerging field of quantum design automation aims to bridge this gap [16–19]. As quantum computing continues to evolve, its design automation is becoming increasingly crucial in navigating through these uncertainties and enabling the design of scalable and reliable quantum computers.

1.2 Overview and Contributions

1.2.1 Abstraction of Design Level

In an integrated circuit or a computer hardware design, there are usually four levels of design: device level, circuit level, block level, and system level. Each level represents a

specific level of abstraction and provides a particular perspective on the design process.

- Device level. The device level is the lowest level of abstraction. The device level corresponds to the design of individual basic elements that form circuits, where designers are concerned with the properties of individual devices

- Circuit level. The circuit level involves the combination of devices into functional units to perform specific tasks. Designers must deal with the interactions between different devices, such as timing, power consumption, and how individual device variations might affect the entire circuit's performance.

- Block level. The block level combines circuits to create larger and more complex functional units, often called "blocks" or "modules". A block-level design needs to consider data flow, control signals, and interface compatibility between blocks.

- System level. At the highest level of abstraction, the system level involves integrating the different blocks to create the full system. A system-level design involves defining the architecture of a system, specifying how blocks interact with each other, and optimizing the system to meet requirements such as performance, power, cost, and reliability.

In a classical electronics system, the example of the above four-level design could be as follows. The device level corresponds to the design of individual transistors, their characteristics, and their physical implementation on a silicon wafer. The circuit level can include the design of analog circuits such as amplifiers, filters, and power circuits, or digital circuits like logic gates, flip-flops, and multiplexers. The block level can include a processor's arithmetic logic unit, memory units, and I/O interfaces. The system level could be the final chip architecture design.

Sharing different computation mechanisms, quantum computing uses quantum phenomena such as superposition and entanglement to perform computation. However, one can similarly draw a parallel between the four levels of design in classical computing and their counterparts in quantum computing. The device level refers to the creation and control of the individual quantum bits or qubits, which are the basic units of quantum information. At this level, the focus is on the physical realization of the qubits. The popular realization platform includes superconducting, photonics, trapped ions, etc. At the circuit level, individual qubits are manipulated and interact with each other to perform quantum computation. This includes the design of quantum logic gates. At the block level, a "block" refers to a group of qubits and the associated quantum circuits that perform a specific function. Most quantum algorithms are developed and implemented at this level. The system level encompasses the entire quantum computer system. Although a current quantum system is much less mature compared to a classical electronics one, a good example could be the quantum processors developed by the big companies, such as Google's Sycamore [20], IBM's a series of quantum processors [21], Quantinuum's system models [22].

The different abstraction levels in both classical electronic and quantum systems are visualized in Fig. 1.1.

1.2.2 Part I: Classical Circuit at the Circuit Level

In the first part of this dissertation, we focus on the uncertainty-aware simulation and optimization of classical circuits at the circuit level.

In classical non-scale circuit design, process variations have long been a challenge. Due to limited knowledge about underlying physics and immature manufacturing technology, this challenge is exacerbated in emerging computing approaches, such as phonics circuits.

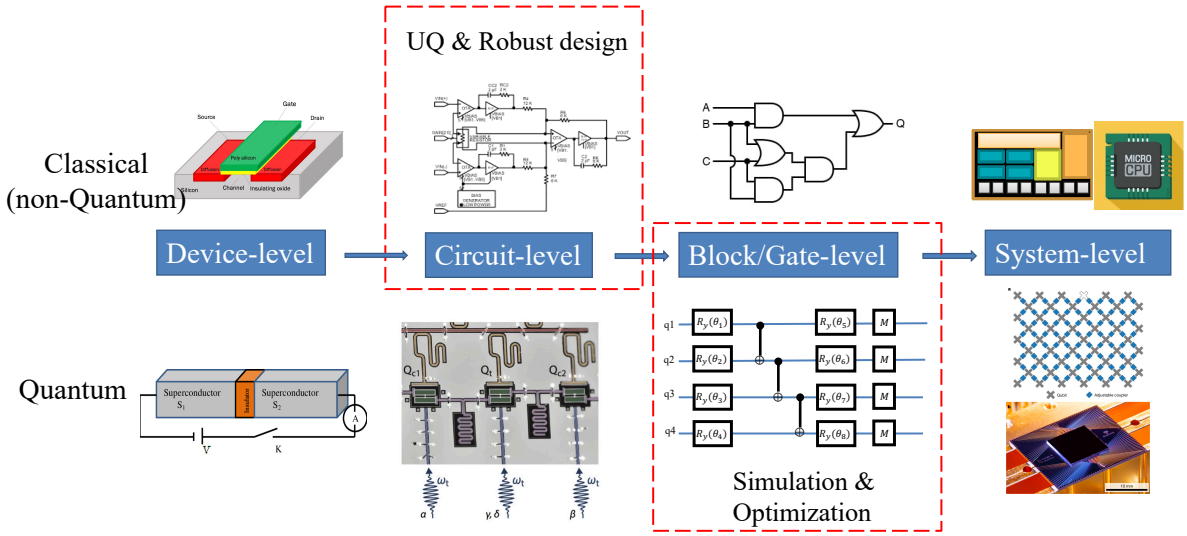

Figure 1.1: Abstraction of different design levels in both conventional and quantum computing. The two highlighted blocks are the topics mainly covered in this dissertation, i.e., uncertainty quantification and optimization of classical circuits and simulation and optimization of quantum algorithms.

To account for uncertainty in circuit design, several main challenges arise: (1) modeling stochastic circuits becomes increasingly difficult when process variations have a high dimension [23, 24], and (2) optimizing performance and yield becomes challenging under uncertainty-aware optimization [25]. (3) very often, the knowledge of reference variation is imperfect, and the variation could even shift over time [26, 27].

Firstly, we propose a tensor-regression-based generalized polynomial chaos expansion model to construct a surrogate model that quantifies the uncertainty of a stochastic circuit and enables fast stochastic simulation [28, 29]. The low-rank structure mitigates the typical curse of dimensionality challenge. Specifically, the model is capable of determining the tensor rank automatically and selecting the simulation samples adaptively.

To address over-conservative design in optimizing yield under uncertainty, a chance-constrained programming formulation is proposed for uncertainty-aware circuit design [30]. A polynomial bounding method is proposed to simplify but strictly bound the stochastic

yield constraints, making the problem tractable.

To deal with the imperfect reference variation model, we introduce a shift-aware formulation [31], distributionally robust optimization, to the EDA community. We assume the variation distribution is unknown and describe it as an uncertainty ball. The design optimization becomes a two-level min-max problem where the design is optimized under the worst-case distribution.

1.2.3 Part II: Quantum Computing at the Block Level

In the second part of this dissertation, we focus on the simulation and optimization of quantum computing at the block level, which focuses on the development of quantum algorithms.

Quantum computing is a challenge for classical computers to simulate, even for noiseless quantum circuits and algorithms, due to exponential memory scaling [32,33]. In contrast to many "training-free" simulators (e.g., state vector, tensor network), we explore a machine-learning-based approach for the efficient simulation of variational quantum algorithms [34]. The machine learning model represents a quantum state in the Hibert space using significantly fewer parameters and mimics state evolution by variational updates of the model parameters. We systematically analyze the advantages and disadvantages of the ML model for variational quantum algorithm simulations [35].

Beyond simulation, to design quantum circuits for practical applications, noise remains a significant obstacle for near-term quantum devices [36, 37]. Due to the inherent mechanism of quantum computing, quantum circuits are highly sensitive to noise, which can reduce the robustness and reliability of quantum solutions. To address the challenges posed by the real-time changing noise of real quantum devices, we propose a distributionally robust optimization formulation and solve it via Bayesian optimization [38]. The

proposed formulation is applied to the parameter optimization of variational quantum algorithms, and we observe significant performance improvements under shifted noise scenarios.

Similar to how electronic design automation has driven the significant progress of modern computers, we hope the techniques and ideas presented in this part and future research can accelerate the development of quantum computing.

1.3 Outline

The dissertation is organized as follows. We discuss the uncertainty quantification and optimization of classical circuits in chapters 2, 3, and 4. In Chapter 2, we introduce a tensor regression model [28,29] for high-dimensional uncertainty quantification of process variations. In Chapter 3, we introduce a chance-constrained programming approach for yield-aware optimization problem [30]. The yield-aware optimization formulation can effectively balance the design yield and performance, avoiding over-conservative design in yield optimization. In Chapter 4, we introduce a distributionally robust optimization for circuits under shifted variations [31]. For the first time, we propose a shift-aware formulation to optimize a circuit when the reference variation information is not precise.

Then, we introduce some quantum computing background in Chapter 5. We discuss the simulation and optimization of quantum computing algorithms in Chapter 6 and 7, respectively. In Chapter 6, we investigate a machine-learning-assisted variational method for simulating quantum algorithms [35]. We comprehensively study the performance and limitations of the variational simulation approach. In Chapter 7, we study the parameter optimization of variational quantum algorithms [38]. Specifically, boosting the reliability of quantum algorithms, we tackle algorithm optimization under real-time quantum noise. The dissertation is concluded and discussed in Chapter 8.

Part I

Uncertainty Quantification and Optimization of Classical Circuits

Chapter 2

High-Dimensional Uncertainty Quantification of Process Variations

2.1 Introduction

In this chapter, we will highlight the challenges of uncertainty quantification of high-dimensional process variations and address them using a tensor-regression-based approach.

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 [4]. Monte Carlo (MC) is one of the most popular methods to quantify chip performance under uncertainty, but it requires a huge amount of computational cost [39]. Instead, stochastic spectral methods based on generalized polynomial chaos (gPC) [40] offer efficient solutions for fast uncertainty quantification by approximating a real uncertain circuit variable as a linear combination of some stochastic basis functions [41–43]. These techniques have been increasingly used in design automation [44–51]. 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 [52,53], hyperbolic regression [54], analysis of variance (ANOVA) [55,56], model order reduction [57], and hierarchical modeling [24, 58], and tensor methods [23, 24].

The low-rank tensor approximation has shown promising performance in solving high-dimensional uncertainty quantification problems [23, 59–64]. 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* [65], existing methods often use a tensor rank pre-specified by the user or use a greedy method to update the tensor rank until convergence [23, 66, 67]. 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.

Main 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 [68–70]. There are some existing works of automatic rank determination [71–73] and adaptive sampling [74, 75] for tensor decomposition and completion. The Bayesian frameworks [76, 77] 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 deter-

mination and adaptive sampling. The main contributions of this chapter include:

- We formulate high-dimensional uncertainty quantification as a tensor regression problem. We 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 by combining the estimation of Voronoi 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 capture the high-dimensional stochastic output well with only 100-600 samples.

2.2 Notation and Preliminaries

Throughout this chapter, 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.

2.2.1 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 [40]:

$$y(\boldsymbol{\xi}) \approx \hat{y}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha} \in \Theta} c_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \tag{2.1}$$

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.,

$$\Theta = \{\boldsymbol{\alpha} | \alpha_k \in \mathbb{N}, 0 \leq \sum_{k=1}^{d} \alpha_k \leq p\}, \tag{2.2}$$

leading to a total of $\frac{(d+p)!}{d!p!}$ terms of expansion. Let $\phi_{\alpha_k}^{(k)}(\xi_k)$ 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 orthonormal polynomial basis:

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \prod_{k=1}^{d} \phi_{\alpha_k}^{(k)}(\xi_k). \tag{2.3}$$

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

$$\langle \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi}) \rangle = \int_{\mathbb{R}^d} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) \Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi} = \delta_{\boldsymbol{\alpha},\boldsymbol{\beta}}. \tag{2.4}$$

The detailed formulation and construction of univariate basis functions can be found in [40, 78].

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 [79] and stochastic testing [42]) and non-intrusive (i.e., sampling) methods (e.g., stochastic collocation based on pseudo-projection or regression [80]). It is well known that gPC expansion suffers the curse of dimensionality. The computational cost grows exponentially as the dimension of $\boldsymbol{\xi}$ increases.

2.2.2 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:

$$\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i_1 \cdots i_d} x_{i_1 \cdots i_d} y_{i_1 \cdots i_d}. \tag{2.5}$$

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 of some rank-1 vectors, which is called CANDECOMP/PARAFAC (CP) decomposition [81]:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \cdots \circ \mathbf{a}_r^{(d)} = [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(d)}]\!], \tag{2.6}$$

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

$$\begin{aligned} \mathbf{X}_{(k)} =& \mathbf{A}^{(k)} \mathbf{A}^{(\backslash k)^T} \text{ with} \\ \mathbf{A}^{(\backslash k)} =& \mathbf{A}^{(d)} \odot \cdots \odot \mathbf{A}^{(k-1)} \odot \mathbf{A}^{(k+1)} \cdots \odot \mathbf{A}^{(1)}, \end{aligned} \tag{2.7}$$

where $\odot$ denotes the Khatri-Rao product, which performs column-wise Kronecker products [81]. More details of tensor operations can be found in [81].

2.3 Proposed Tensor Regression Method

2.3.1 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:

$$\Theta = \{\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_d] \mid 0 \leq \alpha_k \leq p, \forall k \in [1, d]\}. \tag{2.8}$$

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

$$x_{i_1 i_2 \cdots i_d} = c_{\boldsymbol{\alpha}} \text{ and } b_{i_1 i_2 \cdots i_d}(\boldsymbol{\xi}) = \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{2.9}$$

Combining Eqs (2.1), (2.8) and (2.9), the truncated gPC expansion can be written as a tensor inner product

$$y(\boldsymbol{\xi}) \approx \hat{y}(\boldsymbol{\xi}) = \langle \mathcal{X}, \mathcal{B}(\boldsymbol{\xi}) \rangle. \tag{2.10}$$

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:

$$\mathcal{B}(\boldsymbol{\xi}) = \boldsymbol{\phi}^{(1)}(\xi_1) \circ \boldsymbol{\phi}^{(2)}(\xi_2) \circ \cdots \circ \boldsymbol{\phi}^{(d)}(\xi_d), \tag{2.11}$$

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:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(d)} = [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(d)}]\!]. \tag{2.12}$$

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

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^d} h(\mathcal{X}) = \frac{1}{2}\sum_{n=1}^{N} \left(y_n - \langle [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(d)}]\!], \mathcal{B}^n \rangle\right)^2, \tag{2.13}$$

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

2.3.2 Automatic Rank Determination

The low-rank approximation (2.12) 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

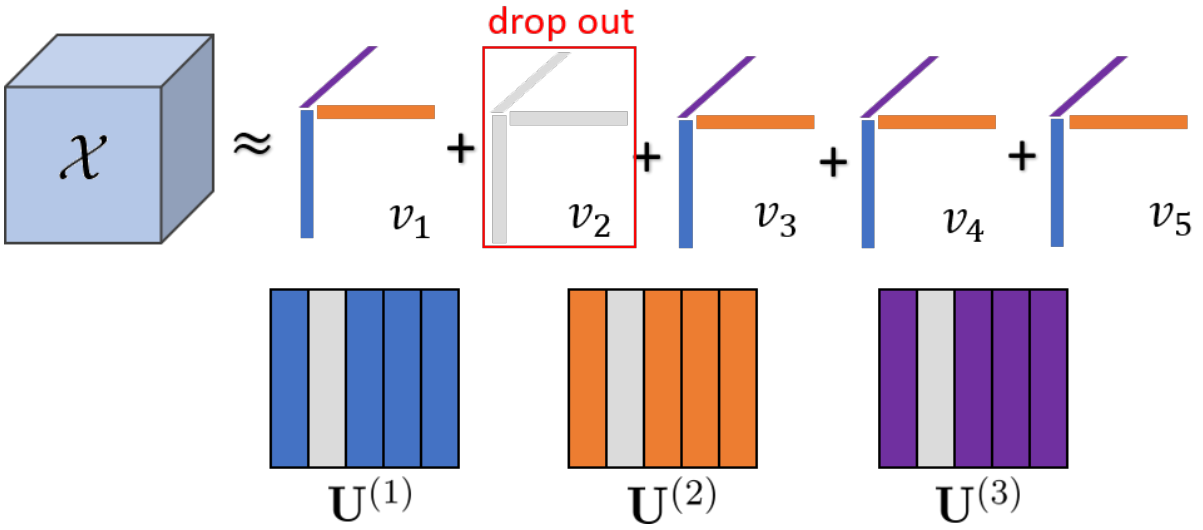

Figure 2.1: Visualization of the tensor rank determination. Here the gray vectors denote some shrinking tensor factors that can be removed from a CP decomposition.

estimation. Specifically, define the following vector:

$$\mathbf{v} := [v_1, v_2, \cdots, v_R] \text{ with } v_r = \left(\sum_{k=1}^{d} \|\mathbf{u}_r^{(k)}\|_2^2 \right)^{\frac{1}{2}} \forall r \in [1, R]. \tag{2.14}$$

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

$$g(\mathcal{X}) = \|\mathbf{v}\|_q, \quad q \in (0, 1]\,. \tag{2.15}$$

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 coeffi-

cients by solving a regularized tensor regression problem:

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^{d}} f(\mathcal{X}) = h(\mathcal{X}) + \lambda g(\mathcal{X}), \tag{2.16}$$

where $\lambda > 0$ is a regularization parameter. As shown in Fig. 2.1, 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.

2.3.3 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 [82]) *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*

$$\|\mathbf{y}\|_\alpha = \min_{\boldsymbol{\eta} \in \mathbb{R}_+^p} \frac{1}{2} \sum_{r=1}^{p} \frac{y_r^2}{\eta_r} + \frac{1}{2} \|\boldsymbol{\eta}\|_\beta, \tag{2.17}$$

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.1.1. ■

If we take $p = R$, $\alpha = q$, and $y_r = v_r$ (defined in (2.14)), on the right-hand side of Eq. (2.17), then we have

$$\hat{g}(\mathcal{X}, \boldsymbol{\eta}) = \frac{1}{2} \sum_{r=1}^{R} \frac{v_r^2}{\eta_r} + \frac{1}{2} \|\boldsymbol{\eta}\|_{\frac{q}{2-q}}. \tag{2.18}$$

The original rank-shrinking function (2.15) is equivalent to

$$g(\mathcal{X}) = \min_{\boldsymbol{\eta} \in \mathbb{R}_+^R} \hat{g}(\mathcal{X}, \boldsymbol{\eta}). \tag{2.19}$$

As a result, we solve the following optimization problem as an alternative to (2.16):

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^d, \boldsymbol{\eta}} \hat{f}(\mathcal{X}) = h(\mathcal{X}) + \lambda \hat{g}(\mathcal{X}, \boldsymbol{\eta}). \tag{2.20}$$

2.3.4 A Block Coordinate Descent Solver for Problem (2.20)

Now we present an alternating minimization solver for Problem (2.20). Specifically, we decompose Problem (2.20) 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:

$$\text{vec}(\mathbf{U}^{(k)}) = (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \tilde{\boldsymbol{\Lambda}})^{-1} \boldsymbol{\Phi}^T \boldsymbol{y}, \tag{2.21}$$

  where $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1, \cdots, \boldsymbol{\Phi}_N]^T \in \mathbb{R}^{N \times R(p+1)}$ with rows $\boldsymbol{\Phi}_n^T = \text{vec}\left(\mathbf{B}_{(k)}^n \mathbf{U}^{(\backslash k)}\right)^T$ for any $n \in [1, N]$, $\tilde{\boldsymbol{\Lambda}} = \text{diag}(\frac{1}{\eta_1}, \ldots, \frac{1}{\eta_R}) \otimes \mathbf{I} \in \mathbb{R}^{R(p+1) \times R(p+1)}$, 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. (2.7), 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. (2.21) to Appendix A.1.2.

- **$\boldsymbol{\eta}$-subproblem:** Suppose that $\{\mathbf{U}^{(k)}\}_{k=1}^d$ are fixed, the formulation of the $\boldsymbol{\eta}$-

subproblem is shown in (2.19). According to lemma 1, we update $\boldsymbol{\eta}$ as

$$\eta_r = (v_r)^{2-q}\|\mathbf{v}\|_q^{q-1} + \epsilon, \tag{2.22}$$

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.

2.3.5 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

$$\begin{aligned} &\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \dots \boldsymbol{\Phi}_N]^T \text{ with} \\ &\boldsymbol{\Phi}_n = \text{vec}(\mathbf{O}_{n,k})^T, \forall n \in [1, N], \\ &\mathbf{O}_{n,k} = \left[\boldsymbol{\phi}^{(k)}(\xi_k^n), \dots, \boldsymbol{\phi}^{(k)}(\xi_k^n)\right] \in \mathbb{R}^{(p+1)\times R}, \end{aligned} \tag{2.23}$$

  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. (2.21) numerically stable. In an adaptive sampling setting (see Section 2.4), we need to solve (2.20) 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. (2.22).

- 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

$$\lambda = \lambda_0 \max(\boldsymbol{\eta}), \tag{2.24}$$

  where λ_0 is chosen via cross validation.

- We stop the block coordinate descent solver for problem (2.20) 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 2.4, is summarized in Algorithm 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

$$\hat{y}(\boldsymbol{\xi}) = \langle \mathcal{X}, \mathcal{B}(\boldsymbol{\xi}) \rangle = \sum_{r=1}^{R} \prod_{k=1}^{d} \left[\boldsymbol{\phi}^{(k)}(\xi_k) \right]^T \mathbf{u}_r^{(k)}. \tag{2.25}$$

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 (2.20) 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 [83].

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. (2.25)

while Adaptive sampling does not stop **do**
  Construct the basis tensor $\mathcal{B}(\boldsymbol{\xi})$
  if No additional samples **then**
    Initialize with Eq. (2.23)
  else
    Initialize $\{\mathbf{U}^{(k)}\}_{k=1}^d$ with the last solution
  end if
  while Tensor regression does not stop **do**
    for $k = 1, 2, \ldots, d$ **do**
      update $\mathbf{U}^{(k)}$ via Eq. (2.21)
    end for
    Update $\boldsymbol{\eta}$ via Eq. (2.22)
    Update regularization parameter λ via Eq. (2.24)
  end while
  Shrink the tensor rank to $\hat{R}$ if possible
  Select new sample pairs based on Algorithm 2
end while

2.4 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) sam-

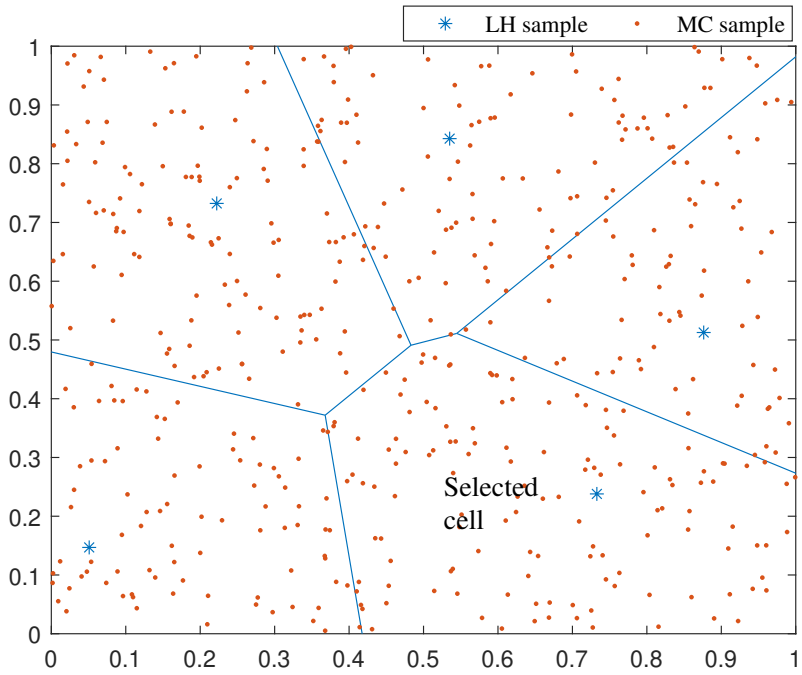

Figure 2.2: An example of Voronoi diagram on $[0,1]^2$. Each LH sample is a Voronoi cell center. The lower right cell should be selected in the first-stage since it has the largest estimated area (volume).

pling method [84]. Specifically, we first generate some standard LH samples $\{\boldsymbol{\zeta}_n^{\mathrm{LH}}\}_{n=1}^N$ 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. Generally, the initial sample size of Θ depends on the number of unknowns in the model. Since problem (2.20) 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.

2.4.1 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 [85]. 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 $\boldsymbol{\xi}_j$

$$\mathrm{hp}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) = \{\boldsymbol{\xi} \in \mathbb{R}^d | \ \|\boldsymbol{\xi} - \boldsymbol{\xi}_i\| \leq \|\boldsymbol{\xi} - \boldsymbol{\xi}_j\|\}. \tag{2.26}$$

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

$$C_i = \bigcap_{\boldsymbol{\xi}_j \in \Omega \setminus \boldsymbol{\xi}_i} \mathrm{hp}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j). \tag{2.27}$$

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.*

An example to show the observation is as below.

Example 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 the 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.*

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. 2.2. 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 [86], to achieve a good estimation accuracy, we use M Monte Carlo samples with $M = 100N$.

2.4.2 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,

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 {ψ_m}_{m=1}^M ∈ [0,1]^d
for m = 1, 2, ..., M do
    Find the closest cell C_n center to ψ_m
    vol(C_n) ← vol(C_n) + 1
end for
Find the cell with the biggest estimated volume vol and the sample set Γ assigned to this cell
Γ_Ω ← Inverse_transform_sampling(Γ)
Calculate the nonlinearity measure γ(Γ_Ω) via Eq. (2.28)
Select ξ* according to Eq. (2.29)
Θ* ← Θ ⋃ {ξ*, y(ξ*)}
```

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 $\mathbf{a} \in \Omega$ [87]. 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

$$\gamma(\boldsymbol{\xi}) = |\hat{y}(\boldsymbol{\xi}) - \hat{y}(\mathbf{a}) - \nabla\hat{y}(\mathbf{a})^T(\boldsymbol{\xi} - \mathbf{a})|. \tag{2.28}$$

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:

$$\boldsymbol{\xi}^\star = \underset{\boldsymbol{\xi}\in\Gamma_\Omega}{\operatorname{argmax}} \left(\gamma\left(\boldsymbol{\xi}\right)\right). \tag{2.29}$$

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 Algorithm 2.

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

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.

2.5 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 $\Psi_{\mathbf{0}}(\boldsymbol{\xi})$:

$$\mu = c_{\mathbf{0}} = x_{11\cdots 1} = \sum_{r=1}^{R} \mathbf{u}_r^{(1)}(1)\mathbf{u}_r^{(2)}(1)\cdots\mathbf{u}_r^{(d)}(1). \tag{2.30}$$

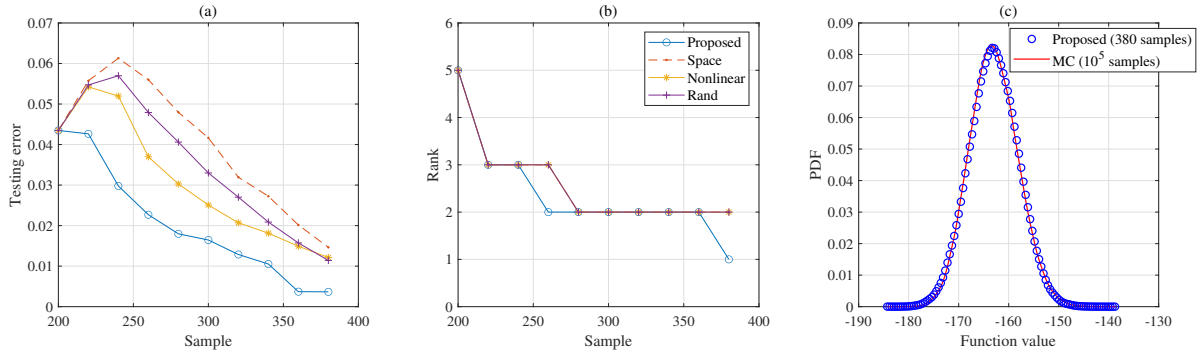

Figure 2.3: Results of approximating the synthetic function. (a) Testing error on 10^5 MC samples. (b) The estimated rank. (c) Probability density functions of the function value.

$x_{11\cdots1}$ 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:

$$\begin{aligned}\sigma^2 &= \sum_{\boldsymbol{\alpha}\in\Theta,\boldsymbol{\alpha}\neq 0} c_{\boldsymbol{\alpha}} = \langle \mathcal{X}, \mathcal{X} \rangle - x_{11\cdots1}^2 \\ &= \sum_{r_1=1}^{R}\sum_{r_2=1}^{R}\prod_{k=1}^{d} \mathbf{u}_{r_1}^{(k)^T}\mathbf{u}_{r_2}^{(k)} - \mu^2.\end{aligned} \tag{2.31}$$

- **Sobol' indices.** Based on the obtained model, we can also extract the Sobol' indices [88,89] for global sensitivity analysis. The main sensitivity index S_j measures the contribution by random parameter ξ_j along to the variance $y(\boldsymbol{\xi})$:

$$S_j = \frac{\mathrm{Var}\left[\mathbb{E}\left[y(\boldsymbol{\xi})|\xi_j\right]\right]}{\sigma^2} \tag{2.32}$$

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

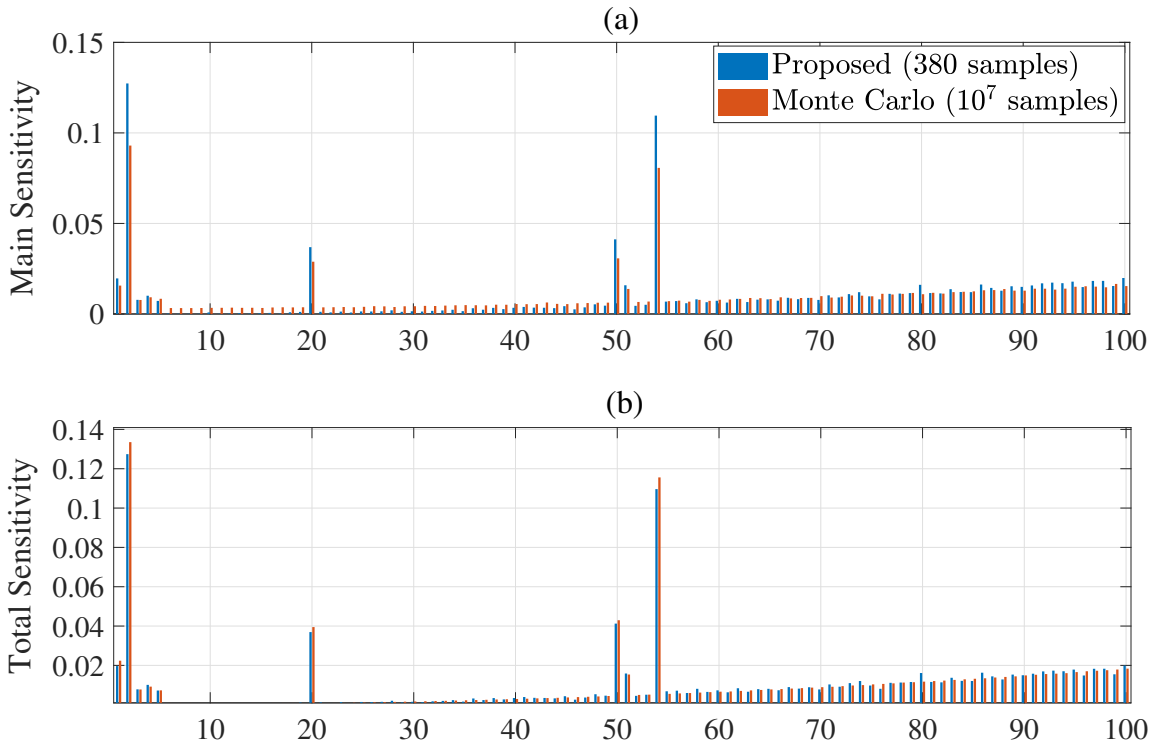

Figure 2.4: Sensitivity analysis of the synthetic function in (2.38). The proposed method fits the results from Monte Carlo [88] with 10^7 simulations very well.

as

$$\begin{aligned}\mathrm{Var}\left[\mathbb{E}\left[y(\boldsymbol{\xi})|\xi_j\right]\right] &= \sum_{i_j=2}^{p+1} x^2_{1\cdots i_j 1\cdots 1} \\ &= \sum_{i_j=2}^{p+1}\left[\sum_{r=1}^{R}\mathbf{u}_r^{(j)}(i_j)\prod_{k\neq j}\mathbf{u}_r^{(k)}(1)\right]^2.\end{aligned} \tag{2.33}$$

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:

$$T_j = 1 - \frac{\mathrm{Var}\left[\mathbb{E}\left[y(\boldsymbol{\xi})|\boldsymbol{\xi}_{\setminus j}\right]\right]}{\sigma^2}. \tag{2.34}$$

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

expectation is estimated as

$$\begin{aligned} &\text{Var}\left[\mathbb{E}\left[y(\boldsymbol{\xi})|\boldsymbol{\xi}_{\setminus j}\right]\right] \\ =&\sum_{(i_1,i_2,\cdots i_d),\ i_j=1} x^2_{i_1\cdots i_d} - x^2_{11\cdots 1} \\ =&\sum_{r_1=1}^{R}\sum_{r_2=1}^{R}\mathbf{u}^{(j)}_{r_1}(1)\mathbf{u}^{(j)}_{r_2}(1)\prod_{k\neq j}\mathbf{u}^{(k)^T}_{r_1}\mathbf{u}^{(k)}_{r_2} - \mu^2. \end{aligned} \tag{2.35}$$

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

2.6 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.

2.6.1 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 2.4.1 (denoted as Space).
- Tensor regression with adaptive sampling based on exploiting nonlinearity only introduced in model with only Section 2.4.2 (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

Table 2.1: 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).

ridge regularization in the regression objective function [70]:

$$\min_{\{\mathbf{U}^{(k)}\}_{k=1}^{d}} f(\mathcal{X}) = h(\mathcal{X}) + \lambda \sum_{k=1}^{d} \|\mathbf{U}^{(k)}\|_{\mathrm{F}}^2. \tag{2.36}$$

The standard ridge regression does not induce a sparse structure. We will keep the tensor rank fixed in solving Eq. (2.36).

- Sparse gPC expansion with a total degree truncation [90] (denoted as Sparse gPC). With the truncation scheme in Eq. (2.2), we compute the gPC coefficients by solving the following problem:

$$\min_{\hat{\boldsymbol{c}}} \frac{1}{2} \sum_{n=1} \left(y_n - \sum_{\boldsymbol{\alpha} \in \Theta} \hat{c}_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_n) \right)^2 + \lambda \|\hat{\boldsymbol{c}}\|_1. \tag{2.37}$$

2.6.2 Synthetic Function (100-dim)

We first consider the following high-dimensional analytical function [91]:

$$y(\boldsymbol{\xi}) = 3 - \frac{5}{d}\sum_{k=1}^{d} k\xi_k + \frac{1}{d}\sum_{k=1}^{d} k\xi_k^3 + \xi_1\xi_2^2 + \xi_2\xi_4 - \xi_3\xi_5 + \xi_{51} + \xi_{50}\xi_{54}^2 + \ln\left(\frac{1}{3d}\sum_{k=1}^{d} k(\xi_k^2 + \xi_k^4)\right) \tag{2.38}$$

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} multi-variate 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. 2.3 (a) shows the relative ℓ_2 testing errors (i.e., $\frac{\|\mathbf{y}(\boldsymbol{\xi})-\hat{\mathbf{y}}(\boldsymbol{\xi})\|_2}{\|\mathbf{y}(\boldsymbol{\xi})\|_2}$) 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. 2.3 (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. 2.3 (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.

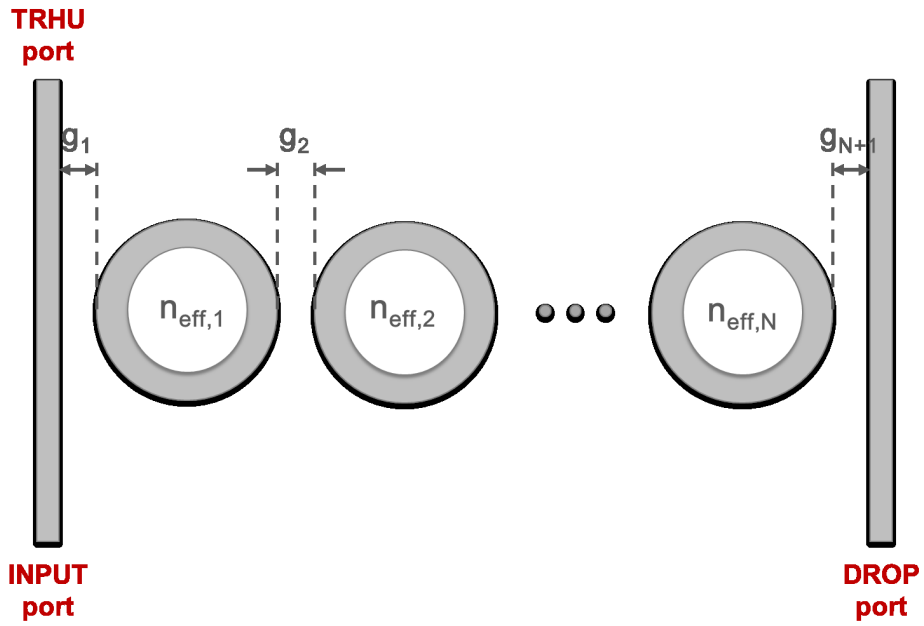

Figure 2.5: Schematic of a band-pass filter with 9 micro-ring resonators.

We compare the complexity and accuracy of different methods in Table 2.1. 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) 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. 2.4 plots the main and total sensitivity metrics from the proposed method and a Monte Carlo estimation [88] 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.

Table 2.2: 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%

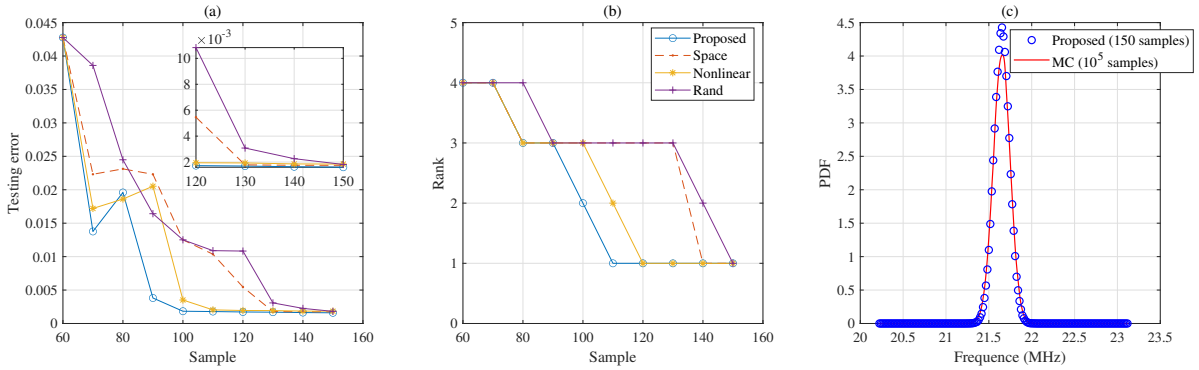

Figure 2.6: Result of the photonic filter. (a) Testing error on 10^5 MC samples. (b) The estimated rank. (c) Probability density functions of the 3-dB bandwidth $f_{3\text{dB}}$ at the DROP port.

2.6.3 Photonic Band-pass Filter (19-dim)

Now we consider the photonic band-pass filter in Fig. 2.5. 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 $q = 0.5$

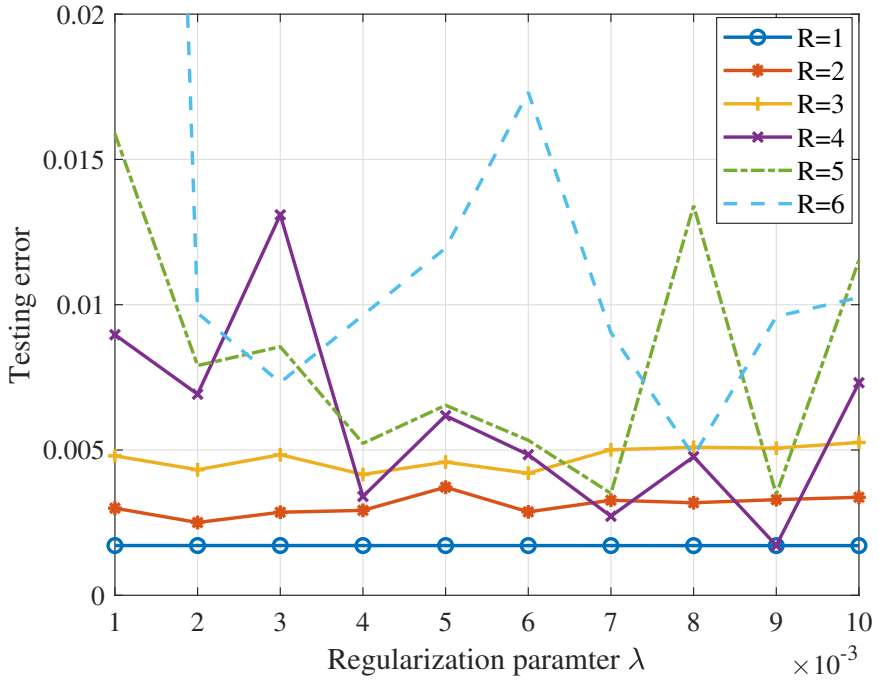

Figure 2.7: One-shot approximations for the photonic band-pass filter with 800 training samples under different ranks and λ. The rank-1 initialization works the best in this example.

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. 2.6 (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. 2.6 (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. 2.6 (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

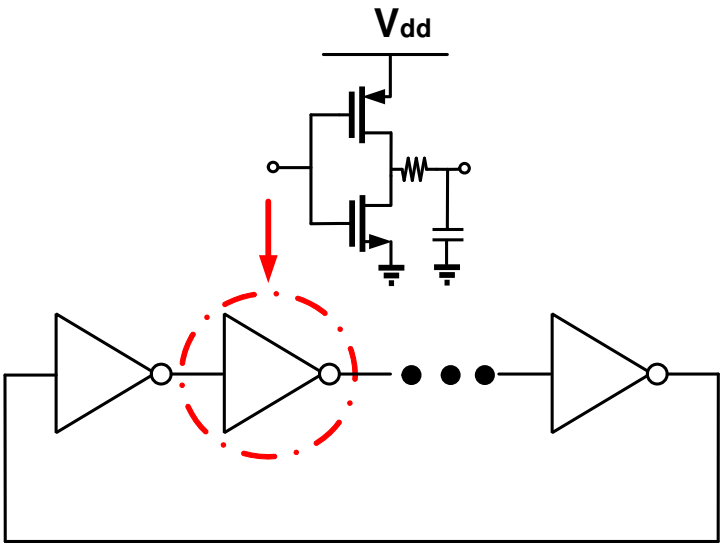

Figure 2.8: Schematic of a CMOS ring oscillator.

approximation with different initial tensor ranks R and different regularization parameters λ as illustrated in Fig. 2.7. 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. 2.6, where the predicted rank is 1. We also compare the complexity and accuracy of all methods in Table 2.2. The proposed method achieves the best accuracy with limited simulation samples.

2.6.4 CMOS Ring Oscillator (57-dim)

We continue to consider the 7-stage CMOS ring oscillator in Fig. 2.8. 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

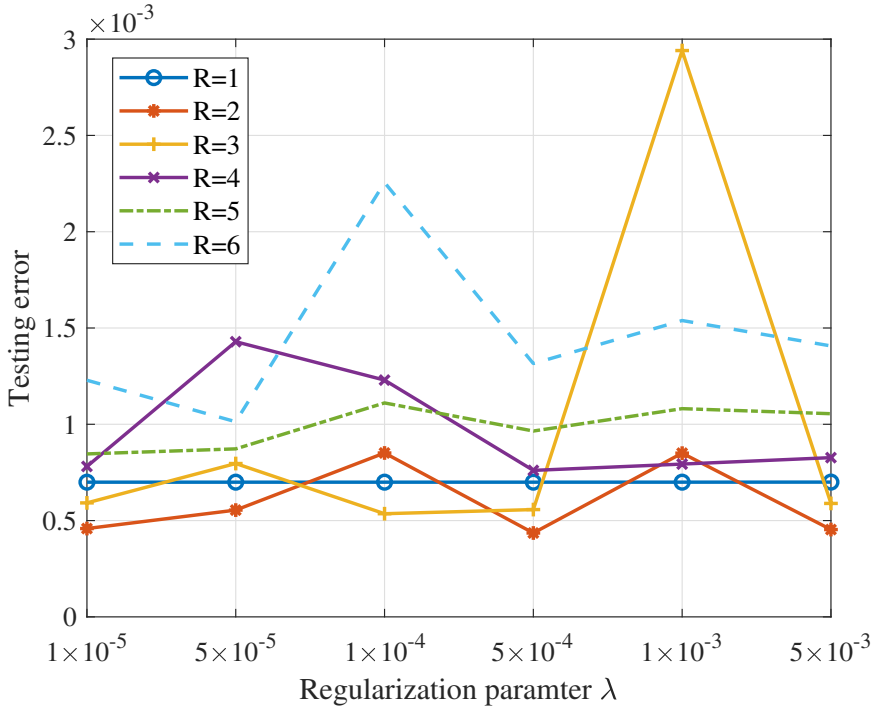

Figure 2.9: One-shot approximations for the CMOS ring oscillator with 150 training samples under different ranks and λ. In this example, the rank-2 model works the best in most cases.

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. 2.10 (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. 2.10 (b) shows that the estimated tensor rank reduces to 2 in all methods. Fig. 2.10 (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. 2.9. Conforming with the results shown in Fig. 2.10, a rank-2 model is more suitable in this example. We compare the proposed

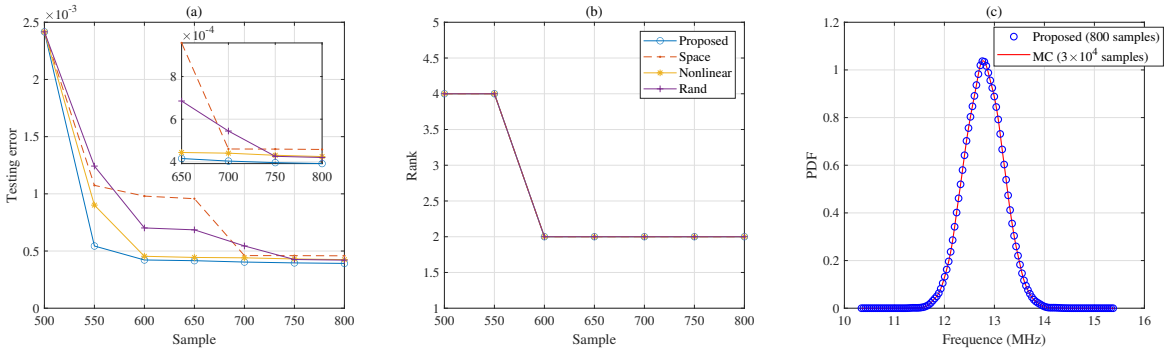

Figure 2.10: Results of the CMOS ring oscillator. (a) Testing error on 3×10^4 MC samples. (b) The estimated rank. (c) Probability density functions of the oscillator frequency.

Table 2.3: 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%

method with the fixed rank model and the 2nd-order sparse gPC in Table 2.3, where the proposed compact tensor model is shown to have the best approximation accuracy.

2.7 Conclusion

This chapter 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.

Chapter 3

Chance-Constrained Yield-Aware Circuit Optimization

3.1 Introduction

In this chapter, we will tackle the issue of over-conservative design in circuit yield optimization and propose a chance-constrained yield-aware formulation for it.

The increasing process variations have resulted in significant performance degradation and yield loss in semiconductor chip design and fabrications [5, 6]. Compared with electronic ICs, photonic ICs are more sensitive to process variations (e.g., geometric uncertainties) due to their large device dimensions compared with the small wavelength. Therefore, yield modeling and optimization for photonic ICs are highly desired [92, 93].

The yield optimization and yield-aware robust design have been investigated in the electronic design automation community for a long time and have been paid increasing attention in the photonic design automation [94–97]. Typical yield-aware design techniques include geometric approaches [98], geostatistics-motivated performance modeling [99], yield-aware Pareto surface [100], yield-driven iterative robust design [101], com-

putational intelligence assisted approaches [102], corner-based method [103], Bayesian yield optimization [104] and so forth. Advanced yield estimators can be generally classified as Monte-Carlo-based [105, 106] and non-Monte-Carlo-based [107–109] methods. Among the non-Monte-Carlo ones, surrogate modeling aims to approximate some circuit behaviors under variations to speed up the sampling and simulation process [66,110,111]. Typical surrogate models include posynomial models [112], linear quadratic models [113], support vector machine [114, 115], Gaussian process [116, 117], sparse polynomial [118], generalized polynomial chaos expansions [40, 42], and some variants [119]. Focusing on generalized polynomial chaos expansion, advanced techniques have also been developed to handle high-dimensional [23, 24, 28, 29, 52], mixed-integer [44] or non-Gaussian correlated [51] process variations. The polynomial-based modeling and optimization has been widely used in both electronics [43, 45, 120–122] and photonic IC design [123–126].

While most existing yield optimization approaches try to maximize the yield of a circuit, the obtained design performance (e.g., signal gain, power dissipation) may be far from the achievable optimal solution. Recently, an alternative approach was proposed in [25] to achieve both excellent yield and design performance. Instead of simply maximizing the yield, the work [25] optimizes a design performance metric while enforcing a high yield requirement. Specifically, the yield requirement is formulated as some chance constraints [127], which are further transformed to tractable constraints of the first and second statistical moments. The chance-constrained modeling itself has been widely used in many engineering fields [128–130]. The moment bounding method offers a provably sufficient condition of the chance constraint. However, the bounding gap may be too large in many applications [131]. The resulting overly-reduced feasible region may lead to an over-conservative design.

Main contributions. This chapter proposes a novel **Po**lynomial **B**ounding method for chance-constrained yield-aware **O**ptimization (PoBO) under truncated non-Gaussian

correlated variations. Leveraging the recent uncertainty quantification techniques [25,51], PoBO employs and modifies the idea of kinship functions [132] to approximate the original chance constraints with a better polynomial bounding method. PoBO provides a less conservative design than moment-based bounding methods [25] while ensuring a pre-specified yield requirement. The specific contributions of this chapter include:

- A better bounding method of the chance constraints via optimal polynomial kinship functions. Compared with existing work in the control community [132], we avoid the assumption of the independence among random variables and the convexity of kinship functions. The relaxation allows more general non-Gaussian correlated uncertainty modeling and tightens the bounding functions. Within a family of polynomial functions, the optimal polynomial kinship functions can be efficiently solved via semidefinite programming. Our bounding method preserves the polynomial formulation of the provided surrogate models. It enables the advanced polynomial optimization solvers, which provide a sequence of convex relaxations via semidefinite optimization and searches for the global design.

- Numerical implementation of the PoBO framework. Based on available uncertainty quantification solvers, we implement PoBO efficiently based on some pre-calculated optimal polynomial kinship functions and quadrature samples and weights without requiring any additional circuit simulations.

- Validations on a synthetic function and two photonic IC design examples. Our method offers better design performance while meeting the pre-specified yield requirements. This method requires a small number of circuit simulations due to its compatibility with recent data-efficient uncertainty quantification methods [23, 51].

While this work focuses on the fundamental theory, algorithms, and their validation on small-size photonic circuits, the proposed method can be combined with sparse or low-

rank surrogate modeling methods [23, 52] to handle large-scale design cases with much more design variables and process variations.

3.2 Background

This section reviews chance-constrained yield-aware optimization and its implementation via moment bounding [25].

3.2.1 Chance-Constrained Yield-Aware Optimization

We denote the design variables by $\mathbf{x} = [x_1, x_2, \ldots, x_{d_1}]^T \in \mathbf{X}$, and the process variations by random parameters $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_{d_2}]^T \in \boldsymbol{\Xi}$. Let $\{y_i(\mathbf{x}, \boldsymbol{\xi})\}_{i=1}^n$ denote n performance metrics that are considered in yield estimation, $\{u_i\}_{i=1}^n$ denote their corresponding upper bounds specifying the design requirements. An indicator function is defined as

$$I(\mathbf{x}, \boldsymbol{\xi}) = \begin{cases} 1, & y_i(\mathbf{x}, \boldsymbol{\xi}) \leq u_i, \forall i = [n]; \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

Here $[n] = \{1, 2, \cdots, n\}$. The yield at a certain design choice $\mathbf{x}$ is defined as

$$Y(\mathbf{x}) = \mathbb{P}_{\boldsymbol{\xi}}(\mathbf{y}(\mathbf{x}, \boldsymbol{\xi}) \leq \mathbf{u}) = \mathbb{E}_{\boldsymbol{\xi}}[I(\mathbf{x}, \boldsymbol{\xi})]. \tag{3.2}$$

In conventional yield optimization, one often tries to achieve the best possible yield. This often requires losing remarkable design performance $f(\mathbf{x}, \boldsymbol{\xi})$ in order to achieve a high yield.

Simply maximizing the yield may lead to an **over-conservative** design. As an example shown in Fig. 3.1, one may lose lots of performance (from 2.2 to 1.4) while just getting marginal yield improvement from 99% to 100%. In order to avoid an over-conservative

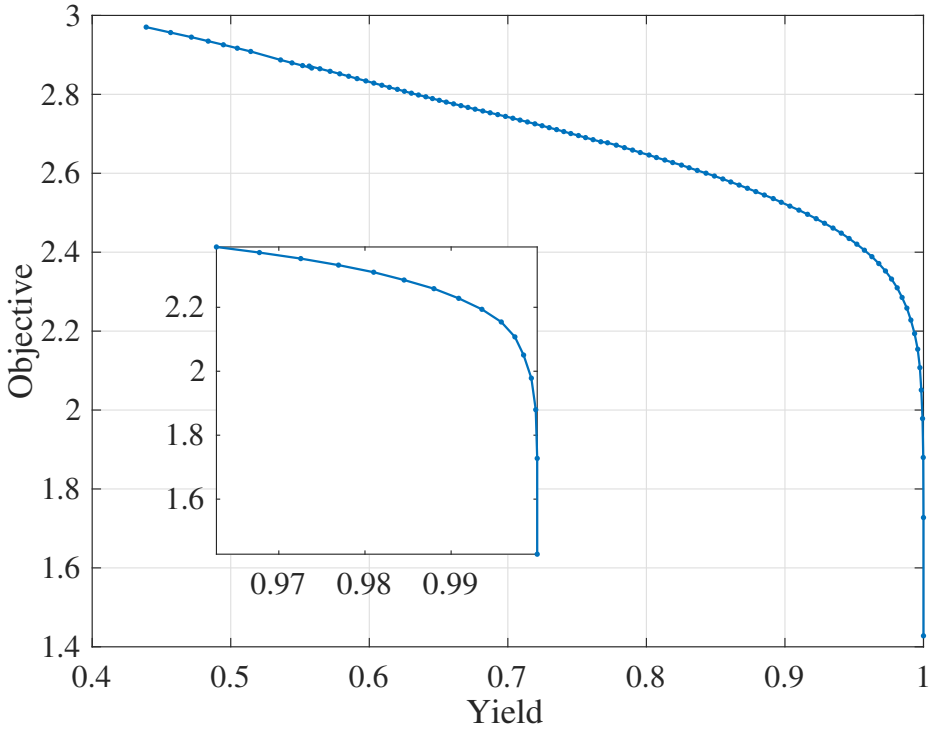

Figure 3.1: The trade-off between yield and an objective design performance for the example from 3.5.1. Simply maximizing the yield can lead to over-conservative performance.

design, a chance-constrained optimization was proposed in [25]:

$$\max_{\mathbf{x}\in\mathbf{X}} \quad \mathbb{E}_{\boldsymbol{\xi}}[f(\mathbf{x},\boldsymbol{\xi})] \tag{3.3a}$$

$$\text{s.t.} \quad \mathbb{P}_{\boldsymbol{\xi}}(y_i(\mathbf{x},\boldsymbol{\xi}) \leq u_i) \geq 1-\epsilon_i, \forall i = [n]. \tag{3.3b}$$

where $f(\mathbf{x},\boldsymbol{\xi})$ is the performance metric that we aim to optimize, and $\epsilon_i \in [0,1]$ is a risk level to control the probability of meeting each design constraint. Instead of simply maximizing the yield, the chance-constrained optimization tries to achieve a good balance between yield and performance: one can optimize the performance $f(\mathbf{x},\boldsymbol{\xi})$ while ensuring a high yield. The circuit yield can be controlled by ϵ_i: reducing ϵ_i ensures a lower failure rate and thus a higher yield.

The chance-constrained optimization (3.3) is generally hard to solve. Firstly, the feasible set produced by the chance constraints is often non-convex and hard to estimate. Secondly, it is also expensive to estimate the design objective function $f(\mathbf{x}, \boldsymbol{\xi})$ and design constraint function $y_i(\mathbf{x}, \boldsymbol{\xi})$ due to the lack of analytical expressions. Fortunately, a moment bounding method was combined with uncertainty quantification techniques in [25] to make the problem tractable.

3.2.2 Moment Bounding Method for (3.3)

Employing the Chebyshev-Cantelli inequality, one can ensure the chance constraint via a moment bounding technique [25, 133]. Specifically, with the first and second-order statistical moments of the constraint function, one can convert the probabilistic constraint in (3.3) to a deterministic one:

$$\begin{aligned} \max_{\mathbf{x}\in\mathbf{X}} \quad & \mathbb{E}_{\boldsymbol{\xi}}[f(\mathbf{x}, \boldsymbol{\xi})] \\ \text{s.t.} \quad & \mathbb{E}_{\boldsymbol{\xi}}[y_i(\mathbf{x}, \boldsymbol{\xi})] + \gamma_{\epsilon_i} \sqrt{\mathrm{Var}_{\boldsymbol{\xi}}[y_i(\mathbf{x}, \boldsymbol{\xi})]} \leq u_i, \forall i \in [n], \end{aligned} \tag{3.4}$$

where constant $\gamma_{\epsilon_i} = \sqrt{\frac{1-\epsilon_i}{\epsilon_i}}$. When the objective and constraint functions are described by certain surrogate models such as generalized polynomial chaos [42, 51], one can easily extract their mean and variances. This can greatly simplify the problem and reduce the computational cost, as shown by the yield-aware optimization of photonic IC in [25].

The moment constraint in (3.4) is a sufficient but unnecessary condition of the original chance constraint in (3.3). Therefore, any feasible solution of (3.4) should satisfy the probability constraint of (3.3). However, the feasible set produced by a moment bounding can be much smaller than the exact one [131]. This usually leads to an over-conservative design solution. When the risk level is very small, the moment bounding method may even produce an empty feasible set, leading to an unsolvable problem (see Section 3.5).

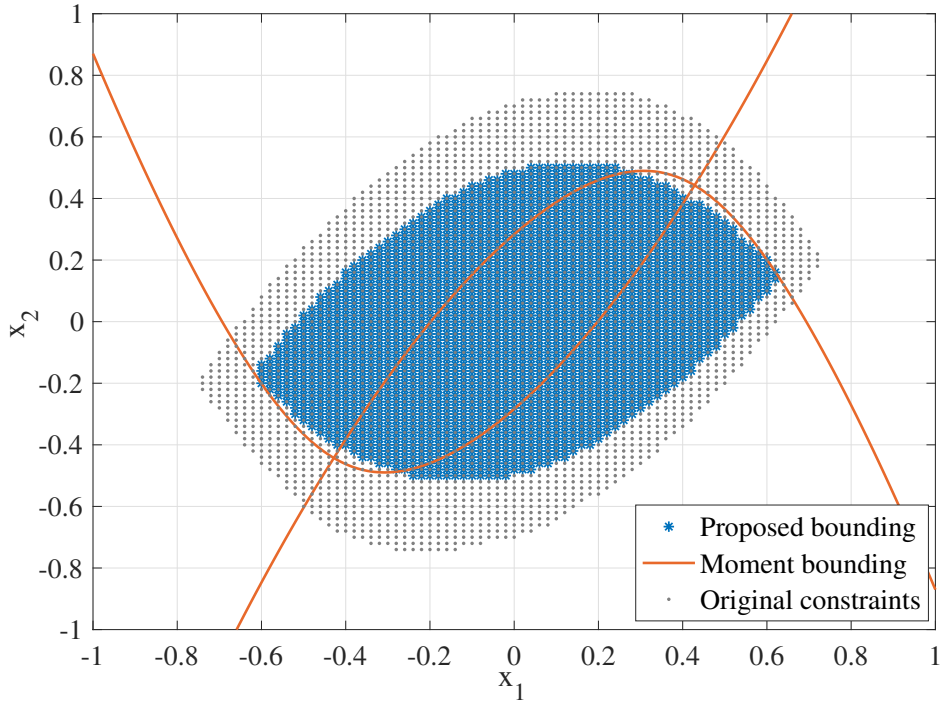

Figure 3.2: The feasible set of an original chance constraint, moment bounding (the cross region of two orange lines), and the proposed polynomial bounding.

3.3 Proposed Polynomial Bounding Method

To avoid the possible over-conservative bounding of the moment methods [25], we propose to bound the chance constraint via a more accurate polynomial method. Fig. 3.2 plots the feasible regions obtained by different bounding methods for the synthetic example in Sec. 3.5.1. For the same chance constraints, the moment bounding method produces a feasible set that is much smaller than the exact one, whereas our polynomial bounding method generates a better approximation of the feasible set. Due to the more accurate approximation of the feasible set, our proposed polynomial bounding method can provide a less conservative design in yield-aware optimization. Now we describe how to generate the polynomial bounds via kinship functions.

3.3.1 Kinship Function

The kinship function was first proposed to construct a convex approximation of an indicator function in [132]. We generalize the concepts of [132] with two relaxations:

- We do not require the convexity of a kinship function.
- We do not require the random variables $\boldsymbol{\xi}$ to be mutually independent. Instead, we consider the more challenging cases where random parameters are truncated non-Gaussian correlated.

We slightly modify the definition of a kinship function.

Definition 1 (Kinship function) *A kinship function $\kappa(z) : [-1, \infty) \to \mathbb{R}$ is a function that satisfies the following constraints:*

- $\kappa(z) = 1$ *when* $z = 0$;
- $\kappa(z) \geq 0$ *for any* $z \in [-1, +\infty)$;
- $\kappa(z_1) \geq \kappa(z_2)$ *for any* $z_1 \geq z_2$ *in the range* $[-1, \infty)$.

Based on kinship functions, the following theorem offers an upper bound for any probability of constraint violations.

Theorem 1 (Risk integral [132]) *Let $\kappa(\cdot)$ be a kinship function, $g(\mathbf{x}, \boldsymbol{\xi}) \geq -1$, $\mu(\boldsymbol{\xi})$ be the density function of random vector $\boldsymbol{\xi}$, and $V_\kappa(\mathbf{x})$ be a risk integral quantity:*

$$V_\kappa(\mathbf{x}) \coloneqq \int_{\boldsymbol{\Xi}} \kappa[g(\mathbf{x}, \boldsymbol{\xi})]\mu(\boldsymbol{\xi})d\boldsymbol{\xi}, \tag{3.5}$$

then we have $\mathbb{P}\{\boldsymbol{\xi} \in \boldsymbol{\Xi} : g(\mathbf{x}, \boldsymbol{\xi}) > 0\} \leq V_\kappa(\mathbf{x})$.

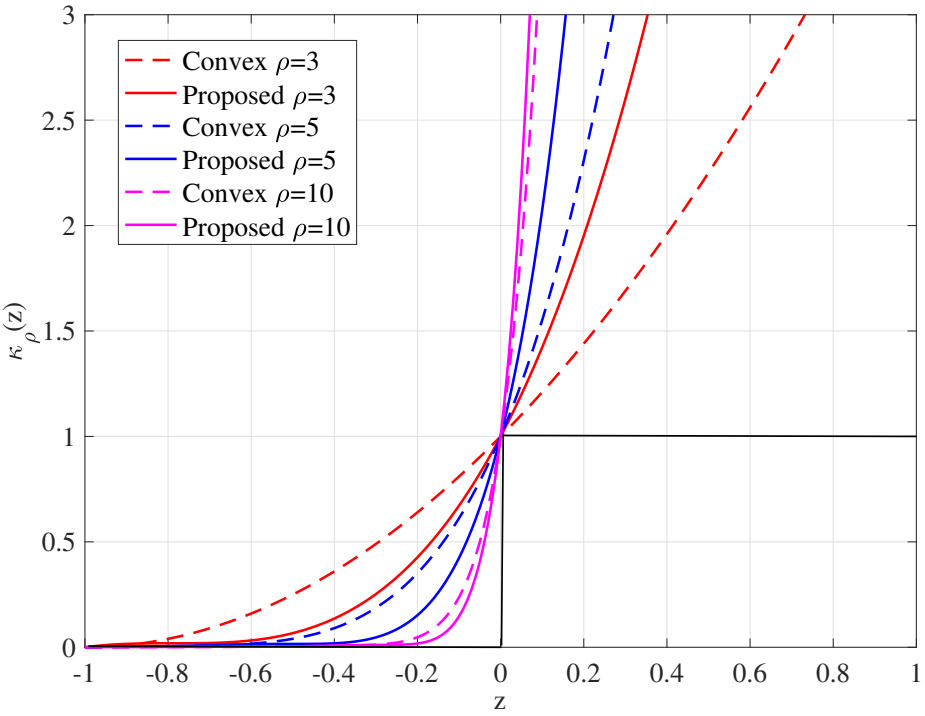

Figure 3.3: Different kinship functions. Black curve: an indicator function.

Proof: According to Definition 1, $\kappa[g(\mathbf{x},\boldsymbol{\xi})]$ is nonnegative in $[-1,\infty)$ and greater than 1 if $g(\mathbf{x},\boldsymbol{\xi}) \geq 0$. Therefore, for any probability measure $\mu(\boldsymbol{\xi})$ on $\boldsymbol{\Xi}$, we have:

$$V_\kappa(\mathbf{x}) \geq \int_{\{\boldsymbol{\xi}\in\boldsymbol{\Xi}, g(\mathbf{x},\boldsymbol{\xi})>0\}} \kappa[g(\mathbf{x},\boldsymbol{\xi})]\mu(\boldsymbol{\xi})d\boldsymbol{\xi} \tag{3.6a}$$

$$\geq \int_{\{\boldsymbol{\xi}\in\boldsymbol{\Xi}: g(\mathbf{x},\boldsymbol{\xi})>0\}} \mu(\boldsymbol{\xi})d\boldsymbol{\xi} = \mathbb{P}\{\boldsymbol{\xi}\in\boldsymbol{\Xi}, g(\mathbf{x},\boldsymbol{\xi})>0\}. \tag{3.6b}$$

■

There exist many possible choices of kinship functions. Next, we will show how to pick some polynomial kinship functions. We consider the polynomial function family because it is compatible with existing surrogate modeling techniques [23,42,52] to facilitate yield-aware optimization.

3.3.2 Optimal Polynomial Kinship Function

An optimal kinship function is defined as a kinship function that minimizes its integral over $[-1, 0]$ [132]:

$$\kappa^\star(\cdot) \coloneqq \underset{\kappa(\cdot)\in\mathcal{K}}{\operatorname{argmin}} \int_{-1}^{0} \kappa(z)dz. \tag{3.7}$$

Here $\mathcal{K}$ is the set of all possible kinship functions that satisfy the constraints in Definition 1. Let $z = g(\mathbf{x}, \boldsymbol{\xi})$, then the above definition can be understood as minimizing the gap between the left- and right-hand sides of (3.6a).

Now we consider choosing a kinship function from a family of order-ρ polynomials $\mathcal{K}_\rho \coloneqq \{\kappa(z)|\kappa(z) = \sum_{i=0}^{\rho} \zeta_i z^i\}$. The optimal polynomial kinship function, denoted as $\kappa_\rho(\cdot)$, can be constructed by solving the following optimization problem:

$$\begin{aligned} \min_{\zeta_0,\ldots,\zeta_\rho} \quad & \int_{-1}^{0} \kappa_\rho(z)dz \\ \text{s.t.} \quad & \kappa_\rho(z) = \zeta_0 + \zeta_1 z + \cdots + \zeta_\rho z^\rho \in \mathcal{K}_\rho, \\ & \kappa_\rho(0) = 1,\ \kappa_\rho(-1) = 0, \\ & \kappa_\rho'(z) \geq 0,\ \forall z \geq -1. \end{aligned} \tag{3.8}$$

The optimization problem can be reformulated as semidefinite programming and the details are provided in Appendix A.2.1. The obtained optimal polynomial kinship function $\kappa_\rho(\cdot)$ with different orders are visualized in Fig. 3.3. Since we relax the convexity requirement, given the same polynomial order, the proposed polynomial function is a tighter approximation to the indicator function than a convex one [132] for $z \in [-1, 0]$.

Based on the obtained optimal polynomial kinship function $\kappa_\rho(\cdot)$, we bound the original chance constraint (3.3b) by enforcing the upper bound of the failure probability below ϵ_i:

$$\mathbb{P}_{\boldsymbol{\xi}}(y_i(\mathbf{x},\boldsymbol{\xi}) > u_i) \leq \int_{\boldsymbol{\Xi}} \kappa_\rho(y_i(\mathbf{x}, \boldsymbol{\xi}) - u_i)\mu(\boldsymbol{\xi})d\boldsymbol{\xi} \leq \epsilon_i. \tag{3.9}$$

The order of $\kappa_\rho(\cdot)$ controls the upper bound of violation probability. When ρ is small, the bounding gap in (3.6a) is large. As $\rho \to \infty$, the polynomial bounding leads to a worst-case robust design optimization according to Theorem 3 in [132]. This is equivalent to setting a risk level $\epsilon_i = 0$, which leads to an extremely over-conservative design. Fortunately, this is not a trouble in practice since we do not use a very high-order polynomial due to the computational issues. In practice, there exists an optimal order $\rho^\star$ for bounding the violation probability most accurately. The optimal $\rho^\star$ is unknown *a-priori*, but heuristically we find that setting $\rho \in [5, 10]$ usually offers an excellent bound. The proposed bounding method can be also extended to deal with joint chance constraints by constructing a multivariate polynomial kinship function.

3.4 The PoBO Framework

Based on the proposed polynomial bounding for chance constraints, we further present the novel PoBO method to achieve less conservative yield-aware optimization.

3.4.1 Workflow of PoBO

Our PoBO framework has two weak assumptions on the design and random variables:

- The design variable $\mathbf{x}$ is box-bounded, i.e., $\mathbf{x} \in \mathbf{X} = [a, b]^{d_1}$. This is normally the case in circuit optimization.
- The process variations $\boldsymbol{\xi}$ are truncated and non-Gaussian correlated with a joint probability density function $\mu(\boldsymbol{\xi})$.

The 2nd assumption is not strong at all. Many practical process variations are correlated and not guaranteed to be Gaussian. Additionally, the values of almost all practical geometric or material parameters are bounded, although some simplified unbounded

distributions (e.g., Gaussian distributions) were used in previous literature for ease of implementation.

The overall flow of PoBO is summarized below.

- **Step 1:** Surrogate modeling. We use the recent uncertainty quantification solver [51] to construct polynomial surrogate models for the objective and constraint functions, i.e., $f(\mathbf{x}, \boldsymbol{\xi}) \approx \hat{f}(\mathbf{x}, \boldsymbol{\xi})$ and $y_i(\mathbf{x}, \boldsymbol{\xi}) \approx \hat{y}_i(\mathbf{x}, \boldsymbol{\xi}), \forall i = [n]$.
- **Step 2:** Bounding the chance constraints via the proposed optimal polynomial kinship functions. This transforms a chance-constrained probabilistic optimization problem into a tractable deterministic one with a high-quality solution.
- **Step 3:** Design optimization. We use a polynomial optimization solver, e.g., the semidefinite programming relaxation [134], to obtain a globally optimal solution.

The PoBO framework reformulates the original chance-constrained optimization (3.3) to the following optimization:

$$\max_{\mathbf{x}\in\mathbf{X}} \quad \mathbb{E}_{\boldsymbol{\xi}}[\hat{f}(\mathbf{x}, \boldsymbol{\xi})] \tag{3.10a}$$

$$\text{s.t.} \quad V_{\kappa}^{(i)}(\mathbf{x}) = \int_{\boldsymbol{\Xi}} \kappa_{\rho}(\upsilon_i(\mathbf{x}, \boldsymbol{\xi}))\mu(\boldsymbol{\xi})d\boldsymbol{\xi} \leq \epsilon_i, \ \forall i \in [n]. \tag{3.10b}$$

Here $\upsilon_i(\mathbf{x}, \boldsymbol{\xi}) \coloneqq \hat{y}_i(\mathbf{x}, \boldsymbol{\xi}) - u_i$, $\hat{f}(\mathbf{x}, \boldsymbol{\xi})$ and $\hat{y}_i(\mathbf{x}, \boldsymbol{\xi})$ are the polynomial surrogate models of $f(\mathbf{x}, \boldsymbol{\xi})$ and $y_i(\mathbf{x}, \boldsymbol{\xi})$, respectively.

In the following subsections, we will describe the implementation details of this PoBO framework.

3.4.2 Building Surrogate Models

High-quality performance models are important to speed up design optimization. We employ the advanced stochastic collocation method with non-Gaussian correlated

uncertainty [25, 51] in Step 1. This method approximates a smooth stochastic function as the linear combination of some orthogonal and normalized polynomial basis functions:

$$f(\mathbf{x}, \boldsymbol{\xi}) \approx \hat{f}(\mathbf{x}, \boldsymbol{\xi}) = \sum_{|\boldsymbol{\alpha}|+|\boldsymbol{\beta}|=0}^{p} c_{\boldsymbol{\alpha},\boldsymbol{\beta}} \boldsymbol{\Phi}_{\boldsymbol{\alpha}}(\mathbf{x}) \boldsymbol{\Psi}_{\boldsymbol{\beta}}(\boldsymbol{\xi}). \tag{3.11}$$

Here $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are two index vectors, $\boldsymbol{\Phi}_{\boldsymbol{\alpha}}(\mathbf{x})$ and $\boldsymbol{\Psi}_{\boldsymbol{\beta}}(\boldsymbol{\xi})$ are two series of orthogonal polynomial basis functions, and p upper bounds the total order of the product of two basis functions. The corresponding coefficients $c_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ are calculated via a projection method using some optimization-based quadrature samples and weights of $\mathbf{x}$ and $\boldsymbol{\xi}$ [51]. When the parameter dimensionality is not high, this method only needs a small number of simulation samples to produce a highly accurate surrogate model with a provable error bound. When the number of dimensions becomes high, we could utilize many existing advanced uncertainty quantification techniques to model the performance more efficiently [29, 53]. It is also possible to extend the proposed chance-constrained yield-aware method to other types of performance models.

3.4.3 Scaling the Yield Metrics $\upsilon_i(\mathbf{x}, \boldsymbol{\xi})$

To bound the failure probability $\mathbb{P}\{\boldsymbol{\xi} \in \boldsymbol{\Xi} : \upsilon_i(\mathbf{x}, \boldsymbol{\xi}) > 0\}$ via the optimal kinship function, $\upsilon_i(\mathbf{x}, \boldsymbol{\xi})$ must be in the range $[-1, \infty)$ according to Definition 1. Once $\upsilon_i(\mathbf{x}, \boldsymbol{\xi})$ is lower bounded, we can always scale it to meet this requirement.

Since $\upsilon_i(\mathbf{x}, \boldsymbol{\xi})$ is a polynomial function in our problem setting, and both $\mathbf{x}$ and $\boldsymbol{\xi}$ are assumed bounded, we can compute the minimum value $\upsilon_i^- \coloneqq \min_{\mathbf{x}\in\mathbf{X}, \boldsymbol{\xi}\in\boldsymbol{\Xi}} \upsilon_i(\mathbf{x}, \boldsymbol{\xi})$. Then we change the lower bound of $\upsilon_i(\mathbf{x}, \boldsymbol{\xi})$ to -1 as follows:

$$\upsilon_i(\mathbf{x}, \boldsymbol{\xi}) \longleftarrow -\tfrac{1}{\upsilon_i^-} \upsilon_i(\mathbf{x}, \boldsymbol{\xi}), \ \forall i \in [n]. \tag{3.12}$$

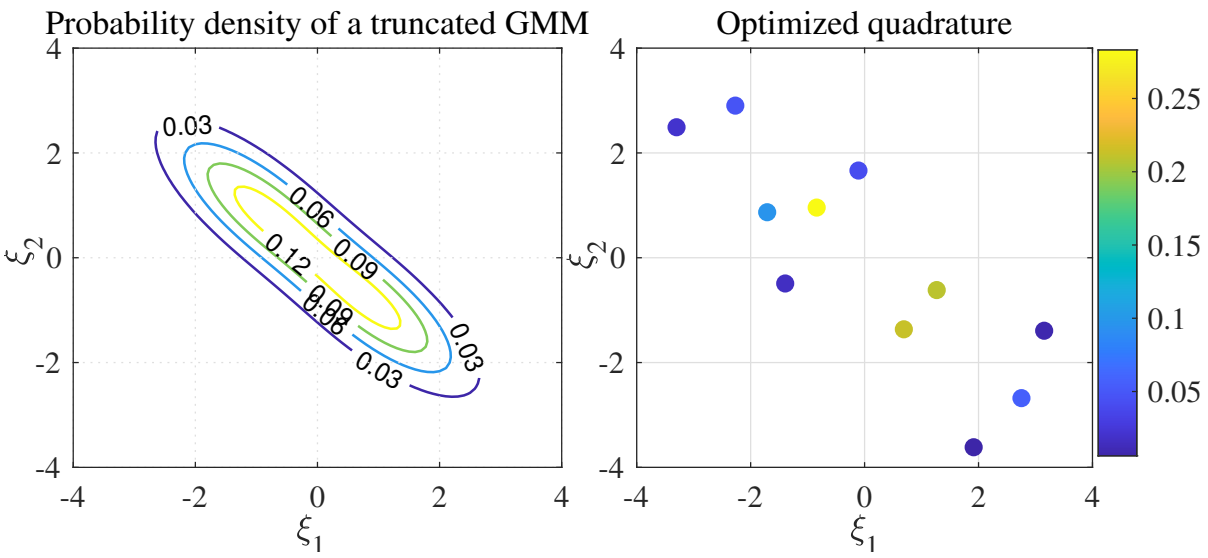

Figure 3.4: An example of optimized quadrature rule for a two-dimensional truncated Gaussian mixture model (GMM) with two components. The color bar of the right figure represents the weights of all samples. The shown quadrature rule satisfies the exact integration up to order 6.

The scaling factor $-\frac{1}{\upsilon_i^-}$ is positive as long as the problem (3.10) is solvable. This is because the existence of $\mathbf{x} \in \mathbf{X}$ and $\boldsymbol{\xi} \in \boldsymbol{\Xi}$ such that $\upsilon_i(\mathbf{x}, \boldsymbol{\xi}) < 0$ is the necessary condition to satisfy the yield constraint. We can easily avoid $\upsilon_i^- = 0$ by adding a sufficiently small perturbation.

3.4.4 Calculating Risk Integral $V_\kappa^{(i)}(\mathbf{x})$

In order to upper bound the probability of violating a design constraint, we need to calculate the integration in (3.10b). This can be a challenging task for a truncated non-Gaussian correlated random vector $\boldsymbol{\xi}$ since classical numerical quadrature rules [135, 136] do not work for non-Gaussian correlated variables. Fortunately, we can reuse the quadrature rule of $\boldsymbol{\xi}$ as the by-product of building the surrogate models in Sec. 3.4.2. Specifically, the quadrature points and weights $\{\boldsymbol{\xi}_l, w_l\}_{l=1}^M$ compute the exact integration

Algorithm 3 Flow of the proposed PoBO.

Input: Box-bounded design variable $\mathbf{x} \in \mathbf{X}$, truncated non-Gaussian correlated variations $\boldsymbol{\xi} \in \boldsymbol{\Xi}$, risk levels ϵ

Output: Optimized design $\mathbf{x}^\star$

1: Formulate the chance-constrained problem (3.3).
2: Obtain surrogate models for the design objective function $f(\mathbf{x}, \boldsymbol{\xi}) \approx \hat{f}(\mathbf{x}, \boldsymbol{\xi})$ and constraint functions $y_i(\mathbf{x}, \boldsymbol{\xi}) \approx \hat{y}_i(\mathbf{x}, \boldsymbol{\xi}), \forall i \in [n]$.
3: Scale the yield metrics $\upsilon_i(\mathbf{x}, \boldsymbol{\xi})$ via Eq. (3.12).
4: Compute the optimal polynomial kinship function $\kappa_\rho(\cdot)$ via (3.8).
5: Calculate risk integral $V_\kappa^{(i)}(\mathbf{x})$ via the quadrature rule obtained from (3.13).
6: Seek the optimal design of problem (3.14) via a global polynomial optimization solver.

up to order $2q$, obtained by solving the following optimization problem [51]:

$$\min_{\boldsymbol{\xi}_l, w_l \geq 0} \sum_{|\boldsymbol{\beta}|=0}^{2q} \left(\mathbb{E}_{\boldsymbol{\xi}} \left[\boldsymbol{\Psi}_{\boldsymbol{\beta}}(\boldsymbol{\xi}) \right] - \sum_{l=1}^{M} \boldsymbol{\Psi}_{\boldsymbol{\beta}}(\boldsymbol{\xi}_l) w_l \right)^2. \tag{3.13}$$

An example of the solved quadrature is shown in Fig. 3.4. The number of quadrature samples could be controlled by tuning the optimization precision. The detailed accuracy analysis and the bound of M are provided in [51], which is omitted here.

Theoretically, we need a quadrature rule to exactly calculate the integration up to order $p\rho$ in (3.10b). The exact quadrature rule can be computed offline via solving (3.13) with $q = \lceil \frac{\rho p}{2} \rceil$. In practice, a low-order quadrature rule, like $q = p$, often offers sufficient numerical accuracy. Therefore, we can directly use the quadrature rule used when building surrogate model (3.11) to calculate the risk integral $V_\kappa^{(i)}(\mathbf{x})$. Since the quadrature points will not be simulated, it does not introduce any additional computational burdens. Based on the quadrature rule, problem (3.10) can be converted to the following

deterministic constrained polynomial optimization (3.14):

$$\begin{aligned} \max_{\mathbf{x}\in\mathbf{X}} \quad & \mathbb{E}_{\boldsymbol{\xi}}[\hat{f}(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} \quad & \sum_{l=1}^{M} w_l \kappa_\rho(\upsilon_i(\mathbf{x},\boldsymbol{\xi}_l)) \le \epsilon_i, \quad \forall i \in [n]. \end{aligned} \tag{3.14}$$

Note that the expectation value in the objective function can be easily obtained since $\hat{f}(\mathbf{x},\boldsymbol{\xi})$ is a generalized polynomial-chaos expansion [42, 51].

3.4.5 Algorithm Summary

We summarize PoBO in Algorithm 3. Below are some remarks:

- In line 4 of Algorithm 3, the optimal polynomial kinship functions (3.8) can be computed offline and stored as a look-up table.
- In line 5 of Algorithm 3, the optimization-based quadrature rule in [51] can be used to calculate the risk integral without any additional simulations. The quadrature rule can be computed offline as well.
- This method enables a global polynomial optimization solver to obtain the optimal design of (3.14).

The curse of dimensionality could be a challenge for Line 2 and Line 6. For the surrogate modeling step, we can utilize some high-dimensional uncertainty quantification techniques [29, 53] to reduce the cost. In the design optimization step, the current limitation comes from the polynomial optimization solver. Typically, the polynomial optimization problem can be reformulated as a convex moment problem. Under very mild assumptions, we can build a series of semidefinite programming problems whose solutions are proved to converge monotonically and asymptotically to the global optimum [137–139].

Table 3.1: Optimization Results for Synthetic Function

Risk level ϵ	Method	Objective	Δ_1 (%)	Δ_2 (%)	Yield (%)
0.01	Moment [25]	N/A*	N/A*	N/A*	N/A*
	Proposed	**1.14**	**1.01**	**0.99**	99.98
0.05	Moment [25]	1.88	5.25	5.26	99.98
	Proposed	**2.11**	**5.21**	**3.97**	98.76
0.1	Moment [25]	2.19	**10.47**	10.98	99.36
	Proposed	**2.26**	10.67	**7.98**	97.08

* The algorithm fails with no feasible solution.

The relaxed semidefinite programming problems have the size of $O({d_1}^p)$ with the number of design variables d_1 and polynomial order p. Although it grows polynomially with the number of design variables, it can be challenging when p is high. Fortunately, the design optimization does not suffer from the number of dimensions of process variations d_2. The challenge caused by the high dimensionality of $\mathbf{x}$ may be addressed in the future by using other nonlinear optimization solvers, or a better polynomial optimization solver (e.g. a sparse polynomial solver) that can exploit the sparse structure of the polynomial surrogate.

3.5 Numerical Results

In this section, we validate the proposed PoBO framework via the synthetic example and two realistic photonic IC examples from [25]. The polynomial optimization is solved via GloptiPoly 3 [134], which is a global optimization solver based on hierarchical semidefinite programming. Our codes are implemented in MATLAB and run on a computer with a 2.3 GHz CPU and 16 GB memory.

Baseline Methods. We choose the moment-bounding chance-constrained optimization [25] as the baseline for comparison. On the photonic IC benchmarks, we further compare our method with the Bayesian yield optimization (BYO) method [104], a recent

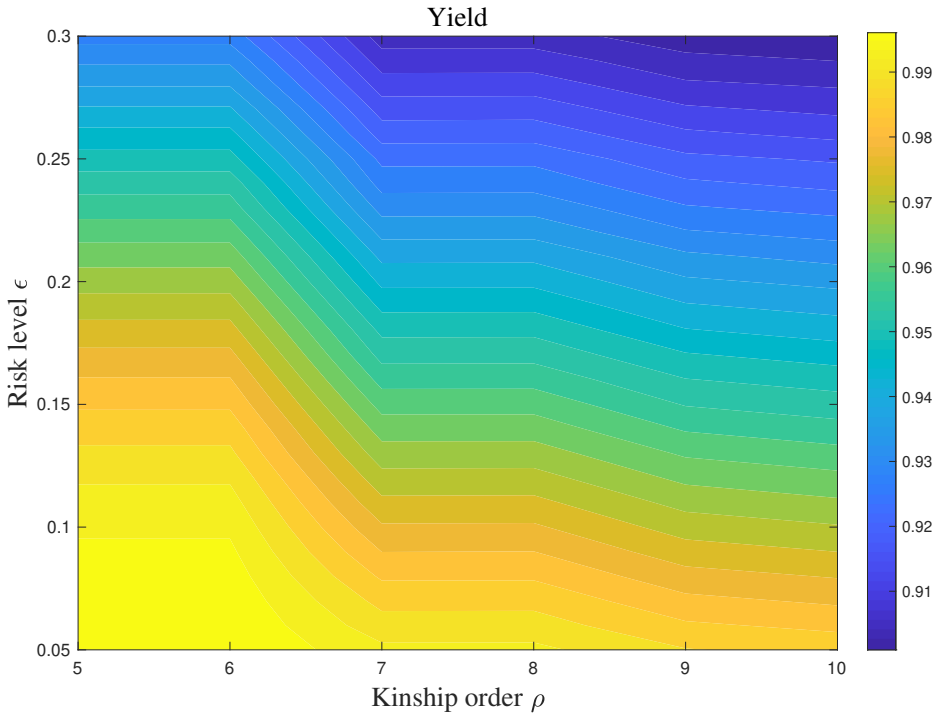

Figure 3.5: Yield of (3.16) given different risk levels ϵ and polynomial kinship orders ρ.

state-of-the-art yield optimization approach.

Gap of chance constraints. We modify the indicator function (3.1) to define an indicator function $I_i(\mathbf{x}, \boldsymbol{\xi})$ for each individual design constraint in yield definition:

$$I_i(\mathbf{x}, \boldsymbol{\xi}) = \begin{cases} 1, & \hat{y}_i(\mathbf{x}, \boldsymbol{\xi}) \leq u_i; \\ 0, & \text{otherwise} \end{cases}, \forall i = [n].$$

With N random samples, the *individual success rate* for each design constraint is evaluated as $Y_i(\mathbf{x}) = \sum_{j=1}^{N} I_i(\mathbf{x}, \boldsymbol{\xi}_j)/N$. The gap for the i-th chance constraint is the relative difference between Y_i and the pre-specified success rate $1 - \epsilon_i$:

$$\Delta_i = \frac{Y_i(\mathbf{x}) - (1 - \epsilon_i)}{1 - \epsilon_i}, \ \forall i = [n]. \tag{3.15}$$

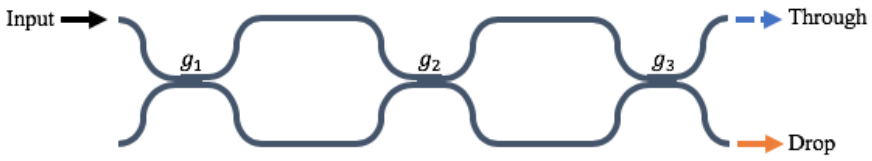

Figure 3.6: The schematic of a third-order Mach-Zehnder interferometer.

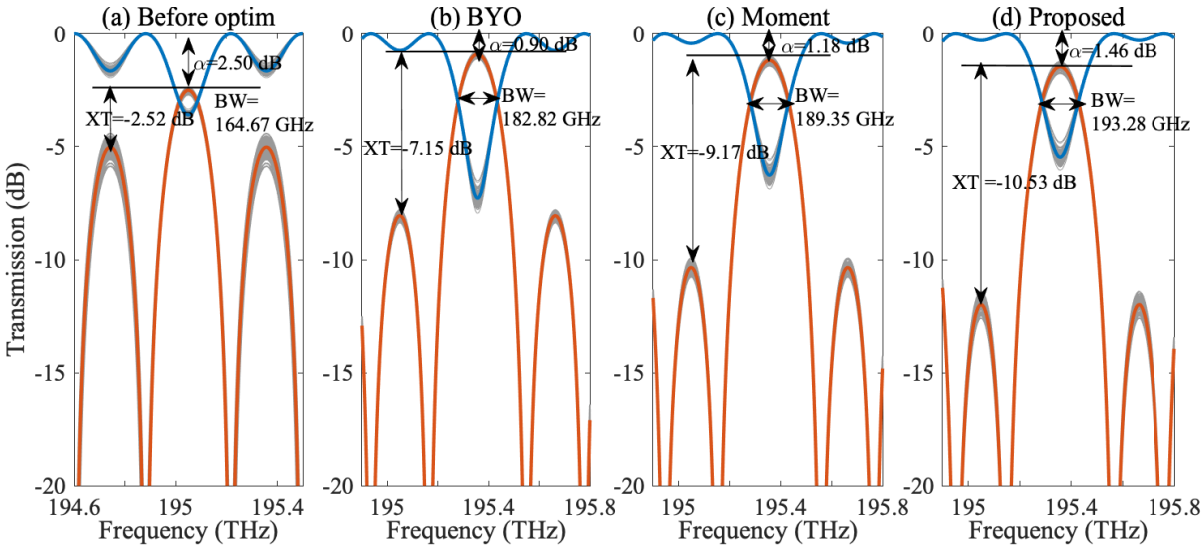

Figure 3.7: The transmission curves of the MZI. The grey lines show the performance uncertainties. The orange and blue curves show the transmission rates at the drop and through ports, respectively. The mean values of the bandwidth, crosstalk, and attenuation are denoted as BW, XT, and α, respectively. (a) The initial design: $\mathbf{x}$=[150, 150, 150]; (b) Design after Bayesian yield optimization [104]: $\mathbf{x}$=[286.63, 170.59, 299.3]; (c) Design with the moment-bounding yield-aware optimization [25]: $\mathbf{x}$=[300, 149.67, 300]; (d) Design with the proposed PoBO method: $\mathbf{x}$=[300, 112.15, 300].

The chance-constrained optimization can always provide a solution to certify the yield requirement controlled by ϵ_i's if a feasible solution exists. Therefore, Δ_i is always non-negative. Typically, a tighter probabilistic constraint bounding leads to a larger feasible region and allows us to explore the optimal design in a larger space, which is more likely to utilize more risk budgets. Therefore, we use Eq. (3.15) to measure the gap of feasible regions and the bounding quality. Notice that we do not attempt to achieve the

Table 3.2: Optimization Results for MZI Benchmark

Risk level ϵ	Method	$\mathbb{E}_{\boldsymbol{\xi}}$[BW] (GHz)	Δ_1 (%)	Δ_2 (%)	Yield (%)	Simulation #
0.05	Moment [25]	184.53	5.26	5.26	100	35
	Proposed	**190.99**	**5.26**	**5.26**	100	35
0.07	Moment [25]	187.02	7.53	7.53	100	35
	Proposed	**192.10**	**7.53**	**7.2**	99.7	35
0.1	Moment [25]	189.35	11.11	11.11	100	35
	Proposed	**193.28**	**11.11**	**4.22**	93.8	35
N/A*	BYO [104]	182.82	N/A*	N/A*	100	2020

* No risk level is defined for BYO method. Correspondingly, no gap Δ is defined.

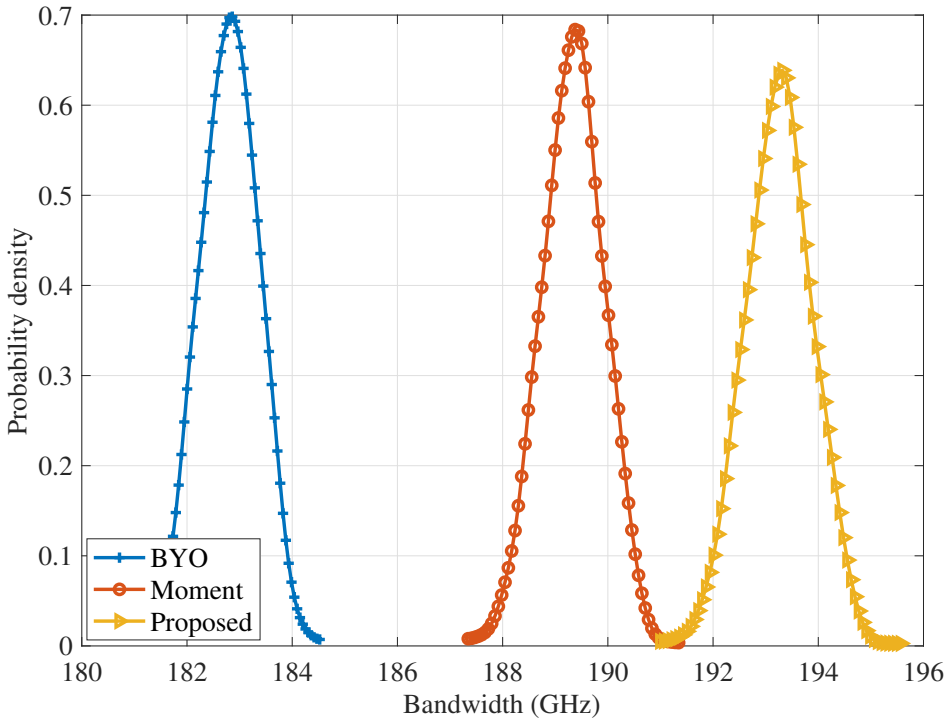

Figure 3.8: The probability density function of the optimized bandwidth of the MZI by Bayesian yield optimization [104], moment bounding [25] and the proposed PoBO (with $\epsilon = 0.1$).

highest yield. Instead, our goal is to *avoid over-conservative design while ensuring the pre-specified yield requirement*. Therefore, given a certain risk level, we **prefer a smaller gap** Δ_i and a less conservative design solution with better objective performance.

3.5.1 Synthetic Function

We first consider a synthetic function with design variables $\mathbf{x} \in \mathbf{X} = [-1, 1]^2$ and random parameters $\boldsymbol{\xi}$ following a truncated Gaussian mixture model. Specifically, we assume $\mu(\boldsymbol{\xi}) = \frac{1}{2}\mathcal{TN}(\bar{\boldsymbol{\mu}}_1, \boldsymbol{\Sigma}_1, \mathbf{a}_1, \mathbf{b}_1) + \frac{1}{2}\mathcal{TN}(\bar{\boldsymbol{\mu}}_2, \boldsymbol{\Sigma}_2, \mathbf{a}_2, \mathbf{b}_2)$, with $\bar{\boldsymbol{\mu}}_1 = -\bar{\boldsymbol{\mu}}_2 = [0.1, -0.1]^T$, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = 10^{-2}\begin{bmatrix} 1 & -0.75 \\ -0.75 & 1 \end{bmatrix}$, $\mathbf{a}_1 = -[0.2, 0.4]^T$, $\mathbf{a}_2 = -[0.4, 0.2]^T$, $\mathbf{b}_1 = [0.4, 0.2]^T$ and $\mathbf{b}_2 = [0.2, 0.4]^T$. Here $\mathcal{TN}(\bar{\boldsymbol{\mu}}, \boldsymbol{\Sigma}, \mathbf{a}, \mathbf{b})$ denotes a distribution that is a normal distribution with mean $\bar{\boldsymbol{\mu}}$ and variance $\boldsymbol{\Sigma}$ in the box $[\mathbf{a}, \mathbf{b}]$.

We consider the following chance-constrained optimization:

$$\begin{aligned} \max_{\mathbf{x}\in\mathbf{X}} \quad & \mathbb{E}_{\boldsymbol{\xi}}[3(x_1 + \xi_1) - (x_2 + \xi_2))] \\ \text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\xi}}((x_1 + \xi_1)^2 + (x_2 + \xi_2) \leq 1) \geq 1 - \epsilon_1, \\ & \mathbb{P}_{\boldsymbol{\xi}}((x_1 + \xi_1)^2 - (x_2 + \xi_2) \leq 1) \geq 1 - \epsilon_2, \end{aligned} \tag{3.16}$$

where the two risk levels are set to be equal $\epsilon_1 = \epsilon_2 = \epsilon$. Remark that we do not require $\epsilon_1 = \epsilon_2$ since our method naturally handles the individual constraints.

We use 2nd-order polynomials to approximate the three analytical functions and bound the chance constraint with an order-10 optimal polynomial kinship function. As shown in Table 3.1, compared with the moment method [25], the proposed PoBO method produces a better objective value and smaller gaps for chance constraints while meeting the pre-specified yield requirement. Clearly, a smaller ϵ_i produces a higher yield. The moment bounding method fails to work when $\epsilon = 0.01$ while our PoBO can still solve this problem. Fig. 3.5 shows the obtained yield under different risk levels ϵ and polynomial kinship orders. The kinship order influences the bounding quality and leads to different yield. However, the results are all of the high quality, leading to certified designs.

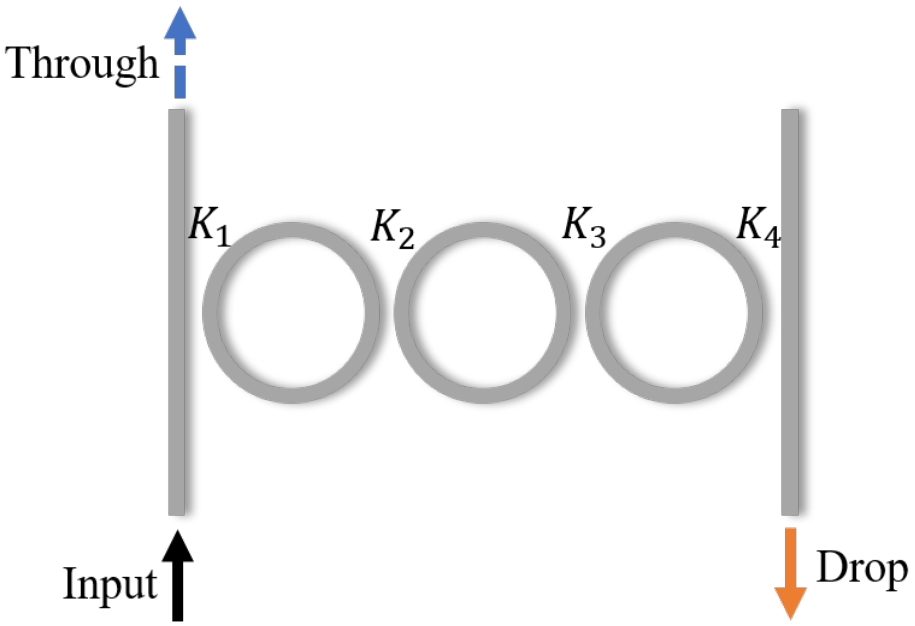

Figure 3.9: The schematic of a microring add-drop filter.

3.5.2 Mach-Zehnder Interferometer

We consider a third-order Mach-Zehnder interferometer (MZI) which consists of three port coupling and two arms, as shown in Fig. 3.6. The coupling coefficients τ between the MZ arms play an important role in the design, whose relationship with the the gap g (nm) is $\tau = \exp(-\frac{g}{260})$. The gap variables $\mathbf{x} = [g_1, g_2, g_3]$ have the design space of $\mathbf{X} = [100, 300]^3$. Their corresponding process variations $\boldsymbol{\xi}$ follows a truncated Gaussian mixture distribution (see Appendix A.2.2). We aim to maximize the expected 3-dB bandwidth (BW, in GHz) with probability constraints on the crosstalk (XT, in dB) and the attenuation (α, in dB) of the peak transmission. Therefore, the yield-aware chance-

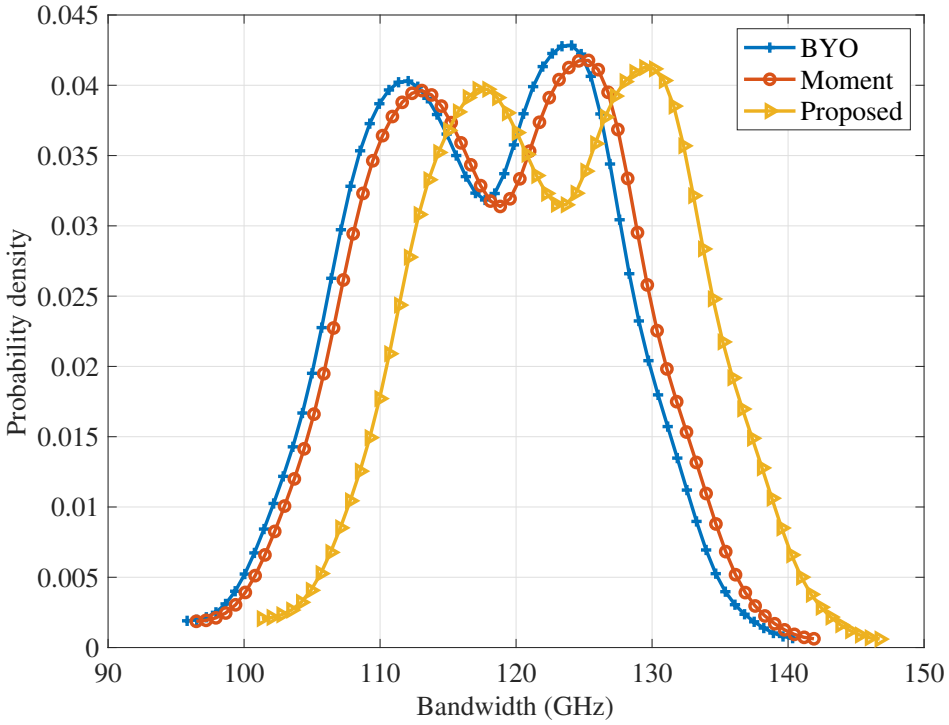

Figure 3.10: The probability density function of the optimized bandwidth of the microring filter by Bayesian yield optimization [104], moment bounding [25] and the proposed PoBO (with $\epsilon = 0.1$).

constrained design is formulated as

$$\begin{aligned} \max_{\mathbf{x}\in\mathbf{X}} \quad & \mathbb{E}_{\boldsymbol{\xi}}[\mathrm{BW}(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\xi}}(\mathrm{XT}(\mathbf{x},\boldsymbol{\xi}) \le \mathrm{XT}_0) \ge 1-\epsilon_1, \\ & \mathbb{P}_{\boldsymbol{\xi}}(\alpha(\mathbf{x},\boldsymbol{\xi}) \le \alpha_0) \ge 1-\epsilon_2. \end{aligned} \tag{3.17}$$

The two risk levels are set to be equal $\epsilon_1 = \epsilon_2 = \epsilon$. The thresholds of the crosstalk (XT_0) and the attenuation (α_0) are -4 dB and 1.6 dB, respectively.

We build three 2nd-order polynomial surrogate models for BW, XT, and α, respectively. We further bound the probabilistic yield constraints via an order-5 optimal polynomial kinship function. The optimized results and comparisons are listed in Table 3.2. It shows that at the same risk level, the proposed PoBO method can achieve larger

bandwidth while meeting the yield requirements and having smaller gaps for the chance constraints. The simulation samples are the ones used for building surrogate models. We list the number of samples to reveal the simulation cost since the simulation time per sample may vary from seconds to hours, which depends on the problem size (small circuits or large circuit) and simulator types (e.g., circuit-level simulation or EM-based PDE simulator). The proposed PoBO requires the same number of simulations as the moment bounding method [25], and both of them require much fewer simulation samples than the Bayesian yield optimization due to the efficient surrogate modeling. Regarding the CPU time of solving the design optimization problem (3.17), the proposed method takes 15.28 s, 18.19 s, and 15.2 s for the three risk levels, respectively. The moment method takes 0.39 s, 2.11 s, and 2.21 s, respectively. Our Kinship-based optimization is slower than the moment-based method because our method uses higher-order polynomials to bound the probabilistic constraints. However, the optimization overhead is negligible compared with the sample simulation time (especially when a PDE-based simulator is employed). The BYO takes even less than 0.1 s in the optimization steps, but it requires a huge number of simulation samples, causing a much larger overall CPU time than chance-constrained optimization. Fig. 3.7 compares the frequency response before and after the yield-aware optimization with $\epsilon = 0.1$. Our PoBO method has a higher expected bandwidth compared with the Bayesian yield optimization, the moment bounding method, and the initial design. Fig. 3.8 further shows the probability density of the optimized bandwidth by different models. It clearly shows that our proposed method produces the highest bandwidth while meeting the yield requirement.

Table 3.3: Optimization Results for Microring Add-drop Filter benchmark

Risk level ϵ	Method	$\mathbb{E}_{\boldsymbol{\xi}}$[BW] (GHz)	Δ_1 (%)	Δ_2 (%)	Yield (%)	Simulation #
0.05	Moment [25]	N/A*	N/A*	N/A*	N/A*	65
	Proposed	**116.85**	**5.26**	**4.84**	99.6	65
0.07	Moment [25]	112.64	7.53	7.42	99.9	65
	Proposed	**120.05**	**7.53**	**6.67**	99.2	65
0.1	Moment [25]	118.47	11.11	10.78	99.7	65
	Proposed	**123.05**	**11.11**	**8.33**	97.5	65
N/A	BYO [104]	117.42	N/A	N/A	95.1	2020

* The algorithm fails with no feasible solution.

3.5.3 Microring Add-Drop Filter

We further consider the design of an optical add-drop filter consisting of three identical silicon microrings coupled in series, as shown in Fig. 3.9. The design variables are the coupling coefficients $\mathbf{x} = [K_1, K_2, K_3, K_4]$ that are to be optimized within the interval of $\mathbf{X} = [0.3, 0.6]^4$. The process variations $\boldsymbol{\xi}$ are described by a truncated Gaussian mixture model (see Appendix A.2.2). The design problem is to maximize the expected 3-dB bandwidth (BW, in GHz) with constraints on the extinction ratio (RE, in dB) of the transmission at the drop port and the roughness (σ_{pass}, in dB) of the passband that takes its standard deviation, formulated as:

$$\begin{aligned} \max_{\mathbf{x}\in\mathbf{X}} \quad & \mathbb{E}_{\boldsymbol{\xi}}[\mathrm{BW}(\mathbf{x}, \boldsymbol{\xi})] \\ \text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\xi}}(\mathrm{RE}(\mathbf{x}, \boldsymbol{\xi}) \geq \mathrm{RE}_0) \geq 1 - \epsilon_1, \\ & \mathbb{P}_{\boldsymbol{\xi}}(\sigma_{\text{pass}}(\mathbf{x}, \boldsymbol{\xi}) \leq \sigma_0) \geq 1 - \epsilon_2. \end{aligned} \tag{3.18}$$

The two risk levels are set to be equal $\epsilon_1 = \epsilon_2 = \epsilon$. The thresholds of the extinction ratio (RE_0) and the roughness of the passband (σ_0) are 20 dB and 0.65 dB, respectively.

Similarly, we build three 2nd-order polynomial surrogate models for BW, RE, and σ_{pass} and bound the chance constraints via an order-5 optimal polynomial kinship func-

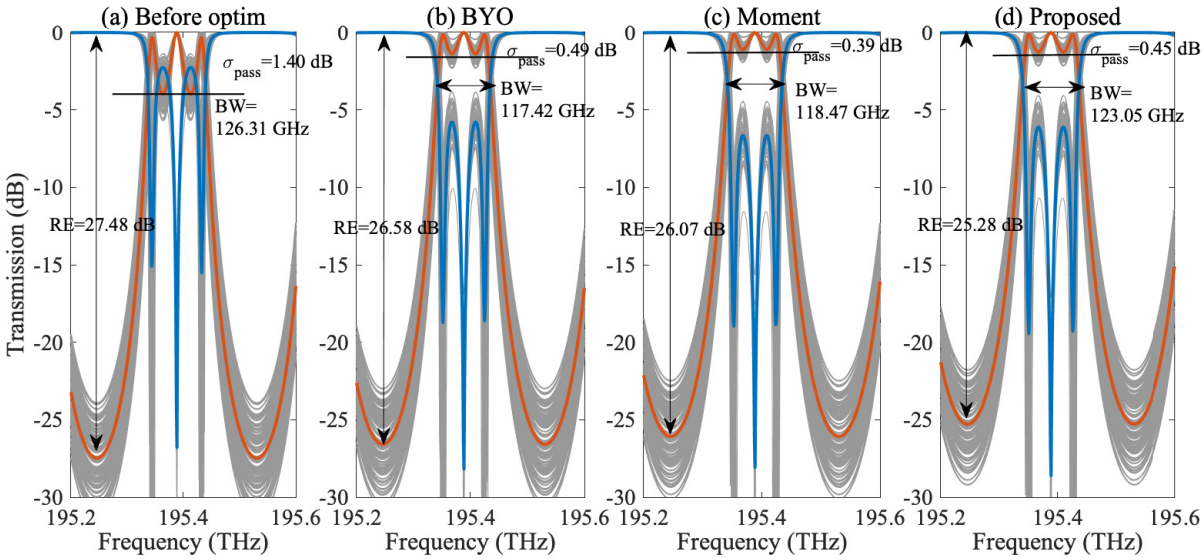

Figure 3.11: The transmission curves of the microring add-drop filter. The grey lines show the performance uncertainties. The orange and blue curves show the transmission rates at the drop and through ports, respectively. The mean values of the extinction ratio, bandwidth, and roughness are denoted as RE, BW, and σ_{pass}, respectively. (a) The initial design (infeasible): $\mathbf{x}$=[0.45, 0.45, 0.45, 0.45]; (b) Design after Bayesian yield optimization [104]: $\mathbf{x}$=[0.5758, 0.3718, 0.3720, 0.5746]; (c) Design with the moment-bounding yield-aware optimization [25]: $\mathbf{x}$=[0.6, 0.3751, 0.3642 0.6]; (d) Design with the proposed PoBO optimization: $\mathbf{x}$=[0.6, 0.3971, 0.3642, 0.6].

tion. The optimized results and comparisons are shown in Table 3.3. The moment bounding [25] fails when $\epsilon = 0.05$ since no feasible solution is found under its over-conservative bounding. For three risk levels, the optimization time for the moment method is N/A (no feasible solution), 5.89 s, 5.5 s, respectively. The proposed method takes 100.58 s, 110.58 s, and 103.74 s, respectively, but the overhead is negligible compared with the simulation cost. The BYO takes less than 0.1 s in the optimization step, but it takes the most overall time due to the high cost of simulating many samples. At all risk levels, the proposed method can achieve larger bandwidth while meeting the yield requirements and having smaller gaps for the chance constraints. As shown in Fig. 3.10, the proposed PoBO has a higher expected bandwidth compared to Bayesian yield optimization and existing yield-aware chance-constrained optimization via moment bounding [25]. Fig. 3.11 shows

the frequency response before and after the yield-aware optimization with $\epsilon = 0.1$. For this microring filter benchmark, we further consider a special case where design objective and constraints are the same quantity. For this case, our method still outperforms others (see the details in Appendix A.2.3).

3.6 Conclusion

This chapter has proposed a novel **Po**lynomial **B**ounding method for chance-constrained yield-aware **O**ptimization (PoBO) of photonic ICs with truncated non-Gaussian correlated uncertainties. In PoBO, we first construct surrogate models with a few simulation samples for the interested quantities based on available uncertainty quantification solvers. To avoid over-conservative design, we have proposed an optimal polynomial kinship function to tightly bound the chance constraints. This bounding method can be efficiently implemented without additional simulations. It also preserves the polynomial form and enables seeking a globally optimal design. The proposed PoBO is verified with a synthetic function, a Mach-Zehnder interferometer, and a microring add-drop filter. In all experiments, the proposed PoBO has achieved the yield requirements, produced tighter bounds on the chance constraints than the state-of-the-art moment bounding method, and led to better design objective performances with a few simulation samples. On the two photonic IC examples, the proposed method has also reduced the simulation samples by 58× and 31× compared with Bayesian yield optimization.

The theoretical and numerical results of this work have laid the foundation of many future topics. Possible extensions of this work include, but are not limited to: (1) improved algorithms to handle many design parameters and process variation, (2) formulations and algorithms to handle joint chance constraints for yield descriptions, (3) PoBO with non-polynomial surrogates. The proposed framework is very generic, and it can also

be employed in other applications beyond EDA, including probabilistic control of energy systems, safety-critical control of autonomous systems, and so forth.

Chapter 4

Distributionally Robust Circuit Optimization under Variation Shifts

4.1 Introduction

In this chapter, we will raise a variation shift issue in circuit modeling and optimization and handle it with the technique of distributionally robust optimization.

In the realm of semiconductor chip design, imperfect nano-fabrications have led to dramatic performance degradation and yield loss [5]. These variations become even more pronounced in emerging computing technologies. The EDA (electronic design automation) community has long been engaged in exploring variation-aware simulation [24,29,42,140,141], modeling [23,142–144], and optimization techniques [102,104,145] for the design and fabrication of integrated circuits, MEMS and photonics.

In most existing approaches, it is assumed that the process variations are described *exactly* by a probability density function (PDF). Under this assumption, variation-aware circuit optimization [25, 30, 102, 104, 145–147] has been formulated as a stochastic optimization problem: the goal is to minimize the expectation value of a cost function subject

to some deterministic or stochastic design constraints. Compared to robust optimization-based design [94,148] that optimizes worst-case circuit performance, stochastic optimization leads to more accurate and less conservative results by accounting for the PDF of process variations. So far, stochastic optimization-based approaches have achieved great success in the EDA field.

In this chapter, we ask a fundamental question: *what if the PDF of process variations is uncertain and/or not exactly known?* This is a critical yet rarely explored question in EDA. In practical EDA flow, the PDF of process variations is normally extracted based on some measurement data of a foundry. As will be explained in Section 4.2.1, the given PDF often differs from the actual one due to the limited size and low quality of measurement data, as well as due to unavoidable errors in the statistical modeling process. Even if the given PDF is accurate enough at the beginning, the actual PDF may still change significantly over time. We call the phenomena of the given PDF differing from the actual one as *variation shifts*. in this chapter, we investigate how to optimize a circuit design under such variation shifts. Definitely, worst-case circuit optimization techniques [94, 148] can still be applied for some cases when variation shift exists and the variation parameters are bounded, at the cost of (probably tremendously) over-conservative design. However, our goal is to develop a rigorous problem formulation and a proof-of-concept solver for shift-aware circuit optimization by considering both the statistical nature and the unknown PDF of process variations.

Main contributions. In this chapter, we investigate the problem formulation, numerical solver, and validation of shift-aware circuit optimization. Our specific contributions include:

- We present, for the first time, a mathematical formulation for the shift-aware circuit optimization problem. Starting with an introduction of the typical sources

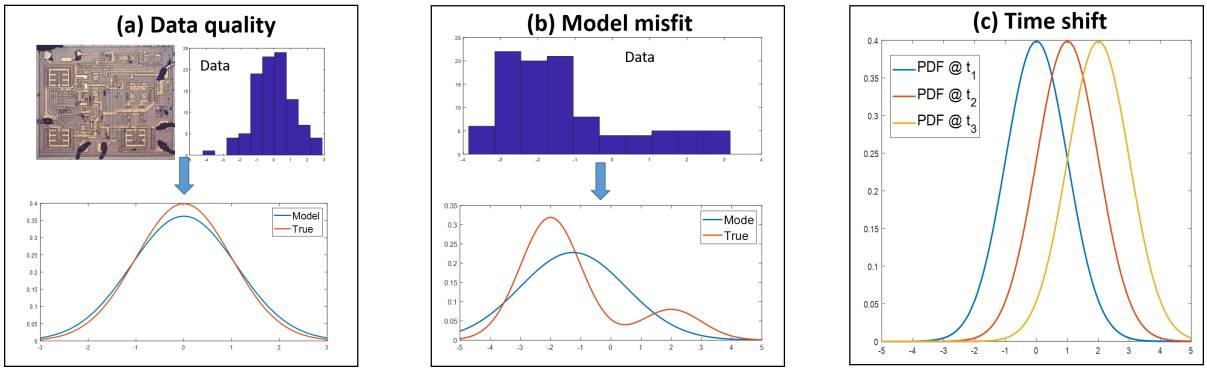

Figure 4.1: Various sources of variation shifts. (a): Insufficient and/or inaccurate measurement data can lead to inaccurate density estimation of process variations. (b): An imperfect chosen (e.g., over-simplified) model can generate a distribution model that is far away from the true one. (3) The distribution of device parameters can shift over time. In all cases, stochastic optimization can produce underperforming results when one uses a fixed PDF model (e.g., a nominal PDF) that differs from the unknown actual one.

of variation shifts in IC design, we formulate shift-aware circuit optimization as a distributionally robust optimization problem.

- We present numerical methods to solve the shift-aware optimization problem. Recognizing the effectiveness of Bayesian optimization in traditional variation-aware IC design optimization [104, 146, 147], we extend its usage to solve our shift-aware circuit optimization problem. Specifically, we employ an uncertainty ball whose radius is defined by φ-divergence to represent the distribution shift of process variations. By leveraging recent advancements of distributionally robust Bayesian optimization (DRBO) [149–151], we can solve our shift-aware circuit optimization problem effectively and efficiently.

- We validate our approach on both photonic and electronic IC benchmarks. Through numerical experiments on two realistic design cases, we demonstrate that the proposed distributionally robust optimization method can maintain excellent perfor-

mance metrics and high yield under various unforeseen potential PDFs of process variations.

We regard this work as a first attempt to address the critical issue of variation shifts. This research could enable a new direction, and motivate the study of numerous shift-aware EDA problems in the future.

4.2 Problem Formulation

4.2.1 Variation Shifts

In classical statistical variation-aware circuit design, process variations are assumed to be described precisely by a reference probability density function (PDF) $\rho_0(\boldsymbol{\xi})$ (named *nominal distribution* in this chapter). Consequently, a circuit optimization framework seeks to minimize the expected value of a cost function $f(\mathbf{x}, \boldsymbol{\xi})$ over a design variable $\mathbf{x}$ within its design space $\mathcal{X}$. This is mathematically expressed as:

$$\min_{\mathbf{x}\in\mathcal{X}} \mathbb{E}_{\rho_0(\boldsymbol{\xi})} \left[f(\mathbf{x}, \boldsymbol{\xi}) \right]. \tag{4.1}$$

A typical variation-aware design workflow consists of three steps. (1) Testing engineers measure some data samples $\{\boldsymbol{\xi}^i\}_{i=1}^M$ of device parameters from specially designed testing circuits. (2) A nominal PDF $\rho_0(\boldsymbol{\xi})$ is extracted from the testing data to describe the process variations. (3) The extracted PDF $\rho_0(\boldsymbol{\xi})$ is then used in an EDA tool to conduct statistical circuit simulation, modeling, and optimization. However, the true PDF of process variations, denoted as $\rho(\boldsymbol{\xi})$, is rarely identical to $\rho_0(\boldsymbol{\xi})$ due to the following reasons.

Poor data quality. Ideally, we can approximate the true PDF $\rho(\boldsymbol{\xi})$ with arbitrarily

high accuracy if we have an infinite number of i.i.d. data samples $\{\boldsymbol{\xi}^i\}$ that precisely follow the distribution $\rho(\boldsymbol{\xi})$. However, in practice, this ideal situation rarely happens since the measurement data samples could be very noisy [26]. Additionally, since fabricating and measuring testing circuits can be costly, we typically only have access to a limited number of data samples, further restricting the modeling accuracy of $\rho_0(\boldsymbol{\xi})$.

Model misfit. In reality, process variations often have a complicated (joint) PDF $\rho(\boldsymbol{\xi})$ (e.g., Gaussian mixture distribution) due to the multi-modal behavior and correlation among random parameters [26, 152]. However, in engineering practice, $\rho_0(\boldsymbol{\xi})$ is often chosen as a trivial distribution (e.g., correlated or independent normal distribution) for simplicity [153, 154]. This inherently introduces a discrepancy between the true and modeled distributions.

Time shift. It is well known that device parameters can shift over time [27, 155–157] due to many factors, such as reliability issues and external environmental impact (e.g., radiation, temperature fluctuations). This phenomenon (including but not limited to the aging effects) implies that even if the initially modeled $\rho_0(\boldsymbol{\xi})$ accurately describes the process variations, its accuracy may decrease as time evolves.

Fig. 4.1 visualizes the mismatch between the modeled density function $\rho_0(\boldsymbol{\xi})$ and true distribution $\rho(\boldsymbol{\xi})$ under the above three scenarios.

4.2.2 Formulation via Distributionally Robust Optimization

Due to the variation shifts, we need to rethink the fundamental problem formulation of variation-aware circuit optimization. As shown in Fig. 4.2 (a), in conventional problem settings, given a particular $\mathbf{x}$, the cost function $\mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\mathbf{x}, \boldsymbol{\xi})]$ can be uniquely determined, allowing the search for its minimum value. However, the exact $\rho(\boldsymbol{\xi})$ is unknown due to variation shifts, which introduces uncertainties to the cost function $\mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\mathbf{x}, \boldsymbol{\xi})]$ as

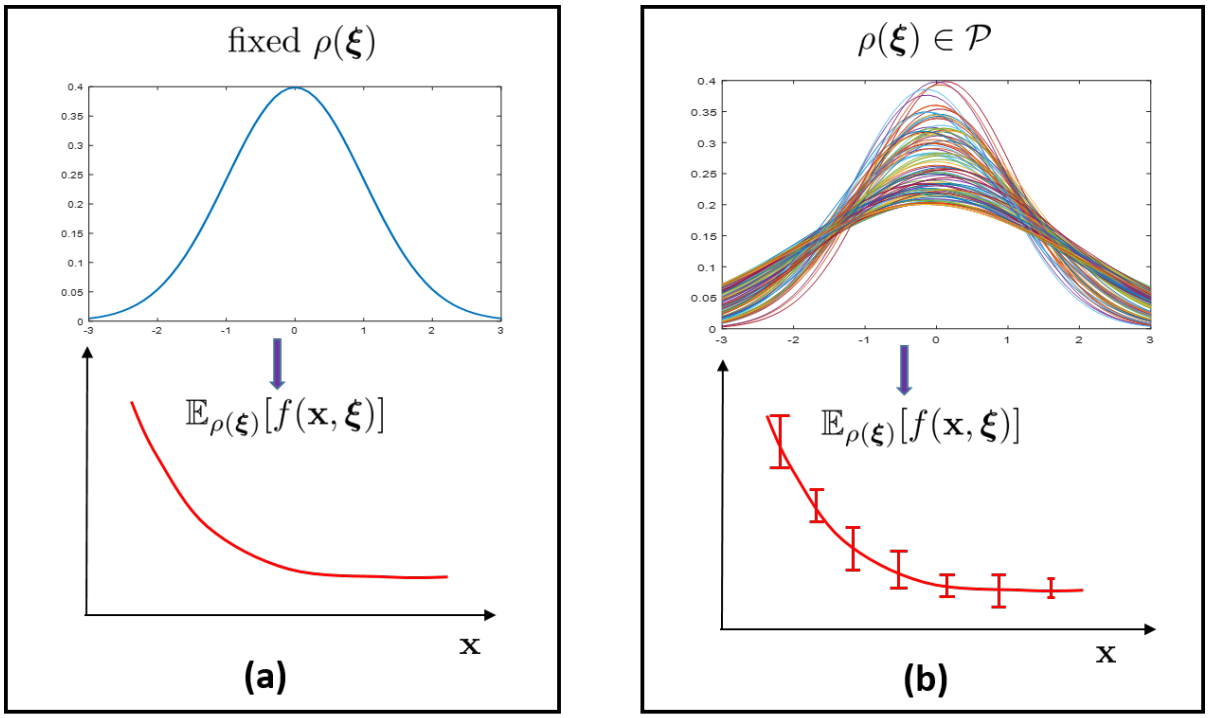

Figure 4.2: Comparison between traditional stochastic circuit optimization and our proposed optimization. (a) In classical variation-aware optimization, $\rho(\boldsymbol{\xi})$ is assumed to be fixed and known, allowing for a unique determination of the cost function $\mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\mathbf{x},\boldsymbol{\xi})]$ for any given design variable $\mathbf{x}$. (b) Under the presence of process variation shifts, the true density $\rho(\boldsymbol{\xi})$ becomes unknown. This consequently introduces uncertainty into the cost function $\mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\mathbf{x},\boldsymbol{\xi})]$, complicating the optimization process.

shown in Fig. 4.2 (b). Even given a specific $\mathbf{x}$, we cannot determine the exact value of this cost function. Therefore, we cannot simply minimize $\mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\mathbf{x},\boldsymbol{\xi})]$.

To address the above challenge, we propose a novel formulation for a generic variation-aware circuit optimization under variation shifts. Specifically, we describe the design optimization as the following *distributionally robust optimization*:

$$\min_{\mathbf{x}\in\mathcal{X}} \sup_{\rho(\boldsymbol{\xi})\in\mathcal{P}} \quad \mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\mathbf{x},\boldsymbol{\xi})], \tag{4.2}$$

where $\mathcal{P}$ is an uncertainty set that includes all possible PDFs of the variation parameters

$\boldsymbol{\xi}$. Intuitively, we will minimize the upper bound of $\mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\mathbf{x}, \boldsymbol{\xi})]$ by considering the variation shifts. In the extreme case where $\mathcal{P}$=$\{\rho_0(\boldsymbol{\xi})\}$, our formulation (4.2) simplifies to a standard stochastic optimization (4.1). Regarding why do we care about the worst-case variations rather than the majority of possible shifted variations? Firstly, technically speaking, it is hard to sample distributions from the uncertainty set $\mathcal{P}$. Secondly, optimization towards the worst-case distributions gives us the robustness to unknowns. Since not all variations within $\mathcal{P}$ can be perfectly known or modeled, designing for worst-case scenarios provides a buffer against unanticipated variations or modeling inaccuracies. Also, the worst-case distribution optimization guarantees circuit performance under various scenarios.

One important case under shifted variations is yield-aware optimization:

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & \mathbb{E}_{\rho(\boldsymbol{\xi})}[f_{\text{obj}}(\mathbf{x}, \boldsymbol{\xi})] \\ \text{s.t.} \quad & \text{Prob}\left[g_k\left(\mathbf{x}, \boldsymbol{\xi}\right) \leq 0, \forall k \in [n]\right] \geq 1 - \tau, \end{aligned} \tag{4.3}$$

where f_{obj} is a design objective, $g_i(\mathbf{x}, \boldsymbol{\xi}) > 0$ describes a violation of the design specification, and τ is a risk tolerance. This formulation aims to optimize the design objective while satisfying a yield requirement [25, 30]. We refer the motivation for such formulation to Chapter 3. To handle the yield constraints, we incorporate these constraints as a penalty term into the objective function:

$$\begin{aligned} f(\mathbf{x}, \boldsymbol{\xi}) &= f_{\text{obj}}(\mathbf{x}, \boldsymbol{\xi}) + \lambda I(\mathbf{x}, \boldsymbol{\xi}) \text{ with} \\ I(\mathbf{x}, \boldsymbol{\xi}) &= \begin{cases} 0, & g_k(\mathbf{x}, \boldsymbol{\xi}) \leq 0, \forall k \in [n], \\ 1, & \text{otherwise.} \end{cases} \end{aligned} \tag{4.4}$$

Here, $I(\mathbf{x}, \boldsymbol{\xi})$ is an indicator function for risk violations and $\lambda \geq 0$ is a parameter for

penalizing the constraint violations. The yield $1-\tau$ is defined as the probability that all risk constraints are met, i.e., $1-\mathbb{E}_{\rho(\xi)}[I(\mathbf{x},\boldsymbol{\xi})]$. The resulting cost function $f(\mathbf{x},\boldsymbol{\xi})$ can then be integrated in the shift-aware optimization defined in (4.2).

4.3 Distributionally Robust Bayesian Optimization Solver

The distributionally robust circuit optimization as defined in (4.2) may be intractable in practice because: (a) the uncertainty set $\mathcal{P}$ may contain an infinite number of PDFs describing process variations; (b) the min-max problem is inherently hard to solve; (c) we do not have an analytical form for $f(\mathbf{x},\boldsymbol{\xi})$ and its simulation could be costly. To tackle these challenges, we will first define the PDF uncertainty set $\mathcal{P}$ appropriately. Subsequently, we utilize distributionally robust Bayesian optimization (DRBO) [149–151], a recently developed technique in the machine learning community, to efficiently solve problem (4.2).

4.3.1 Distribution Uncertainty Set

In our approach, we model the PDF uncertainty set $\mathcal{P}$ as a ball whose center is the nominal distribution $\rho_0(\boldsymbol{\xi})$ (which is often an inaccurate PDF provided by a foundry) and whose radius ε is measured by a distribution divergence $\mathcal{D}$:

$$\mathcal{P} := \mathcal{B}(\rho_0) = \{\rho : \mathcal{D}(\rho_0, \rho) \leq \varepsilon\}. \tag{4.5}$$

Here $\mathcal{D}(\rho_0,\rho)$ measures the difference between $\rho_0(\boldsymbol{\xi})$ and $\rho(\boldsymbol{\xi})$. In practice, we can decide the value of radius ϵ based on estimations of variation shifts, which will be discussed

later.

There are several choices for the divergence $\mathcal{D}$, including maximum mean discrepancy, Wasserstein distance, φ-divergence, etc [158]. Here, we choose the φ-divergence (also called as f-divergence) for its advantages in computational efficiency. Let ρ and ρ_0 be two distributions such that ρ is absolutely continuous with respect to ρ_0, the φ-divergence from ρ to ρ_0 is defined as [159]:

$$\mathcal{D}_\varphi(\rho, \rho_0) \triangleq \mathbb{E}_{\rho_0}\left[\varphi\left(\frac{d\rho}{d\rho_0}(\boldsymbol{\xi})\right)\right], \tag{4.6}$$

where $\varphi : \mathbb{R} \to (-\infty, \infty]$ is a convex, lower semi-continuous function such that is $\varphi(1) = 0$ and $d\rho/d\rho_0$ is a *Radon-Nikodym* derivative. By choosing different functions for φ, Eq. (4.6) can cover many popular divergences, such as χ^2-divergence, total variation distance, and KL-divergence.

Based on the uncertainty set $\mathcal{P}$ defined in (4.6), we can rewrite the distributionally robust optimization problem (4.2) as follows:

$$\min_{\mathbf{x}\in\mathbf{X}} \sup_{\rho(\boldsymbol{\xi})\in\mathcal{B}_\varphi(\rho_0)} \mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\mathbf{x}, \boldsymbol{\xi})]. \tag{4.7}$$

In this chapter, we choose $\varphi(u) = (u-1)^2$ to measure the radius of $\mathcal{P}$ by χ^2-divergence. By applying the theorem 2 outlined below, we can further simplify Problem (4.7) to a single-level optimization problem:

$$\min_{\mathbf{x}\in\mathbf{X}} \ \mathbb{E}_{\rho_0(\boldsymbol{\xi})}[f(\mathbf{x}, \boldsymbol{\xi})] + \sqrt{\varepsilon \cdot \mathrm{Var}_{\rho_0(\boldsymbol{\xi})}[f(\mathbf{x}, \boldsymbol{\xi})]}. \tag{4.8}$$

Note that we can still simplify the min-max optimization if the radius is measured by other divergences within the φ-divergence family.

Theorem 2 (modified from [151]) *Let $\varphi : \mathbb{R} \longrightarrow (-\infty, \infty]$ be a convex lower semi-continuous mapping such that $\varphi(1) = 0$. Its convex conjugate $\varphi^\star$ is defined as $\varphi^\star(u) = \sup_{v\in\mathbb{R}} uv - \varphi(v)$. Let f be measurable and bounded. For any $\varepsilon > 0$, it holds that*

$$\min_{\mathbf{x}\in\mathbf{X}} \sup_{\rho(\boldsymbol{\xi})\in\mathcal{B}_\varphi(\rho_0)} \mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\mathbf{x},\boldsymbol{\xi})] =$$
$$\min_{\mathbf{x}\in\mathbf{X},\tau\geq 0, b\in\mathbb{R}} \left(-b + \tau\varepsilon + \tau\mathbb{E}_{\rho_0}\left[\varphi^\star\left(\frac{b + f(\mathbf{x},\boldsymbol{\xi})}{\tau}\right)\right]\right).$$

This theorem simplifies the two-level min-max problem into one simple minimization one with two additional 1-dimensional variables τ and b. When $\varphi(u) = (u-1)^2$, which corresponds to the χ^2 divergence, we have $\varphi^\star(u) = \frac{u^2}{4} + u$. Plugging this into the above equation and solving the 1-dimensional optimization problems for τ and b, which both have analytical solutions, yields the reformulated problem in Eq. (4.8). Compared to the original theorem in Ref [151], we make slight modifications by switching the min-max order of the original problem and allowing $\boldsymbol{\xi}$ to be a higher-dimensional variable. This change does not affect the proof sketch.

4.3.2 DRBO Workflow

Next, we explain how to solve (4.8) via DRBO with a few circuit simulation samples. Similar to a standard Bayesian optimization (BO), DRBO sequentially builds a probabilistic surrogate model of $f(\mathbf{x},\boldsymbol{\xi})$ and explores the design space by minimizing an acquisition function. The overall DRBO algorithm is summarized in Algorithm 4, and the key steps are explained below.

- **Step 1**. Build a probabilistic surrogate model of $f(\mathbf{x},\boldsymbol{\xi})$. Here we choose a Gaussian process model [160], which has been widely used in BO-based circuit optimization.

- **Step 2**. Minimize the acquisition function $A(\mathbf{x})$:

$$\mathbf{x} = \underset{\mathbf{x}\in\mathbf{X}}{\operatorname{argmin}}\, A(\mathbf{x}). \tag{4.9}$$

The acquisition function serves as a important role in guiding the exploration of design space by optimizing some informative metrics, such as expectation improvement and lower confidence bound (LCB). In this chapter, we use the LCB metric to define $A(\mathbf{x})$. This approach allows us to effectively balance between exploiting promising current designs and exploring the design space characterized by significant model uncertainty:

$$\begin{aligned} A(\mathbf{x}) := \frac{1}{L}\sum_{l=1}^{L}[\mu(\mathbf{x},\boldsymbol{\xi}^l) - \sqrt{\beta}\sigma(\mathbf{x},\boldsymbol{\xi}^l)]+ \\ \sqrt{\frac{\varepsilon}{L}\sum_{l=1}^{L}\left(\mu(\mathbf{x},\boldsymbol{\xi}^l) - \bar{\mu}\right)^2}, \end{aligned} \tag{4.10}$$

where $\mu(\mathbf{x},\boldsymbol{\xi})$ and $\sigma(\mathbf{x},\boldsymbol{\xi})$ represent the predictive mean and standard deviation offered by the probabilistic surrogate model, and β is a balancing factor. A finite number of samples $\{\boldsymbol{\xi}^l\}_{l=1}^{L}$ sampled from ρ_0 are used to estimate the performance mean and variance over the nominal variation and $\bar{\mu} = \frac{1}{L}\sum_{l=1}^{L}\mu(\mathbf{x},\boldsymbol{\xi}^l)$ denotes the performance mean.

- **Step 3**. If convergence is not achieved, we draw a new sample from ρ_0 and augment the existing sample set. Then the algorithm returns to Step 1.

Remarks. Compared with standard BO, the DRBO algorithm aims to find a robust solution under shifted variations by penalizing an additional term associated with the variance. In other words, without the second term in (4.8) and consequently in (4.10), DRBO would simply reduce to standard BO.

Algorithm 4 Overall DRBO algorithm

Input: Initial sample set $\mathcal{S}_0 = \{\mathbf{x}^i, \boldsymbol{\xi}^i, f(\mathbf{x}^i, \boldsymbol{\xi}^i)\}_{i=1}^M$, nominal PDF of variations ρ_0, uncertainty ball radius ε, maximum iteration T

Output: The optimal circuit design $\mathbf{x}^\star$ for Problem (4.8)

1: **for** $t = 1, 2, ..., T$ **do**
2: Construct a probabilistic surrogate model based on $\mathcal{S}_{t-1}$
3: Solve the next query point $\mathbf{x}_t$ via (4.9) with acquisition function (4.10)
4: Sample variation $\boldsymbol{\xi}_t \sim \rho_0$ and simulate $f(\mathbf{x}_t, \boldsymbol{\xi}_t)$
5: Augment data set $\mathcal{S}_t \leftarrow \mathcal{S}_{t-1} \cup (\mathbf{x}_t, \boldsymbol{\xi}_t, f(\mathbf{x}_t, \boldsymbol{\xi}_t))$
6: **end for**
7: Return the optimal design $\mathbf{x}^\star$

Despite many methods to construct the distribution set [158], we choose the φ-divergence since it is particularly suitable for circuit optimization. Firstly, φ-divergence covers many commonly used divergence measurements and thus effectively describes the shifted variations. Secondly, both the design variables and the variations of analog circuits are continuous, while some methods require us to discretize them [149, 150]. More importantly, the min-max problem can be reformulated into a single-level minimization problem, offering significant computational advantages. In circuit optimization, computational cost is often a primary concern. The variations often have high dimensionality, and without the reformulation, the computational burden of a two-level min-max optimization can grow significantly as the dimension of variations increases. However, through the use of φ-divergence for modeling and the subsequent reformulation, the computational cost of DRBO is almost the same as a standard Bayesian optimization.

4.4 Implementation Details

In this section, we will show some implementation details of the DRBO algorithm.

Probabilistic surrogate model. In this chapter, we choose the commonly used Gaussian processes (GP) as the probabilistic surrogate model. For simplicity, we denote

$\boldsymbol{\theta} = (\mathbf{x}, \boldsymbol{\xi})$. Let $\boldsymbol{\Theta} = \{\boldsymbol{\theta}^i\}_{i=1}^M$ be a set of training samples, and let $\mathbf{y} = \{f(\boldsymbol{\theta}^i)\}_{i=1}^M$ be their simulation outputs. Given a pre-specified prior mean $m(\boldsymbol{\theta})$ and a kernel function $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$, a GP model assumes that the output $\mathbf{y}$ follow a Gaussian distribution:

$$\text{Prob}(\mathbf{y}) \sim \mathcal{N}(\mathbf{y}|\mathbf{m}, \mathbf{K}), \tag{4.11}$$

where $\mathbf{m} \in \mathbb{R}^M$ is a mean vector, with the i-th element being $m(\boldsymbol{\theta}^i)$ and $\mathbf{K} = \mathbf{k}(\boldsymbol{\Theta}, \boldsymbol{\Theta}) \in \mathbb{R}^{M \times M}$ is a covariance matrix with the (i, j)-th element being $k(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)$. For a new data $\boldsymbol{\theta}'$, we can predict its posterior mean and variance from the GP model as follows:

$$\begin{cases} \mu(\boldsymbol{\theta}') = \mathbf{k}(\boldsymbol{\theta}', \boldsymbol{\Theta})^T \mathbf{K}^{-1} \mathbf{y} \\ \sigma^2(\boldsymbol{\theta}') = k(\boldsymbol{\theta}', \boldsymbol{\theta}') - \mathbf{k}(\boldsymbol{\theta}', \boldsymbol{\Theta})^T \mathbf{K}^{-1} \mathbf{k}(\boldsymbol{\Theta}, \boldsymbol{\theta}'), \end{cases} \tag{4.12}$$

In our cases, we choose the prior mean as $m(\boldsymbol{\theta}) = 0$ and the kernel as a Matern function. The hyperparameters in the GP model are optimized by maximizing the log marginal likelihood.

Improved modeling of the cost function. In certain situations, after applying a penalty term for yield constraints as shown in (4.4), the cost function may become non-smooth, introducing additional challenges for Gaussian process modeling. To address this issue, we employ two distinct GP models to separately estimate circuit performance and feasibility, with the feasibility being estimated by a Gaussian process classifier [161]. Depending on the target applications, a variety of surrogate models, such as advanced variants of Gaussian process [116], Bayesian neural networks [147], and others, could potentially be leveraged to enhance modeling accuracy and achieve superior design results.

Stop criteria. Indeed, in our formulation, choosing the optimal $\mathbf{x}$ after multiple iterations of sampling is not a trivial task. In a much simpler optimization setting,

such as $\min_{\mathbf{x}} f(\mathbf{x})$, we can take the optimal design by $\mathbf{x}^\star = \operatorname{argmin} f(\mathbf{x})$. However, our cost function involves an expectation over the variations, specifically $\mathbb{E}_{\rho_0(\boldsymbol{\xi})}[f(\mathbf{x}, \boldsymbol{\xi})]$ as presented in (4.8). Estimating this expectation through simulating $f(\cdot)$ would be computationally burdensome. To address this, we select $\mathbf{x}$ with the minimal value of (4.8) according to the posterior distribution of $\hat{f}(\mathbf{x}, \boldsymbol{\xi})$. This is a common strategy in such contexts [162, 163]. Beyond the optimal design selection, one strategy for early stop could be that the algorithm returns a stable solution after several consecutive iterations.

Minimizing the acquisition function. Many numerical optimization algorithms and software, such as [164], can be employed to minimize the acquisition function (4.9). In our implementation, we simply adopt another Bayesian optimization as the optimizer.

In Line 4 of Algorithm 4, only one variation sample is drawn from the nominal PDF $\rho_0(\boldsymbol{\xi})$ under the newly determined design $\mathbf{x}_t$. In the implementation, we found that a batch-wise strategy of sampling variations can accelerate algorithm convergence. This strategy groups multiple variation samples with the newly acquired design sample, which helps the GP model to model the cost functional around the new design. Note that the grouping strategy is different from batch-wise Bayesian optimization, where a batch of design variables is determined from the acquisition function. A batch-wise version in solving $\mathbf{x}$ could potentially improve the DRBO efficiency and be of independent interest.

Given the stop criteria of selecting the optimal $\mathbf{x}$ based on the posterior distribution, an augmented sampling set improves the estimation of this distribution, which in turn assists the algorithm in identifying the optimal design. The sample set can be augmented either by grouping more variation samples or by conducting more rounds of iterations. Further investigation is needed to achieve a good balance between these two strategies in terms of computational efficiency.

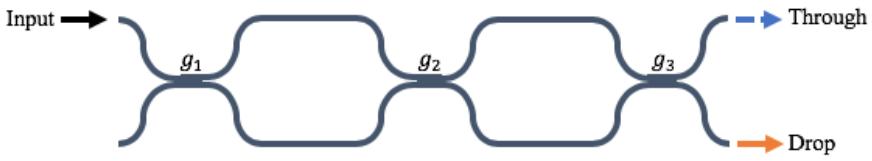

Figure 4.3: The schematic of a third-order Mach-Zehnder interferometer.

4.5 Numerical Results

In this section, we apply the DRBO algorithm in the yield-aware optimization of a photonic IC and an electronics IC and show the robustness of the proposed design under the shift variations.

Baseline: We compare the proposed DRBO algorithm with a standard LCB method [165], which serves as our baseline. LCB method assumes that the process variation follows a fixed distribution $\rho_0(\boldsymbol{\xi})$, whose acquisition function (4.10) degenerates to $A(\mathbf{x}) = \frac{1}{L}\sum_{l=1}^{L}[\mu(\mathbf{x}, \boldsymbol{\xi}^l) - \sqrt{\beta}\sigma(\mathbf{x}, \boldsymbol{\xi}^l)]$. Therefore, it does not consider the variation shifts and thus operates as a standard Bayesian optimization method. It can also be viewed as a DRBO with $\varepsilon = 0$. Specifically, this is exactly what existing Bayesian optimization solvers do in statistical circuit design optimization.

Algorithm evaluation. In this comparison, both DRBO and LCB are implemented with the same nominal variation distribution $\rho_0(\boldsymbol{\xi})$ and with the same initial setup to search optimal designs. To evaluate the robustness of these designs against distribution shifts, we then test the resulting designs under various unforeseen variation distributions.

As previously discussed in Section 4.2.1, we have introduced three scenarios to demonstrate practical variation shifts. For the photonics benchmark, we assume that the variation shifts arise due to *time shift*. In the electronic IC benchmark, we assume that the variation shifts arise from *poor data quality* and *model misfit*, for illustrative purposes. It is critical to note that in realistic applications, it is highly probable that all three

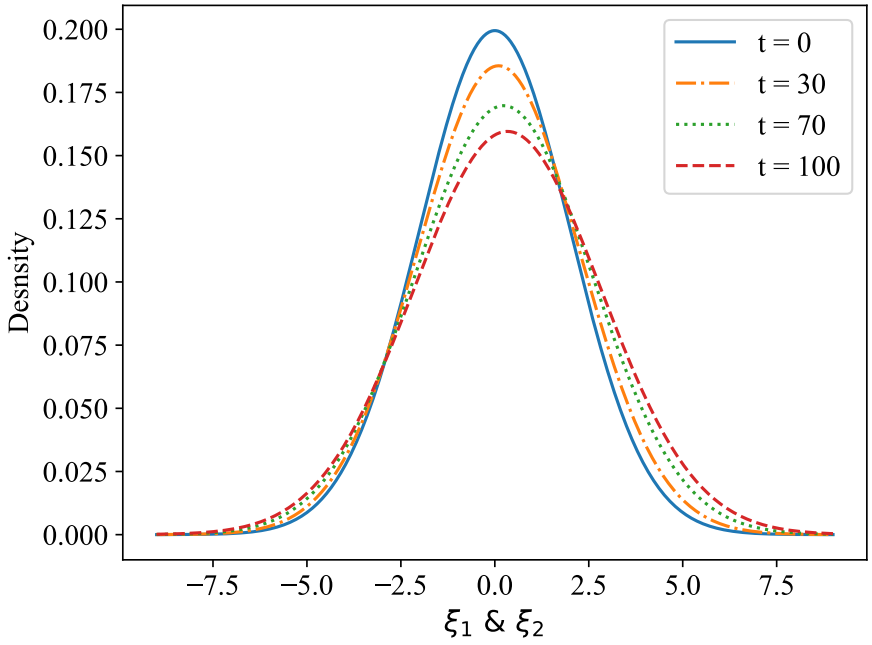

Figure 4.4: Examples of PDFs for the shifted (testing) variation distributions at different time steps, changing from the nominal distribution at $t = 0$ to a highly shifted one at $t = 100$.

scenarios would coexist.

4.5.1 Photonics IC: Mach-Zehnder Interferometer

Setup. We consider a third-order Mach-Zehnder interferometer (MZI) which consists of three port coupling and two arms, as shown in Fig. 4.3. By fixing the first gap parameter, we design the other two gap parameters $\mathbf{x} = [g_1, g_2] \in [100, 300]^2$ nm of the two arms under variations $\boldsymbol{\xi} \in \mathbb{R}^2$. We aim to maximize the expected 3-dB bandwidth (BW, in GHz) with risk constraints on the crosstalk (XT, in dB) and the attenuation (α, in dB) of the peak transmission. The design objective is defined as $f(\mathbf{x}, \boldsymbol{\xi}) = -\mathrm{BW}(\mathbf{x}, \boldsymbol{\xi}) + \lambda I(\mathbf{x}, \boldsymbol{\xi})$

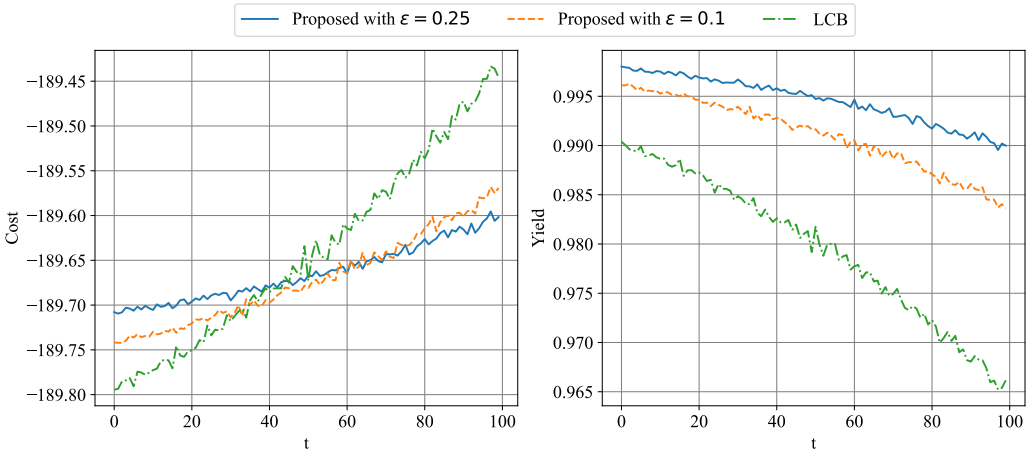

Figure 4.5: The cost function and yield estimated by 10^4 simulated samples during the Bayesian optimization iterations in optimizing the MZI. Regarding the cost function $f(\mathbf{x}, \boldsymbol{\xi})$, the LCB outperforms the DRBO ones when t is relatively small. However, as t increases, the DRBO methods begin to achieve better performance. Regarding the yield, the DRBO method with a larger ε consistently achieves the highest yield in this case. Because it is more conservative in considering the worst case of variation shifts.

with

$$I(\mathbf{x}, \boldsymbol{\xi}) = \begin{cases} 0, & \mathrm{XT}(\mathbf{x}, \boldsymbol{\xi}) \leq \mathrm{XT}_0, \alpha(\mathbf{x}, \boldsymbol{\xi}) \leq \alpha_0 \\ 1, & \text{otherwise.} \end{cases} \tag{4.13}$$

Here, we aim to evaluate the design robustness under the time shift of the variation distribution. Without loss of generalization, we assume the PDFs of the two variations follow the same time-dependent Gaussian distribution

$$\rho(\xi_1(t)) = \rho(\xi_2(t)) \sim \mathcal{N}(\mu(t), \sigma(t)) \text{ with}$$
$$\mu(t) = t/300, \ \sigma(t) = 2 + 0.005t.$$

The PDFs of the resulting time-shift variations are shown in Fig. 4.4. Specifically, the nominal distribution ρ_0 corresponds to the variation PDF at $t = 0$.

For this benchmark, which involves two-dimensional design and variation variables,

we begin by randomly selecting an initial sample set of 200 pairs of $(\mathbf{x}, \boldsymbol{\xi})$. Here, $\mathbf{x}$ is uniformly sampled from the design variable range, and $\boldsymbol{\xi}$ is sampled from the nominal distribution. We conduct experiments using five different sets of initial samples for all algorithms, reporting the mean and standard deviation of the results. To accelerate the learning process, after determining the next design in an iteration, we use the grouping strategy to pair this design with a batch of 5 variation samples from the nominal distribution to augment the sample set. For all algorithms, we use $L = 300$ samples in evaluating the acquisition function and set the maximum iteration count $T = 150$ as the stopping criterion.

Results. We report the cost $f(\mathbf{x}, \boldsymbol{\xi})$ and yield of each iteration under shifted variations in Fig. 4.5. In terms of evaluation, a lower cost function is more desirable, while a higher yield is better. Specifically, we compare the performance among the LCB and the DRBO algorithm, using uncertainty set radius of $\varepsilon = 0.25$ and $\varepsilon = 0.1$. At each time step t, we obtain a design (not a query point) based on the model's posterior distribution and evaluate its performance and yield under the shifted variation distribution. The results at specific time points are detailed in Table 4.1. Initially, at $t = 0$, the LCB outperforms the DRBO since the testing (true) distribution is exactly the nominal one. However, as t increases, and the testing distribution begins to diverge from the nominal one, LCB begins to degrade, eventually becoming worse than DRBO after $t > 30$. In contrast, the DRBO algorithms demonstrate increased robustness to variation shifts over time. Specifically, when the shift is relatively limited (e.g., when $t < 60$), the DRBO with $\varepsilon = 0.1$ performs better than the one with $\varepsilon = 0.25$ since a larger uncertainty set leads to a more conservative design optimization. However, when the shift becomes more pronounced, a larger ε shows beneficial.

Parameter Analysis. We perform an analysis of the sample size L used in evaluating the acquisition function (4.10). For each setting, we present the mean and standard

Table 4.1: Performance of the designs in the MZI benchmark tested under variations at different t.

Time step	Method	Cost function	Yield (%)
t = 0 (Nominal)	Proposed ($\varepsilon = 0.25$)	-189.69±0.099	99.9±0.11
	Proposed ($\varepsilon = 0.10$)	-189.74±0.085	99.6±0.36
	LCB [165]	**-189.80±0.015**	99.1±0.41
t = 30	Proposed ($\varepsilon = 0.25$)	-189.68±0.087	99.7±0.21
	Proposed ($\varepsilon = 0.10$)	**-189.71±0.049**	99.4±0.64
	LCB [165]	-189.71±0.044	98.6±0.61
t = 70	**Proposed ($\varepsilon = 0.25$)**	**-189.65±0.069**	99.5±0.35
	Proposed ($\varepsilon = 0.10$)	-189.65±0.037	98.9±0.96
	LCB [165]	-189.57±0.094	97.5±0.92
t = 100	**Proposed ($\varepsilon = 0.25$)**	**-189.61±0.059**	99.3±0.57
	Proposed ($\varepsilon = 0.10$)	-189.57±0.093	98.3±1.45
	LCB [165]	-189.45±0.124	96.6±1.17

deviation of five repeated experiments with different initial samples. Given that we use a finite number of samples to estimate the nominal distribution, more samples from the distribution lead to a more accurate estimation, and consequently, more robust optimization and design. As shown in Table 4.2, with the increase of L, the design performance generally improves, evidenced by a lower mean and smaller standard deviation in the cost function $f(\mathbf{x}, \boldsymbol{\xi})$.

However, increasing the number of samples L in estimating the mean and standard deviation inherently takes more computational time, leading to a trade-off between computational time and robustness. In the benchmark presented, the problem size is relatively small, so the inference time in a GP model can be almost ignored. However, for higher-dimensional problems, the computational cost could become a significant factor. There are also many potential accelerations towards the DRBO solvers, such as parallelizing the optimization of the acquisition function by starting from different initial points, using the strategy to augment the sample set as previously discussed, and so forth.

Table 4.2: Resulting cost function of DRBO with $\varepsilon = 0.25$ when using different L in optimizing the MZI.

L	$t = 0$	t = 30	t = 70	t = 100
100	-189.68±0.104	-189.65±0.093	-189.58±0.135	-189.52±0.161
200	-189.70±0.058	-189.67±0.039	-189.62±0.065	-189.57±0.125
300	-189.69±0.094	-189.68±0.087	-189.65±0.068	-189.61±0.056
400	-189.70±0.085	-189.69±0.070	-189.66±0.051	-189.62±0.026
500	**-189.75±0.038**	**-189.72±0.021**	**-189.68±0.016**	**-189.62±0.036**

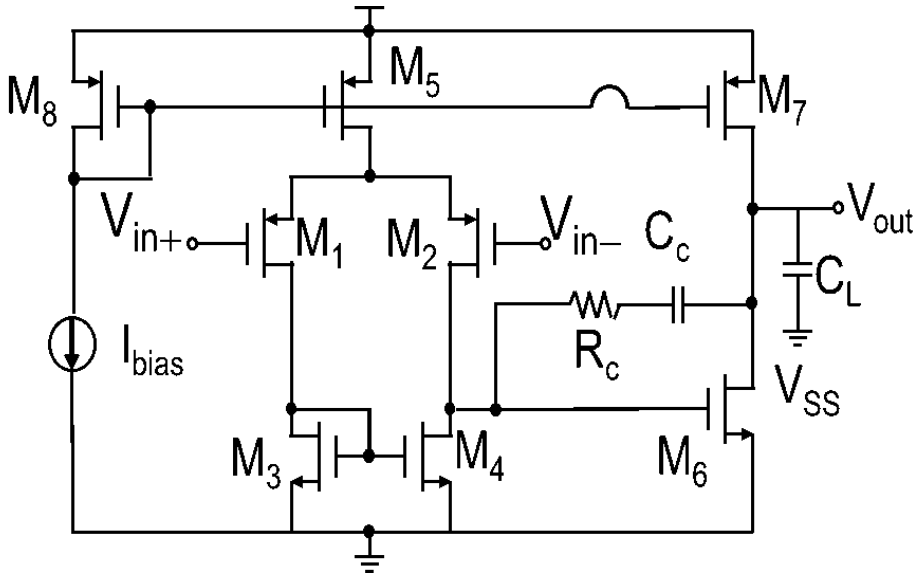

Figure 4.6: The schematic of a two-stage amplifier.

4.5.2 Electronics IC: Two-stage Amplifier

Setup. We consider a two-stage operational amplifier (Op-Amp) whose schematic is shown in Fig. 4.6. Let W_i denote the width (in μm) of the transistor M_i. To reduce the systematic offset voltage, we enforce $W_1 = W_2$, $W_3 = W_4$, and set a ratio $R = W_7/W_5 = W_6/2W_3$. We consider $\mathbf{x} = [W_1, W_6, W_7, R]$ as the design variables for optimization. However, due to the process variations in manufacturing, the enforced equalities may not hold true. Consequently, we consider $\boldsymbol{\xi} = [\epsilon_2, \epsilon_4]$ as the process variations, where $W_2 = (1 + \epsilon_2)W_1$ and $W_4 = (1 + \epsilon_4)W_3$. These variations have been

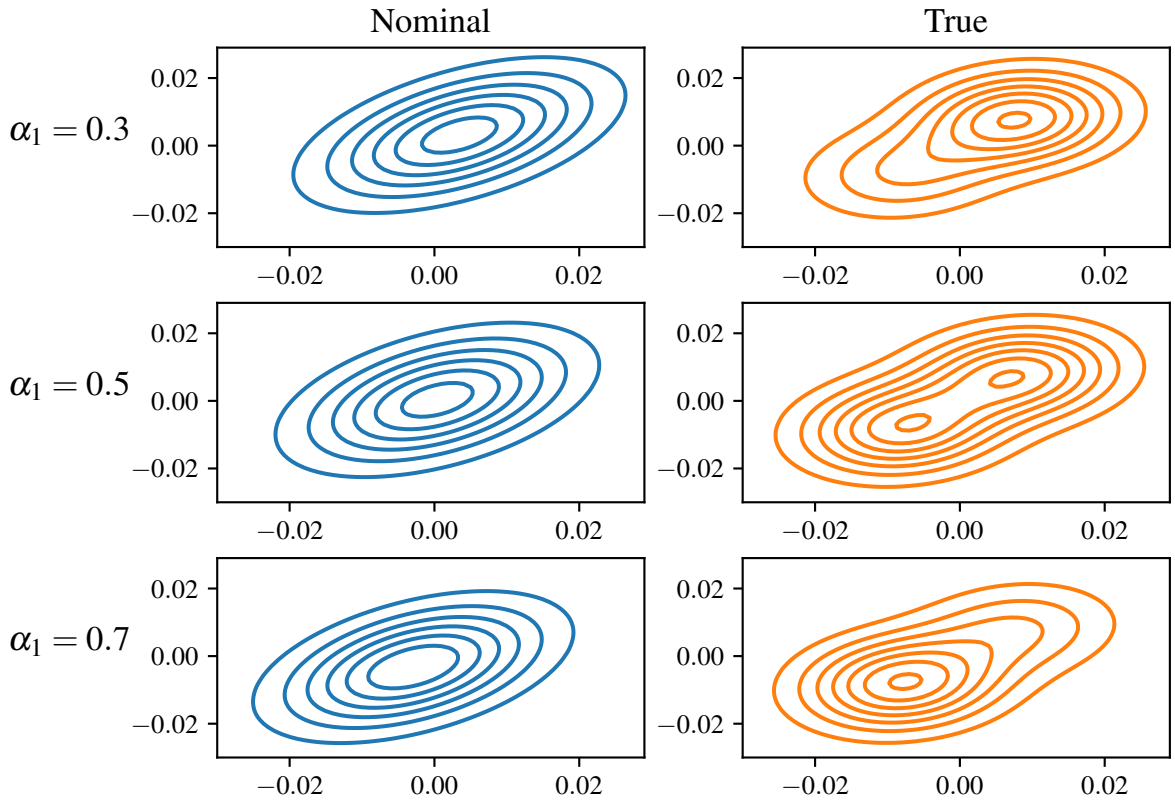

Figure 4.7: PDFs of the nominal Gaussian distribution $\rho_0(\boldsymbol{\xi})$ and the true Gaussian mixture distribution $\rho(\boldsymbol{\xi})$. We consider three cases where the true variations is a mixture of two Gaussian distributions with $\alpha_1 = 0.3, 0.5$, and 0.7.

found to significantly impact the circuit's performance [166]. Here we aim to minimize the power (in mw) while having risk constraints on the gain (in dB), unity gain frequency (UGF, in MHZ), and phase margin (PM, in degree). The design objective is defined as: $f(\mathbf{x}, \boldsymbol{\xi}) = -\text{power}(\mathbf{x}, \boldsymbol{\xi}) + \lambda I(\mathbf{x}, \boldsymbol{\xi})$ where $I(\mathbf{x}, \boldsymbol{\xi}) = 0$ when $\text{gain}(\mathbf{x}, \boldsymbol{\xi}) \geq 30$, $\text{UGF}(\mathbf{x}, \boldsymbol{\xi}) \geq 120$ and $\text{PM}(\mathbf{x}, \boldsymbol{\xi}) \geq 60$ are satisfied, otherwise $I(\mathbf{x}, \boldsymbol{\xi}) = 1$.

We assume that the true parameter distribution $\rho(\boldsymbol{\xi})$ follows a (multivariate) Gaussian mixture model (GMM), which is a weighted sum of multiple (multivariate) Gaussian

distributions:

$$\rho(\boldsymbol{\xi}) = \sum_{i=1}^{n} \alpha_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad \text{with} \quad \sum_{i=1}^{n} \alpha_i = 1.$$

In this study, we set $n = 2$. The variables $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ denote the means of the two Gaussian distributions, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ denote their covariance matrices, and α_1 and α_2 represent their respective mixing weights. Specifically, we set $\boldsymbol{\mu}_1(\boldsymbol{\xi}) = -\boldsymbol{\mu}_2(\boldsymbol{\xi}) = [0.008, 0.008]$ and $\boldsymbol{\Sigma}_1(\boldsymbol{\xi}) = \boldsymbol{\Sigma}_2(\boldsymbol{\xi}) = 10^{-2} \begin{bmatrix} 0.8 & 0.1 \\ 0.1 & 0.8 \end{bmatrix}$.

To simulate variation shifts due to model misfit and poor data quality, we draw 500 noisy samples from the true GMM $\rho(\boldsymbol{\xi})$ and fit a single Gaussian distribution to these samples. This fitted Gaussian serves as our nominal distribution $\rho_0(\boldsymbol{\xi})$. The poor data quality is represented by adding noise to the samples from the true distribution: $\boldsymbol{\xi}_{\text{noisy}} = \boldsymbol{\xi} + \boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, 10^{-4}\mathbf{I})$. In Fig. 4.7, we can clearly see the divergence between $\rho_0(\boldsymbol{\xi})$ and $\rho(\boldsymbol{\xi})$ under different mixing weights α_1.

Similar to the previous MZI example, for this benchmark involving four design variables and two process variation variables, we randomly select 500 pairs of $(\mathbf{x}, \boldsymbol{\xi})$ as the initial samples. During the Bayesian iterations, we group each query point with 10 $\boldsymbol{\xi}$. We repeat the experiments using five different sets of initial samples for all algorithms, subsequently reporting the mean and standard deviation of the results. We set the maximum iteration $T = 160$, and use $L = 300$ for (4.10). To further stabilize the design solution in this benchmark, we take the final optimal design as the output design with the lowest cost function from the last five Bayesian optimization iterations.

Results. We evaluate the design performance of both LCB and DRBO with $\varepsilon = 0.05$ under three specific cases of variation shifts, namely, where the true distributions have different mixing weights $\alpha_1 = 0.3, 0.5$, and 0.7. Additionally, we also test the design

performance under the nominal distributions as a reference. For each design, its cost function and yield are estimated based on 5×10^3 simulated samples. The resulting design performance and comparisons are comprehensively outlined in Table 4.3. When we evaluate the designs under the nominal distributions, both the DRBO and LCB methods achieve comparable results, with a similar cost function and yield. This shows the effective optimization of two methods in the absence of variation shift. However, when we evaluate the designs under the true distributions, the DRBO solution demonstrates significantly better robustness under the shifts. This is evidenced by the lower cost function and higher yield associated with the DRBO solution. The observations are consistent across all three true distribution cases.

Specifically, when compared to the evaluations under nominal evaluation, the performance of the LCB design decreases significantly as evidenced by an increase in the cost function and a decrease in yield. These results show that a design quality could be highly sensitive to the variation shifts, highlighting the importance of the proposed shift-aware optimization formulation and the DRBO solver.

Parameter Analysis. Here, we analyze the impact of the radius ε of the uncertainty ball under the true variations where $\alpha_1 = 0.3$. The radius ε represents our estimation of the potential variation shifts. In this testing case, the true χ^2-divergence between true distribution and nominal distribution is approximately 0.1. Table 4.4 depicts the results obtained from running the DRBO with different values of ε. As we can see, with the increase in ε, the obtained cost function first decreases and then increases. The optimal cost function is achieved when the value of ε is closest to the actual divergence. It indicates that in real-world applications, users should select the value of ε based on their estimation of variation shifts. While we observe a consistent improvement in yield as ε increases in this case, the monotonic relationship between the radius ε and the yield is not strictly guaranteed in our yield-aware optimization problem. To prioritize yield

Table 4.3: Performance of the designs in the two-stage amplifier benchmark tested under different distributions.

α_1	Testing $\rho(\boldsymbol{\xi})$	Method	Cost function	Yield (%)
0.3	Nominal	Proposed	0.2088±0.0037	93.32±1.36
		LCB [165]	0.2124±0.0034	91.73±2.02
	True	Proposed	**0.2195±0.0043**	91.63±2.02
		LCB [165]	0.2282±0.0159	86.63±4.01
0.5	Nominal	Proposed	0.2142±0.0035	91.91±2.91
		LCB [165]	0.2182±0.0047	90.70±2.20
	True	Proposed	**0.2191±0.0039**	91.36±3.04
		LCB [165]	0.2290±0.0115	87.78±1.47
0.7	Nominal	Proposed	0.2241±0.0067	88.09±0.98
		LCB [165]	0.2256±0.0112	88.07±3.19
	True	Proposed	**0.2231±0.0104**	88.13±1.76
		LCB [165]	0.2287±0.0144	87.11±3.72

optimization, we can simply assign a large value for the penalty factor λ. Meanwhile, other variants of distributionally robust yield optimization formulations deserve further investigation.

4.6 Conclusion

In practical IC design, we only have limited and inexact statistical knowledge of process variations. Therefore the actual PDF of process variations can be quite uncertain, and it often differs from the given statistical model. In this chapter, we outlined three possible scenarios of distributional shift - poor data quality, model misfit, and time shift - which are likely to occur concurrently in real-world applications. To find a design that is robust against such variation shifts, we formulated a novel shift-aware circuit optimization problem as a distributionally robust optimization problem. This formulation optimizes the circuit design when the actual variation distribution is different from the nominal

Table 4.4: Parameter analysis on the radius ε of the uncertainty ball under true distribution with $\alpha_1 = 0.3$.

Radius ε	Cost function	Yield (%)
0.15	0.2287±0.0144	92.71±0.94
0.10	**0.2181±0.0044**	92.54±1.61
0.05	0.2195±0.0043	91.63±2.02
0.01	0.2257± 0.0057	88.71±1.17
0 (LCB)	0.2282±0.0159	86.63±4.01

one. With the appropriate modeling of the uncertainty set for variation distributions, we effectively and efficiently solve the proposed shift-aware circuit optimization via a distributionally robust Bayesian optimization (DRBO) approach. We validate the proposed method using two IC benchmarks - a photonic IC and an electronic IC. Our approach successfully solves out designs that demonstrate greater robustness against variation shifts, whereas traditional stochastic optimization methods significantly underperform under the variation shifts.

Possible extensions of this work include, but are not limited to: (i) heuristics or theory to determine the radius of the uncertainty ball, (ii) techniques to handle high-dimensional design variables and process variations, (iii) further acceleration of the distributionally robust Bayesian optimization algorithms.

Part II

Simulation and Optimization of Quantum Algorithms

Chapter 5

Background of Quantum Computing

In this chapter, we introduce some basic conceptions of quantum computing.

Quantum State. In quantum computing, a state vector, typically denoted as $|\boldsymbol{\psi}\rangle$, can describe a system. For a simple two-level quantum system (a qubit), this could be written as:

$$|\boldsymbol{\psi}\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{5.1}$$

Here, $|0\rangle$ and $|1\rangle$ represent the two basis states (similar to the 0 and 1 in classical computing), while α and β are complex coefficients. According to the principles of quantum mechanics, the absolute squares of these coefficients, $|\alpha|^2$ and $|\beta|^2$, give the probabilities of measuring the states $|0\rangle$ and $|1\rangle$ respectively. Moreover, due to the conservation of probability, these values sum to 1: $|\alpha|^2 + |\beta|^2 = 1$

The density operator (or density matrix), $\boldsymbol{\rho}$, provides a more general description of a quantum state, especially when the system is in a mixed state, meaning a statistical mix of different quantum states. If we have a system in state $|\boldsymbol{\psi}_i\rangle$ with probability p_i,

the density operator is given by:

$$\boldsymbol{\rho} = \sum p_i \left|\boldsymbol{\psi}_i\right\rangle \left\langle\boldsymbol{\psi}_i\right| \tag{5.2}$$

The symbol $\langle\boldsymbol{\psi}_i|$ is the complex conjugate transpose (also known as the bra) of the state vector $|\boldsymbol{\psi}_i\rangle$ (also known as the ket).

Quantum Gate. Quantum gates are the building blocks of quantum circuits and represent the basic operations that can be applied to qubits. Like classical logic gates, they perform operations on a certain number of input qubits and produce an output. However, unlike classical gates, quantum gates can be reversible and can operate on superpositions of their inputs. Let's consider a single qubit gate, the Pauli-X gate, often compared to a classical NOT gate. It is represented by the matrix:

$$\mathrm{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Multi-qubit gates are also crucial in quantum computing. An example of such a gate is the CNOT gate, a two-qubit gate that flips the second qubit (target) if and only if the first qubit (control) is in state $|1\rangle$. The matrix representing the CNOT gate is:

$$\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Quantum gates are usually represented as unitary matrices, as they preserve the quantum mechanical principle of conservation of probability. The state vectors and density

operators, when operated on by these gates, transform the state of the quantum system to enable computations.

Measurement. Measurement is an action that determines the state of a quantum system. Given a qubit in state $|\boldsymbol{\psi}\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, a measurement of this state will give the result 0 with probability $|\alpha|^2$, and 1 with probability $|\beta|^2$. These probabilities are calculated from the square of the absolute value of the coefficients of the state vector, as per the Born rule.

$$\text{Prob}(0) = |\alpha|^2, \quad \text{Prob}(1) = |\beta|^2$$

This leads to the post-measurement states being $|0\rangle$ and $|1\rangle$, respectively. In the case of a density operator $\boldsymbol{\rho}$ representing our quantum state, the probability of obtaining a specific measurement outcome m can be calculated as follows:

$$\text{Prob}(m) = \text{Tr}(\mathbf{E}_m \boldsymbol{\rho}),$$

where $\mathbf{E}_m$ is the measurement operator corresponding to the outcome m, and $\text{Tr}(\cdot)$ denotes the trace of a matrix. After the measurement, the quantum state collapses to the corresponding eigenstate of the measurement.

Quantum Circuit. Two popular trends of quantum computing are digital and analog quantum computing, which differ in how they use quantum phenomena to process information. Digital quantum computing involves encoding information into qubits and applying sequences of unitary gate operations (quantum gates) to these qubits. Analog quantum computing, also known as quantum annealing or adiabatic quantum computing, takes advantage of continuous quantum dynamics. Analog quantum computing is particularly suited to solve optimization problems and simulations of quantum dynamics.

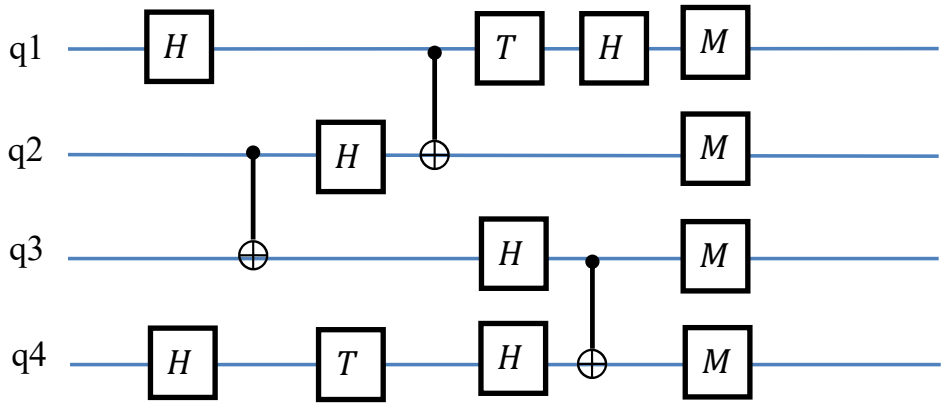

Figure 5.1: An example of a quantum circuit.

In this dissertation, we mainly focus on digital quantum computation. Digital quantum computing is considered universal since it holds general-purpose computing capabilities.

The quantum circuit is a model that describes the computation process as a sequence of quantum gates, with qubits as the basic unit of quantum information. The circuit offers a framework for representing quantum algorithms and computations in a manner that is easy to understand and implement on a quantum computer. An example of a quantum circuit is shown in Fig. 5.1. The quantum circuit belongs to the block level in the design abstraction discussed in Sec. 1.2.1.

Variational Quantum Algorithm. Variational quantum algorithms are a class of quantum algorithms that rely on classical optimization to find the best parameters for a quantum computation. They're a type of hybrid quantum-classical method, and are well-suited for near-term quantum computers, often referred to as Noisy Intermediate-Scale Quantum (NISQ) devices, which can handle a middle-scale quantum system but still suffer from significant noise.

In a variational quantum algorithm, a problem is mapped onto a parameterized quantum circuit (also called a variational quantum circuit), where the outcome of the com-

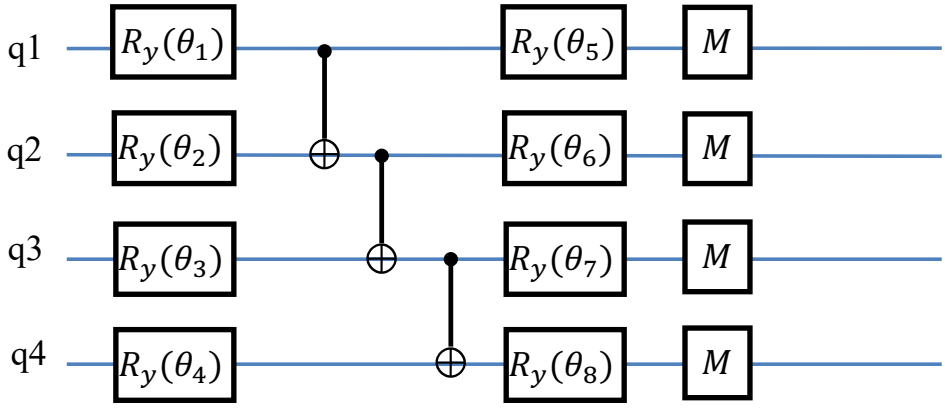

Figure 5.2: An example of a variational quantum circuit.

putation depends on a set of parameters. The structure of such a circuit ansatz typically consists of a series of gates that prepare an initial state, followed by a sequence of gate operations that depend on a set of parameters. An example of a variational quantum circuit is shown in Fig. 5.1. A classical optimizer is used to iteratively adjust these parameters based on the result of measurements made on the quantum circuit, aiming to find the minimum of a specific cost function. These variational quantum algorithms are designed to maximize the efficiency of quantum resources and offer a practical pathway for the near-term deployment of quantum computing technologies.

Chapter 6

Variational Method for QAOA Simulation

6.1 Introduction

In this chapter, we will discuss the classical simulation of quantum algorithms and specifically focus on a variational approach.

In light of the rapid advancements in quantum hardware, quantum computing has demonstrated potential advantages in specific applications [20, 167]. However, the practical application of quantum computing remains limited due to imperfections in current quantum hardware, such as intrinsic quantum noise) and the high cost of public access. Therefore, classical simulation of quantum systems [32, 168] plays an important role in the development, verification, and benchmarking of quantum algorithms. Some widely-used classical simulators include state vector simulation [169], tensor network models [170–174], decision diagrams [175], and customized simulators for certain circuit families like the Clifford circuits [176, 177].

The simulation of a quantum approximate optimization algorithm (QAOA) [178]

has gained considerable attention [171, 179]. With its potential to show advantages in optimization problems, QAOA is one notable example of variational quantum algorithms. QAOA and its variants [180, 181] have been extensively studied in recent years, including its theoretical foundations and mechanism [182–185], parameter optimization [38,186,187], hardware realizations [188,189] and benchmarking [190]. Regarding its simulation, while QAOA energy could usually be simulated efficiently using the lightcone structure [178,191], simulating its output state on a classical computer presents a significant challenge.

Recently, a classical simulation of the QAOA algorithm using a quantum neural state was proposed in [34]. Its key idea is to parameterize the quantum state by a specific neural network model, restricted Boltzmann machine (RBM), and then simulate the gates in the circuit by variationally updating the RBM parameters. Neural network models have been widely used in quantum many-body systems, and states approximated by these models are often referred to as neural quantum states in the literature [192–196]. A series of works have investigated the capacity of neural quantum state models, demonstrating their remarkable expressive power [197–201]. The equivalence between RBM and tensor network models was shown in Ref. [202].

Regarding using the neural quantum state as a circuit simulator, Ref. [34] demonstrated that the variational method can accurately simulate quantum QAOA circuits in the near-optimal parameter regime. However, the performance of this method in broader contexts is yet to be thoroughly investigated.

Main contributions. In this chapter, we discuss and analyze the performance of variational methods in simulating QAOA circuits. It is widely recognized that the entanglement entropy (EE) of a quantum system is the bottleneck of its tensor network representation [203]. The scaling of EE has been used to quantify the complexity of a quantum circuit [204], and entropy tracking has also been used to analyze the impact of

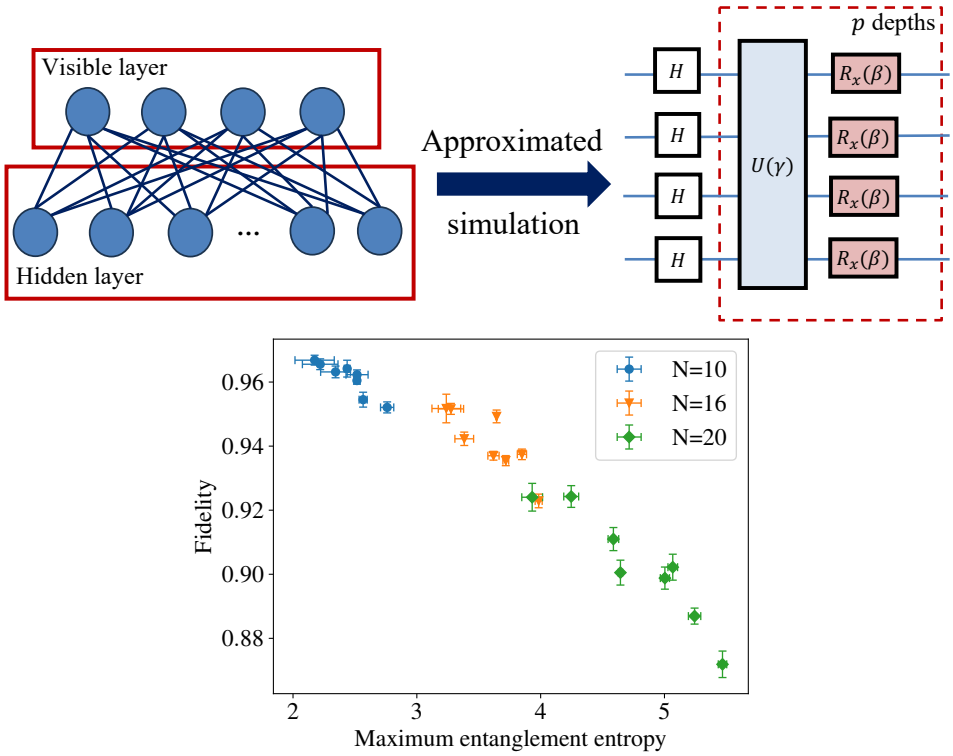

Figure 6.1: The overview figure of this chapter. The QAOA circuit is approximately simulated by a variational method parametrized by an RBM model. The bottom figure illustrates the simulation results of QAOA circuits with depths ranging from $p = 1$ to $p = 8$, applied to MaxCut graphs of sizes $N = 10, 16, 20$. We focus on the relationship between the fidelity of variational simulation and the entanglement entropy of the circuit. Each data point represents the simulation results of a specific QAOA circuit corresponding to a particular depth p and graph size N. Our findings indicate that as the graph size N increases, the maximum entanglement entropy rises while the fidelity decreases.

noise on quantum algorithms [205,206]. Our study reveals that EE continues to present a challenge for the simulation performance of variational methods. As depicted in Fig. 6.1, the approximation accuracy noticeably decreases as the state EE increases. After identifying the limitations of the variational method, we attempt to improve simulation accuracy by investigating two key components of variational simulation algorithms: the size of the RBM model and the quality of Markov Chain Monte Carlo (MCMC) sampling. The size of an RBM model is related to the model's capacity to represent a quantum

state, and the MCMC samples are utilized in stochastic optimization during gate approximation. Our findings indicate that the simulation performance can be improved by increasing the model size and the number of MCMC samples. However, these approaches could considerably increase computational costs, and the degree of improvement remains limited. These observations underline the challenges associated with simulating complex circuits via variational methods and emphasize the need for further research to broaden the regime of simulatable circuits using this approach.

6.2 Background and Formulation

6.2.1 MaxCut Problem

Given a graph $G = (V, E)$ with vertices V and edges E, the MaxCut problem aims to find a cut that partitions the graph vertices into two sets that have the largest number of edges between them. The associated cost Hamiltonian can be written as:

$$\mathbf{H}_p = \sum_{(i,j)\in E} \mathbf{I} - \mathbf{Z}_i\mathbf{Z}_j, \tag{6.1}$$

where $\mathbf{Z}_i$ is a Pauli-Z operator. We aim to search for a bitstring $\mathbf{x}$ that minimizes the cost $\langle \mathbf{x}|\mathbf{H}_p|\mathbf{x}\rangle$.

6.2.2 Quantum Approximate Optimization Algorithm (QAOA)

QAOA was proposed to solve combinatorial optimization problems by preparing a N-qubit quantum state using a sequence of alternating operators, which are parameterized

by $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$, with a predefined circuit depth p:

$$|\boldsymbol{\psi}_p\rangle = \mathbf{U}_M(\beta_p)\mathbf{U}_P(\gamma_p)\cdots\mathbf{U}_M(\beta_1)\mathbf{U}_P(\gamma_1)\,|\boldsymbol{\psi}_0\rangle\,. \tag{6.2}$$

$\mathbf{U}_P(\gamma_l) = e^{-i\gamma_l\mathbf{H}_P}$ is the phase sepeartion operator with $\mathbf{H}_P$ being the problem Hamiltonian (6.1). $\mathbf{U}_M(\beta_l) = e^{-i\beta_l\mathbf{H}_M}$ is the mixing operator with $\mathbf{H}_M = \sum_{j\in V}\mathbf{X}_j$ and $\mathbf{X}_j$ is the Pauli-X matrix. The initial state $|\boldsymbol{\psi}_0\rangle = |+\rangle^{\otimes N}$ is a superposition of all computational basis with $|+\rangle$ being a uniform superposition state of a single qubit. QAOA is a hybrid variational quantum algorithm where the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are optimized via a classical optimization routine.

6.2.3 Variational Simulation of QAOA

For a N-qubit quantum system, the Hilbert space is spanned by the computational basis $|\mathbf{x}\rangle : \mathbf{x} \in \{0,1\}^N$ of classical bit strings $\mathbf{x} = (x_1, \ldots, x_N)$. An arbitrary state in the Hilbert space is represented as $|\boldsymbol{\psi}\rangle = \sum_{\mathbf{x}} \psi(\mathbf{x})\,|\mathbf{x}\rangle$. The idea of the variational simulation is to approximate $|\boldsymbol{\psi}\rangle$ via some parameterized neural network models and simulate gates by updating its parameters variationally. More specially, a restricted Boltzmann machine (RBM) has been used to approximate the quantum state [34]:

$$\psi(\mathbf{x}) \approx \psi_{\boldsymbol{\theta}}(\mathbf{x}) \equiv \exp\left(\sum_{j=1}^{N} a_j x_j\right) \cdot \prod_{k=1}^{N_h}\left[1 + \exp\left(b_k + \sum_{j=1}^{N} W_{jk}x_j\right)\right]. \tag{6.3}$$

Here, the RBM model has complex-valued parameters $\boldsymbol{\theta} = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$ where $\mathbf{a} \in \mathbb{C}^N$ are visible biases, $\mathbf{b} \in \mathbb{C}^{N_h}$ are hidden biases, and $\mathbf{W} \in \mathbb{C}^{N\times N_h}$ are weights. The schematic of an RBM model is shown in the upper left of Fig. 6.1.

An advantage of utilizing an RBM model is that given a representation of (6.3), some one- and two-qubit gates operators can be exactly simulated as the mappings between

two different sets of parameters $\boldsymbol{\theta} \to \boldsymbol{\theta}'$. Given a gate G, the parameter mapping is determined by solving nonlinear equations:

$$\langle \mathbf{x}|\boldsymbol{\psi}_{\boldsymbol{\theta}'}\rangle = C\,\langle \mathbf{x}|\mathrm{G}|\boldsymbol{\psi}_{\boldsymbol{\theta}}\rangle\,, \forall \mathbf{x} \in \{0,1\}^N. \tag{6.4}$$

If there exists a solution $\boldsymbol{\theta}'$ for all bitstrings and any constant C, then the gate G can be exactly simulated. For example, the widely used ones include the Pauli-XYZ gates, Z rotation (RZ) gates, and the ZZ rotation (RZZ) gate with an additional hidden neuron in RBM. Note that the RZZ gate is one of the most fundamental components in QAOA algorithms for combinatorial optimization problems. See the exact detailed update rules in Appendix A.3.1.

Other gates that do not have exact solutions of (6.4) need to be approximated through a variational optimization scheme. Notably, the RX gate in the QAOA belongs to this category. Suppose we have a current quantum state $|\boldsymbol{\psi}_{\boldsymbol{\theta}}\rangle$ parametrized by $\boldsymbol{\theta}$ and the target state $|\boldsymbol{\phi}_{\boldsymbol{\theta}'}\rangle$ which is parametrized by $\boldsymbol{\theta}'$ (denoted as $\boldsymbol{\psi}_{\boldsymbol{\theta}}$ and $\boldsymbol{\phi}_{\boldsymbol{\theta}'}$ in the equations for simplicity), we aim to minimize the distance between them:

$$\begin{aligned} \min_{\boldsymbol{\theta}'} \quad & D(\boldsymbol{\phi}_{\boldsymbol{\theta}'}, \boldsymbol{\psi}_{\boldsymbol{\theta}}) = 1 - F(\boldsymbol{\phi}_{\boldsymbol{\theta}'}, \boldsymbol{\psi}_{\boldsymbol{\theta}}), \text{with} \\ & F(\boldsymbol{\psi}_{\boldsymbol{\theta}}, \boldsymbol{\phi}_{\boldsymbol{\theta}'}) = \frac{|\langle \boldsymbol{\phi}_{\boldsymbol{\theta}'}|\boldsymbol{\psi}_{\boldsymbol{\theta}}\rangle|^2}{\langle \boldsymbol{\phi}_{\boldsymbol{\theta}'}|\boldsymbol{\phi}_{\boldsymbol{\theta}'}\rangle\,\langle \boldsymbol{\psi}_{\boldsymbol{\theta}}|\boldsymbol{\psi}_{\boldsymbol{\theta}}\rangle}. \end{aligned} \tag{6.5}$$

The above minimization problem can be solved by some stochastic optimizers. In this chapter, we adopt the Stochastic Reconfiguration algorithm [207–209], which is also used in [34]. We will illustrate the detailed optimization iterations to update RBM parameters in Appendix A.3.2.

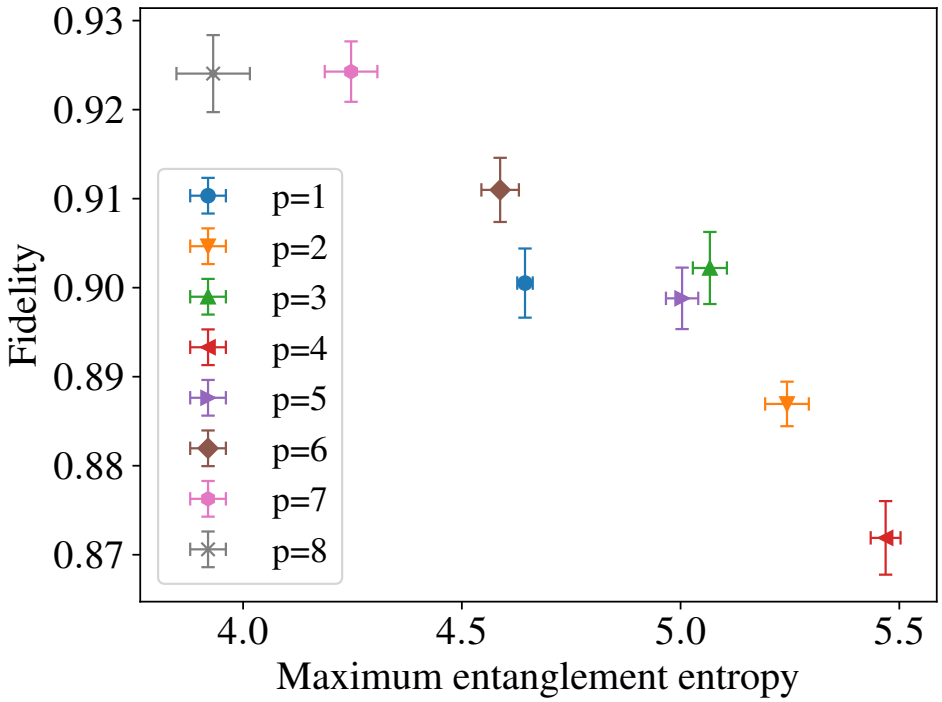

Figure 6.2: The relation between the maximum entanglement entropy and simulation fidelity of QAOA circuits with different depths on $N = 20$ MaxCut graphs. The maximum entanglement entropy refers to the highest value observed among all the layers in the QAOA circuit, while the fidelity is computed based on the final state of the circuit.

6.3 Entanglement Entropy Limits the Variational Simulation

In this section, we begin by introducing entanglement entropy (EE) and its measurement which is used for the variational simulation method. Then, we explore the relationship between the approximation performance and EE.

6.3.1 Entanglement Entropy

EE is a key measure of quantum correlations between subsystems within a larger quantum system. It quantifies the amount of information or uncertainty that is shared

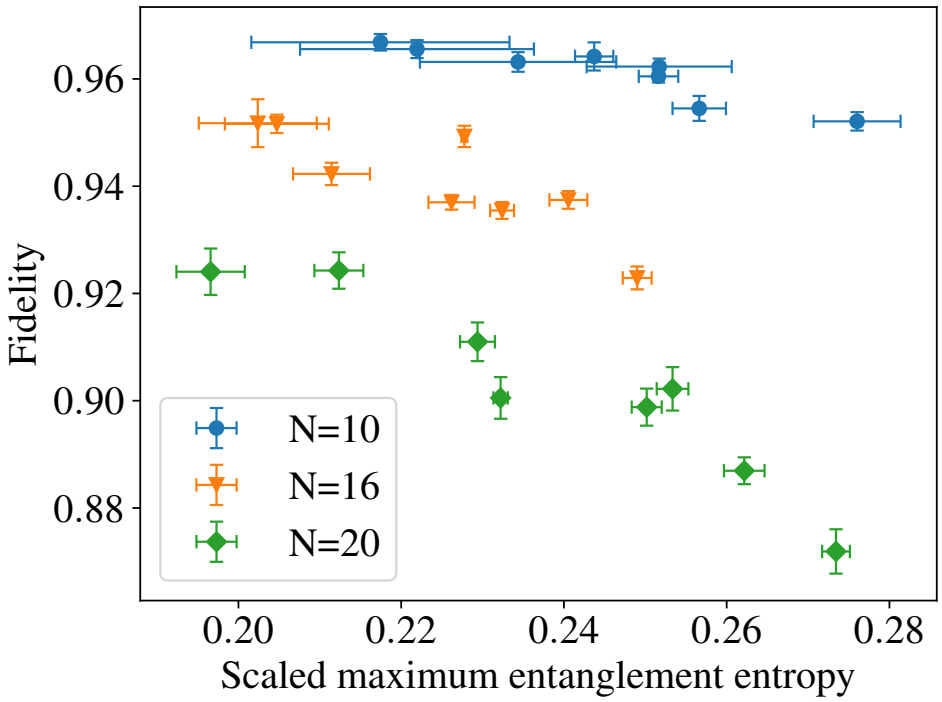

Figure 6.3: System size N strongly influences the maximum entanglement entropy. By rescaling the entanglement entropy by $\frac{1}{N}$, simulation results of different sizes exhibit consistent trends but with varying fidelity. The data presented in the bottom panel of Fig. 6.1 corresponds to the same dataset as the non-rescaled data.

between two subsystems, capturing the degree of entanglement between them. Given a bipartition of a quantum system into two subsystems, A and B, the entanglement entropy is the Von Neumann entropy of a subsystem:

$$S(\boldsymbol{\rho}_A) = -\mathrm{Tr}(\boldsymbol{\rho}_A \log(\boldsymbol{\rho}_A)), \tag{6.6}$$

where $\boldsymbol{\rho}_A = \mathrm{Tr}_B(\boldsymbol{\rho})$ is the reduced density matrix of subsystem A.

EE can also be understood by Schmidt decomposition. Given a pure state $|\boldsymbol{\psi}\rangle$ within a composite Hilbert space $\mathcal{H}_A \bigotimes \mathcal{H}_B$, its Schmidt decomposition can be expressed as

$$|\boldsymbol{\psi}\rangle = \sum_i \lambda_i |\mathbf{u}_i\rangle_A |\mathbf{v}_i\rangle_B, \tag{6.7}$$

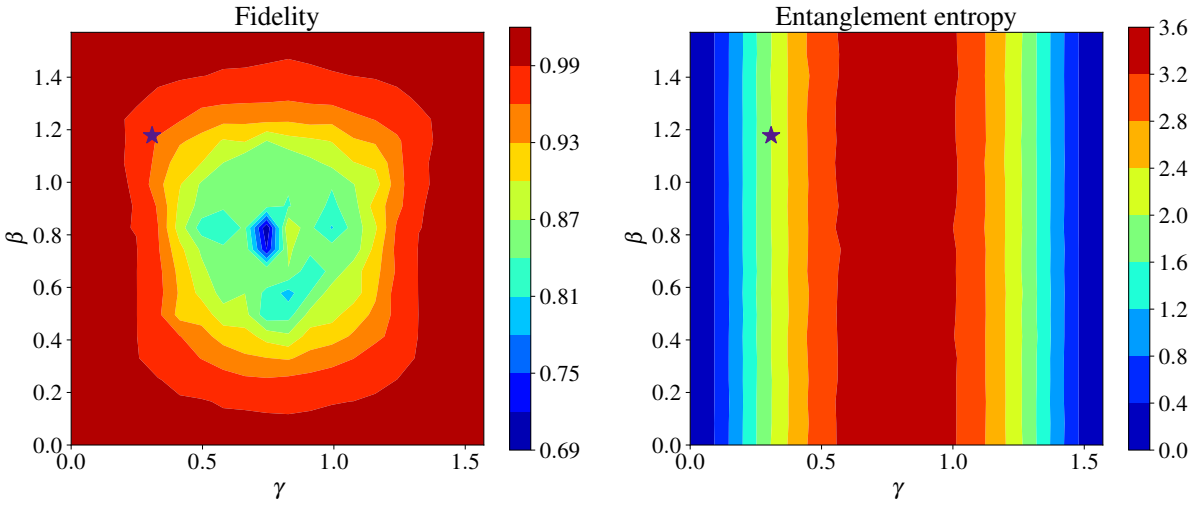

Figure 6.4: The fidelity and entanglement entropy of a $p = 1$ QAOA circuit on $N = 8$-node MaxCut graphs, whose γ and β is determined by a grid search with the resolution of 20×20. The circuit fidelity exhibits a high sensitivity to the chosen parameter settings. Circuits with relatively higher entanglement entropy tend to have lower fidelity in simulation. The optimal QAOA parameters are indicated by a star symbol in the heatmap, which is associated with high fidelity and low entanglement entropy.

where λ_i denote the Schmidt coefficients, and $|\mathbf{u}_i\rangle_A$ and $|\mathbf{v}_i\rangle_B$ are orthonormal sets of states in the Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively. The entanglement entropy S between two subsystems is given as

$$S(\boldsymbol{\rho}_A) = S(\boldsymbol{\rho}_B) = -\sum_i \lambda_i^2 \log(\lambda_i^2) \tag{6.8}$$

Importantly, given a $|\boldsymbol{\psi}\rangle$, the value of its EE depends on the partition order we used.

In our context, to fairly capture the maximal entanglement of a subsystem, we always partition the state into two equally sized but randomly selected subsystems. Given a N-qubit state $|\boldsymbol{\psi}\rangle$ (where N is assumed to be even), with all qubits denoted by a set $Q = \{1, 2, \ldots, N\}$, we estimate its entanglement entropy as follows:

1. Sample a random subset A containing $\frac{N}{2}$ distinct elements from Q such that $A \subset Q$

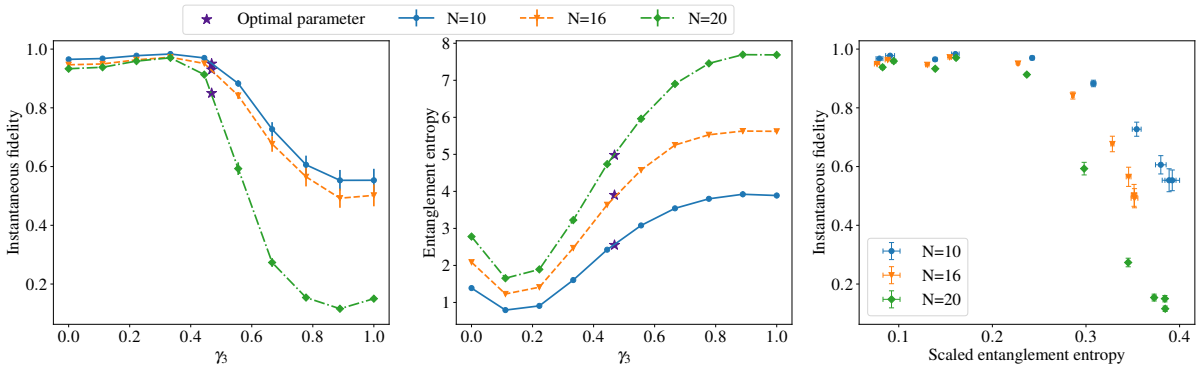

Figure 6.5: The fidelity and entanglement entropy of a $p = 8$ QAOA circuit at the $l = 3$ layer are investigated on $N = 10, 16, 20$-node MaxCut graphs, where all parameters are optimized except for γ_3, which is determined through a grid search. The value of the γ_3 parameter strongly impacts both the EE of the state and the circuit fidelity. Notably, a high entanglement entropy associated with a chosen γ_3 leads to a significant drop in fidelity. The optimal γ_3 is also highlighted in the figures.

and $|A| = \frac{N}{2}$.

2. Calculate the EE using Eq. (6.8).

3. Repeat the above steps and report the statistical results as the EE of $|\psi\rangle$.

Note that while our focus is on pure states in this chapter, the estimation method can be easily extended to mixed states by flattening the density matrices.

6.3.2 Variational Simulation & Entanglement Entropy

Intuitively, it is more challenging to represent a state with higher EE using a tensor network model. We confirm that this holds true for the variational method. Our experiments focus on the QAOA circuit for the MaxCut problem. For N-node graphs, we use multiple non-homogeneous 3-regular graphs to benchmark the simulations. For $N < 10$, we use all the available non-homogeneous graphs. For $N \geq 10$, we randomly use 10 graphs for benchmarking since the number of non-homogeneous graphs grows exponentially. The EE is estimated by randomly selecting 100 permutations of partition

orders, and we take its average value. In all the figures and tables, without specification, we report the average value of the interested quantities (e.g., the fidelity and EE) over the graph instances, and the error bars are the standard deviations of the average values.

QAOA with near-optimal parameters.

First, we examine the approximated simulation performance of QAOA circuits with near-optimal parameters. Here, the QAOA parameters are determined by the fixed angle conjecture [210], a well-established set of parameters that work well for regular MaxCut graphs.

In Fig. 6.2, we present the simulation results for MaxCut graphs of size $N = 20$ with QAOA depth $p = 1, 2, \ldots, 8$. Fidelity is calculated exactly as the overlap between the approximated final state and the exact final state. Regarding the maximum entanglement entropy, we collected the EEs of the simulated states after each QAOA layer $l (1 \leq l \leq p)$ and reported the maximum observed value. As is evident from the results, fidelity and maximal EE are highly correlated. Simulating a QAOA circuit with a higher EE proves to be more challenging numerically. Interestingly, a higher-depth QAOA does not necessarily translate to increased simulation difficulty for the same graph. We notice the highest maximal EE and lowest fidelity at the $p = 4$ QAOA circuit. However, as p increases from 5 to 8, a decrease in maximal EE and an increase in fidelity are observed. This suggests that the variational method may be better suited for simulating higher-depth QAOA circuits (with optimized parameters).

Next, we replicate these simulations for MaxCut graphs of sizes $N = 16$ and $N = 10$. Noting that the maximum EE of a circuit is heavily dependent on the circuit size N, we perform a rescaling of the maximum EE by a factor of $\frac{1}{N}$ and illustrate the results in Fig.6.3. Across varying N, the relationship between fidelity and scaled maximal EE remains consistent: circuits with larger EEs are more challenging to approximate. Given

the same variational methods setup, the simulation becomes more difficult as the size of the QAOA circuit increases. This suggests that simulating a larger-sized QAOA circuit could be more challenging. If we omit the scaling of maximal EE, the results for different N are shown at the bottom of Fig. 6.1.

QAOA with unoptimized parameters.

To extend our understanding of the variational method performance beyond the near-optimal parameter space, we conducted simulations of QAOA with unoptimized parameter settings.

First, we performed a grid search over the two-dimensional parameter space, encompassing one-dimensional γ and one-dimensional β, for a QAOA circuit of size $N = 8$ and depth $p = 1$. The heatmap of fidelity and EE under varying parameters is shown in Fig. 6.4. Notably, even for a graph of this relatively small size, the simulation fidelity demonstrates significant sensitivity to the QAOA parameters. The regions of low-fidelity parameters largely coincide with areas of high EE, reaffirming the strong relationship between EE and simulation performance. Although there exist certain parameters where high EE coincides with high simulation fidelity, we argue that for a substantial parameter space (including the near-optimal space), a circuit with larger EE is generally more challenging to approximate. We also highlight the optimal parameters from the fixed-angle conjecture in the figures, which corresponds to a low EE and high fidelity.

To deepen our understanding of high-depth QAOA simulation performance, we developed a one-dimensional grid search strategy due to the prohibitive computational costs of a full grid search in high-dimensional space. This strategy sweeps γ_l for a specific layer l ($1 \leq l \leq p$) in a depth-p QAOA circuit while holding all other $2p - 1$ parameters constant at their optimal values. The QAOA circuit is then simulated up to layer l, and the fidelity at this stage, which is referred to as the *instantaneous fidelity*, is used as the

measure of performance. We use the term "instantaneous" to differentiate the simulation fidelity associated with a specific layer from the (final) fidelity, which corresponds to the simulation of the entire depth-p QAOA circuit.

In Fig. 6.5, we report the sweeping simulation results of the $l = 3$ layer of a $p = 8$ QAOA for graphs with different sizes N. Once again, the simulation performance exhibits sensitivity to the value of γ_3, and we note the strong correlation between EE and instantaneous fidelity. The optimal γ_3 appears in a region of relatively high fidelity and low EE. The associated scatter plot visually illustrates the instantaneous fidelity and EE of QAOA circuits for specific γ_3 values. By rescaling EE by $\frac{1}{N}$, we observe a consistent trend among different-sized QAOA circuits. However, larger-sized QAOA circuits appear to be more sensitive to γ_3 value, implying that as N increases, the parameter space that is "easy to simulate" becomes narrower.

6.3.3 Remarks

It is worth mentioning that we are not claiming that the RBM model can not represent a highly entangled state. Many papers have explored the entanglement scaling of neural network states and have shown their exceptional expressive power [197–200]. In fact, in our simulations, an RBM model with low fidelity also manifests a comparable EE to the exact state. However, our focus lies on applying the variational method in circuit simulation, which involves extensive optimization of RBM parameters.

Revisiting Fig. 6.4, we note that the value of β does not influence the EE as the mixing layer consists only of single-qubit gates. The subtle irregularities in the EE heatmap at a fixed γ value can be attributed to the stochastic estimation of EE. Meanwhile, the heatmap indicates that the simulation fidelity varies a lot for a fixed γ but differing β. This suggests that other factors other than EE also have a role in influencing variational

Algorithm 5 The variational method for simulating QAOA for MaxCut

Input: A depth-p QAOA circuit, graph $G = (V, E)$

Initialize the RBM model $|\boldsymbol{\psi}_{\boldsymbol{\theta}}\rangle$ for the initial state $|\boldsymbol{\psi}_0\rangle$
for $l = 1, 2, \ldots, p$ **do**
    # Simulate the problem Hamiltonian layer exactly
    for edges in E **do**
        Update the parameter θ analytically
    end for
    # Simulate the mixer layer approximately
    for nodes in V **do**
        while not converge **do**
            Sample from current and target states $\mathbf{x}_{\boldsymbol{\psi}} \sim |\boldsymbol{\psi}_{\boldsymbol{\theta}}|^2$, $\mathbf{x}_{\boldsymbol{\phi}} \sim |\boldsymbol{\phi}|^2$
            Update $\boldsymbol{\theta}$ via stochastic reconfiguration based on samples of $\mathbf{x}_{\boldsymbol{\psi}}$ and $\mathbf{x}_{\boldsymbol{\phi}}$
        end while
    end for
    Compress the RBM model
end for

simulation.

Having explored the relationship between EE and simulation fidelity, we will now delve into the details of the variational simulation algorithm and conduct an analysis of its performance.

6.4 Analyze the Performance of Approximation

In this section, we aim to analyze the model performance from the algorithmic perspective, with a particular focus on enhancing the simulation fidelity, especially when the QAOA parameters are beyond the near-optimal space. A detailed illustration of the variational simulation algorithm can be found in Algorithm 5, with a detailed explanation provided in Appendix A.3.1.

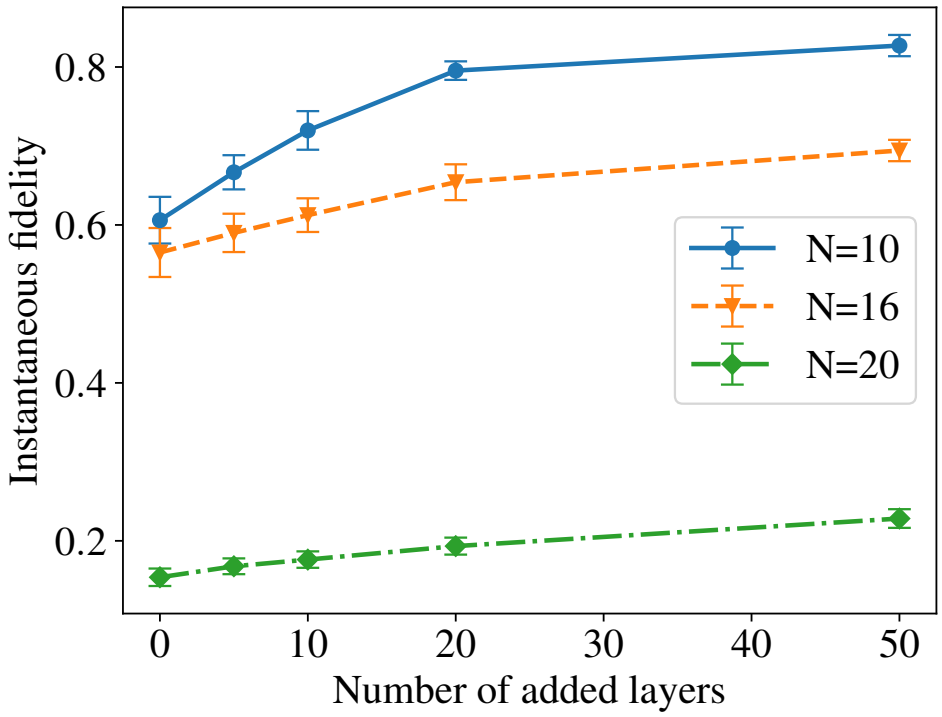

Figure 6.6: We explore the impact of increasing the number of hidden layers in the RBM model for approximating a $p = 8$ QAOA circuit at the third layer on $N = 10, 16, 20$-node MaxCut instances. All parameters of the circuit are optimized, except for the third γ which is set to 0.9. By enlarging the RBM model, we successfully improve the fidelity of the approximation. However, it is important to note that the degree of improvement in fidelity for larger-sized circuits is limited despite the increase in the size of the RBM model.

6.4.1 Enlarge RBM Models

An intuitive approach to improving simulation performance is to enlarge the size of the RBM model, thereby increasing its capacity. Consequently, before simulating the mixer layer (between lines 7 and 8 in Algorithm 5), we add additional hidden layers with zero values.

We also tried sequentially adding additional hidden layers after each RX gate (between lines 8 and 9 in Algorithm 5). The results of the latter approach are similar to the former strategy. Hence, we report the results of enlarging the RBM model using the former method. It is probable that a more informative initialization of the added hidden layers

Table 6.1: The Simulation Fidelity of Different Graphs under RBM Models with Different Sizes.

N	# added hidden layer	# parameter	Fidelity
10	0	186	0.606 ± 0.030
	20	406	0.795 ± 0.011
	50	736	0.827 ± 0.013
16	0	441	0.565 ± 0.031
	20	781	0.654 ± 0.023
	50	1291	0.694 ± 0.013
20	0	671	0.154 ±0.011
	20	1091	0.193 ±0.011
	50	1721	0.228 ±0.012

could contribute to improved simulation, which could be worth further investigation.

In Fig. 6.6, we examine a depth-$p = 8$ QAOA circuit with an unoptimized $\gamma_3 = 0.78$ and simulate the circuit up to the $l = 3$ layer using variously sized RBM models. As can be read from Fig. A.3, such QAOA circuits have high EE and low instantaneous fidelity. For simplicity, we only augment the RBM model when simulating the third layer of the QAOA circuit, as this layer's parameter is the only one that remains unoptimized. As can be observed from the results, increasing the size of the RBM model does improve the simulation fidelity. However, the scope of improvement is relatively limited, especially for larger N graphs. We argue that it is due to the size of the target state being 2^N, which increases exponentially with N, while the number of parameters only grows polynomially with N. For an RBM model with N visible nodes, the addition of one hidden layer equates to an increase of $(N + 1)$ parameters. A summary of the RBM parameters and their performance is provided in Table 6.1. It suggests that learning a highly entangled state from a size-N QAOA becomes progressively more challenging by simply augmenting more hidden layers.

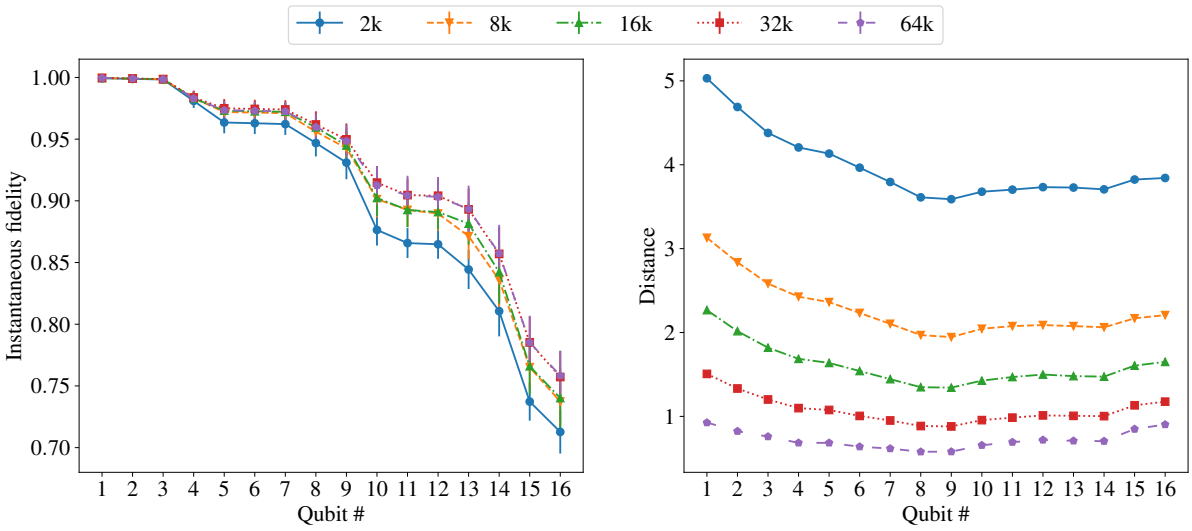

Figure 6.7: We show the results of the variational simulation under different sampling setups for a $N = 16$ MaxCut problem using a $p = 1$ QAOA circuit with optimized β and γ values, specifically $\gamma = 0.7$. By approximating the RX gate one by one, we observe changes in both fidelity and distance. Increasing the number of MCMC samples proves beneficial in reducing the distance between the true probability distribution and the estimated one, while also increasing the fidelity of the approximation.

6.4.2 Increase the Number of MCMC Samples

Beyond the structure of the RBM model, the optimization algorithm employed to update the RBM parameters also holds critical importance in the variational simulation. For our experiments, we utilize the Stochastic Reconfiguration method (refer to Appendix A.3.2). In Stochastic Reconfiguration, the quality of sampling (as noted in line 10 of Algorithm 5) is of significant importance as it serves as the foundation of estimating gradients in the RBM parameter update. Here, we use the Markov Chain Monte Carlo (MCMC) method [211] to sample from quantum states that are parametrized by RBM models. In our experiments, we propose a sample in the MCMC chain by randomly flipping one bit. Without specification, we use a total of 8000 samples from four independent chains, warming up each MCMC chain with an additional 8000 samples and taking 20 steps between two accepted samples. To explore how sampling quality impacts

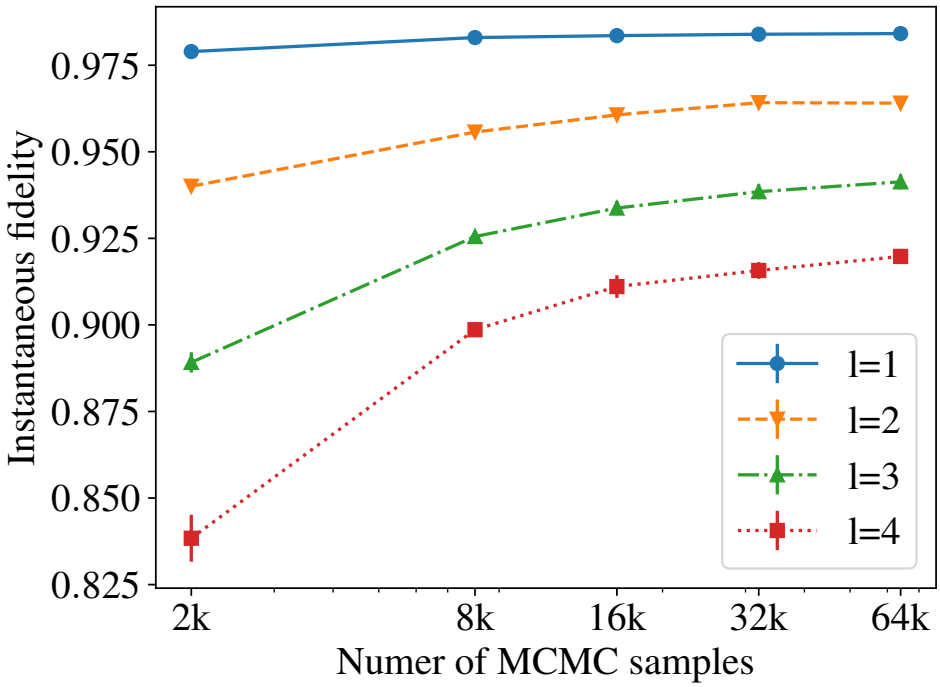

Figure 6.8: The performance of $p = 4$ QAOA circuits on a $N = 16$ problem under different numbers of MCMC samples. The parameters of the circuits are well-optimized, except for γ_4, which is fixed at 0.3. We observe that increasing the number of MCMC samples leads to improved fidelity at each layer of the QAOA circuit.

simulation performance, we modified the number of MCMC samples used to simulate the same QAOA circuits.

In Fig 6.7, we examine a $N = 16, p = 1$ QAOA circuit with an unoptimized $\gamma = 0.7$ and an optimal β. Its heatmaps, generated from scanning two parameters, would be similar to Fig. 6.4. We monitored the fidelity after simulating RX gates individually and labeled the fidelity after simulating specific RX gates as instantaneous fidelity. We also reported the distance between the exact and approximated states along the simulation path of RX gates. This distance measures the Kullback-Leibler (KL) divergence between the exact probability distribution and the empirically estimated one obtained from MCMC samples. As we increase the number of MCMC samples, we observe a significant improvement in the simulation fidelity and a decreasing sampling distance. It implies that

accurate sampling from the quantum states is crucial for the variational method, and increasing the number of samples is a straightforward approach to improving the sampling quality.

We also validated our observation regarding the number of samples in higher-depth QAOA circuits in Fig. 6.8. An increasing number of samples can benefit the instantaneous fidelity in each layer. However, as evident from the figure, the degree of improvement progressively decays with a growing number of MCMC samples. We also attempted to sample from the exact state during stochastic optimization in our experiments, yet a significant gap in fidelity remained. This suggests that enhancements in simulation performance from merely improving sampling quality and increasing the number of MCMC samples are limited, similar to the observations from augmenting the RBM model.

Meanwhile, some studies have shown that it could be challenging for classical techniques, such as MCMC, to sample from QAOA states whose probability distribution is close to a low-temperate Boltzmann distribution [212, 213]. This inherent difficulty in classical sampling could potentially be an obstacle to optimizing RBM parameters. While increasing the number of MCMC samples might enhance the estimates, it would invariably lead to increased computational costs. In our experiments, we utilize all available MCMC samples to estimate the gradients and other related quantities in the Stochastic Reconfiguration. One possible solution could be to use a batch of MCMC samples for gradient estimation, although this would require further benchmarking for validation.

In summary, our exploration has analyzed two essential ingredients of the variational simulation methodology within this section. We noted the potential improvement from increasing the RBM model size and the number of MCMC samples during stochastic optimization. Nevertheless, both of these strategies entail substantial additional computational costs, which could constrain the performance of classical simulations. More significantly, our findings suggest that the fidelity improvements from these two approaches

are limited. Consequently, the development of a comprehensive solution to enhance the variational method for simulating QAOA circuits, particularly in the context of beyond-near-optimal parameter spaces, remains an area of significant interest.

6.5 Conclusion

In this chapter, we have discussed the variational method for simulating QAOA circuits. Intuitively, the variational method is suitable for this task since it can simulate the problem Hamiltonian layer, which consists of RZZ gates, analytically, with only the RX gates in the mixing layer requiring approximation. As reported in the previous work [34], the method can efficiently simulate the QAOA circuits with near-optimal parameters, achieving high fidelity. However, the variational method often encounters difficulties when simulating circuits with high entanglement entropy, such as most QAOA circuits with unoptimized parameters. We tried to improve the simulation performance by augmenting the RBM model and enhancing the sampling quality during the stochastic optimization of model parameters. While both strategies yielded performance improvements, they also introduced additional computational costs and, notably, their potential for further improvement appeared limited. This underscores the need for improvements to the variational method to facilitate the simulation of QAOA circuits beyond near-optimal parameter spaces and more entangled circuits. Possible directions for exploration include a more suitable neural network structure for variational simulation, utilizing a better parameter optimization algorithm, and developing a parallel and distributed scheme for variational simulation.

In our experiments, we restricted our simulations to $N = 20$ MaxCut graphs due to the memory constraints for reporting the fidelity and entanglement entropy which requires us to store the full quantum state. However, the variational method itself is

capable of simulating much larger graphs with a reasonable degree of entanglement entropy. For larger-scale simulations of QAOA with optimized parameters, compared with traditional simulation methods like tensor network models, the variational method does offer advantages in terms of memory requirement, as reported in Ref. [34].

Chapter 7

Distributionally Robust Variational Quantum Algorithms with Shifted Noise

7.1 Introduction

This chapter will introduce a parameter optimization approach for quantum algorithms under shifted noise.

Variational quantum algorithms (VQAs) [214] have the potential to demonstrate quantum advantage and have been applied in diverse fields, such as optimization [215, 216], finance [183, 217, 218], machine learning [219–221], quantum simulation [222–224], and chemistry [225–227]. However, parameter optimization is a substantial challenge for VQAs [228].

Numerous efforts have been made to optimize VQA parameters [186, 229–231]. One critical challenge for VQA parameter optimization is quantum noise [205, 232, 233], which limits their capabilities and introduces additional complexities to parameter optimization.

Modeling and mitigating hardware noise is a core part of Near-term Intermediate-scale Quantum (NISQ) algorithms [234–236]. Quantifying and improving the reliability and robustness of a VQA has been an important task and has gained increasing attention recently. To name a few, machine learning methods have been used to estimate the reliability of a quantum circuit [237]; noise-aware ansatz design methodologies [238] and robust circuit realization from a lower-level abstraction [189, 239] have also been investigated.

A more challenging yet practical problem is the sensitivity of quantum noise to the real-time environment. Suppose we have an accurate model of the quantum noise as a reference. However, the quantum noise can change significantly under different environmental conditions in real time, making the reference noise model inaccurate. Some studies [240, 241] have considered the reproducibility and stability under different noise models. We refer to this phenomenon of noise change as "noise shift".

In this chapter, we ask a fundamental question: *Can we optimize the VQA parameters such that they are robust to potentially shifted (unknown) noise?* We assume that we have access to a fixed noise model, but the actual noise level is an unknown random variable with an unknown PDF. This fixed PDF represents our limited knowledge about the potential noise shift.

To optimize VQA parameters under such unknown noise, for the first time, we propose a new min-max optimization formulation. Such an optimization formulation is called distributionally robust optimization (DRO) in the classical operation research community [158, 242–244]. DRO is an advanced optimization framework that aims to find solutions resilient against a range of possible probability distributions rather than a single expected distribution. In our context, we aim to optimize parameters based on the worst-case distribution of noise levels. This task, while distinct, complements error mitigation efforts. Rather than attempting to reduce quantum noise, our focus is on opti-

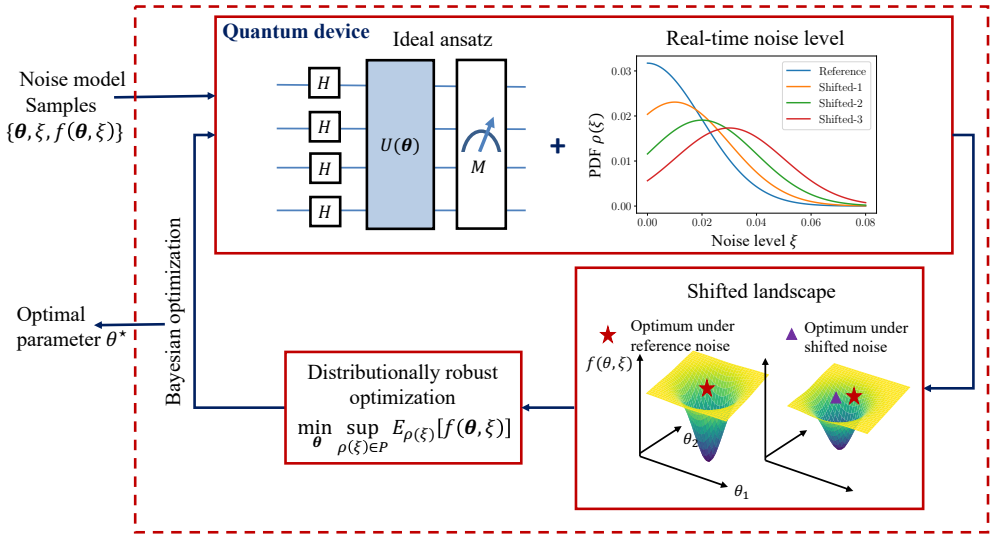

Figure 7.1: Overview of the distributionally robust variational quantum algorithms. Given an ideal ansatz and noise model, we assume the noise level is a random variable that can change in real time. We have samples of the noise level variable ξ from a reference distribution, ansatz parameter $\boldsymbol{\theta}$, and the corresponding VQA performance $f(\boldsymbol{\theta}, \xi)$. With the real-time noise, the VQA landscape and its optimum $\boldsymbol{\theta}$ can potentially change. Specifically, the optimal $\boldsymbol{\theta}$ under a certain noise level may not perform well under another noise level. Likewise, an optimal $\boldsymbol{\theta}$ under a reference noise level PDF may not perform well under another noise level PDF. To address the landscape shift, we reformulate the parameter optimization problem as a min-max formulation to find a robust parameter $\boldsymbol{\theta}$. In other words, we aim to optimize the performance under the worst-case noise level PDF. We use a distributionally robust Bayesian optimization solver to solve the new parameter optimization formulation, which is still handled by classical computers.

mizing parameters in the presence of potentially shifting noise. Various error mitigation techniques can be seamlessly integrated with our method.

Main contributions. In this chapter, we investigate the problem formulation, numerical solver, and validation of variational quantum algorithm training under unknown shifted noise. The overview is illustrated in Fig. 7.1. Our specific contributions include:

- To be robust against the real-time shifted noise, we formulate the problem of optimizing VQA algorithms as a distributionally robust optimization that aims to

optimize a targeted performance under the worst-case noise distribution.

- To solve the distributionally robust optimization, we model the unknown PDFs as a distributional uncertainty set that is defined by Maximum Mean Discrepancy (MMD). We then solve the min-max problem using a distributionally robust Bayesian optimization (DRBO) method [149, 151, 245]. Recently, Bayesian optimization (BO) has attracted attention in the field of quantum algorithm optimization. [246–251].

- We validate the proposed min-max formulation on two popular VQAs, including QAOA for the MaxCut problem and VQE with hardware-efficient ansatz for the one-dimensional Heisenberg model. The numerical results show that the proposed parameter optimization algorithm has better performance than conventional ones under the shifted noise.

7.2 Problem Formulation and Method

Variational quantum algorithms (VQAs) are a class of algorithms in quantum computing that utilize a hybrid approach, combining classical and quantum computing resources to solve computational problems. They are especially pertinent for use with Noisy Intermediate-Scale Quantum (NISQ) devices, which are the currently available quantum hardware.

The core idea of VQAs is to define a parameterized quantum circuit (ansatz) that manipulates the state of a quantum system in a way that depends on a set of parameters $\boldsymbol{\theta}$. Given an interested observable $\mathbf{O}$, these parameters are then optimized classically to minimize an objective function $\langle \boldsymbol{\psi}(\boldsymbol{\theta})|\mathbf{O}|\boldsymbol{\psi}(\boldsymbol{\theta})\rangle$, which is evaluated by the quantum system. However, due to the hardware noise, the actual ansatz and the resulting quantum

state $|\boldsymbol{\psi}(\boldsymbol{\theta},\xi)\rangle$ differ from the ideal one.

7.2.1 Distributionally robust optimization formulation of VQAs

We assume we have access to a fixed noise model and have estimations on its noise level, which follows a certain PDF $\xi \sim \rho(\xi)$. Let $f(\boldsymbol{\theta},\xi) = \langle\boldsymbol{\psi}(\boldsymbol{\theta},\xi)|\mathbf{O}|\boldsymbol{\psi}(\boldsymbol{\theta},\xi)\rangle$ be the interested quantity evaluated from an ansatz parametrized by $\boldsymbol{\theta}$ under a noise level ξ, then a standard variational quantum optimization algorithm becomes stochastic programming:

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}_{\rho(\xi)}[f(\boldsymbol{\theta},\xi)] \tag{7.1}$$

For simplicity, we consider the noise level for a single noise model as a scalar, but it can be seamlessly extended to a high-dimensional case.

However, due to the real-time noise, the PDF of the noise level ξ can shift and become unknown. As a result, we assume that $\rho(\xi)$ is not exactly known and it can be any PDF inside a set $\mathcal{P}$, which makes it impossible to obtain a deterministic value of $\mathbb{E}_{\rho(\xi)}[f(\boldsymbol{\theta},\xi)]$. As a result, we try to optimize the worst-case value of $\mathbb{E}_{\rho(\xi)}[f(\boldsymbol{\theta},\xi)]$ by solving

$$\min_{\boldsymbol{\theta}} \sup_{\rho(\xi)\in\mathcal{P}} \quad \mathbb{E}_{\rho(\xi)}[f(\boldsymbol{\theta},\xi)]. \tag{7.2}$$

When the uncertainty set degenerates to $\mathcal{P} = \{\rho(\xi)\}$, problem (7.2) degenerates to the standard stochastic optimization probelm in (7.1). On the other hand, the problem degenerates to a robust optimization under the worst noise level when the PDF of a noise level degenerates to a Dirac function.

The distributionally robust circuit optimization (7.2) may be intractable in practice because: (a) $\mathcal{P}$ may contain an infinite number of PDFs describing process variations; (b) the min-max problem is hard to solve by nature; (c) we do not have an analytical

form for $f(\boldsymbol{\theta}, \xi)$ under the presence of noise. In this section, we properly define the PDF uncertainty set $\mathcal{P}$ and solve problem (7.2) leveraging distributionally robust Bayesian optimization (DRBO) [149–151] developed recently in the machine learning community.

7.2.2 Distribution Uncertainty Set

We model the PDF uncertainty set $\mathcal{P}$ as a ball whose center is the nominal distribution $\rho_0(\boldsymbol{\xi})$ and radius ε is measured by a distribution divergence $\mathcal{D}$:

$$\mathcal{P} := \mathcal{B}(\rho_0) = \{\rho : \mathcal{D}(\rho_0, \rho) \leq \varepsilon\}. \tag{7.3}$$

There are many options for the divergence $\mathcal{D}$, including Maximum Mean Discrepancy, Wasserstein distance, φ-divergence, etc [158]. Here, we choose the Maximum Mean Discrepancy (MMD).

MMD aims to compare the means of samples drawn from two distributions in a high-dimensional reproducing kernel Hilbert space (RKHS) induced by a positive definite kernel function [252]. For the tractability of the problem, we discretize the noise level in a finite space Ξ with n parts. Then, let $\mathcal{H}_M$ be an RKHS with corresponding kernel $k_M : \Xi \times \Xi \to \mathbb{R}$, we can embed the distributions ρ_0 (similarly for ρ) into $\mathcal{H}_M$ via the mean embedding:

$$m_{\rho_0} := \mathbb{E}_{\xi \sim \rho_0}[k_M(\xi, \cdot)], \quad \text{such that}$$
$$\langle m_{\rho_0}, k_M(\xi', \cdot) \rangle = \mathbb{E}_{\xi \sim \rho_0}[k_M(\xi', \xi)], \forall \xi \in \Xi.$$

Then the MMD between two distributions ρ_0 and ρ over Ξ is defined as

$$\mathcal{D}(\rho_0, \rho) := \|m_{\rho_0} - m_\rho\|_{\mathcal{H}}, \tag{7.4}$$

where $\|\cdot\|_{\mathcal{H}} = \sqrt{\langle\cdot,\cdot\rangle}$ is the Hilbert norm. Let $w_i = \rho_0(\xi_i)$ and $w_i' = \rho(\xi_i)$ be the density probability of two discrete distributions, if we replace the expectation with the empirical expectation, i.e., $m_\rho = \sum_{i=1}^n w_i k_M(\xi_i, \cdot)$ and $m_{\rho'} = \sum_{i=1}^n w_i' w_i' k_M(\xi_i, \cdot)$, Eq. (7.4) can be written as:

$$\mathcal{D}(\rho_0, \rho) = \sqrt{(\mathbf{w} - \mathbf{w}')^T \mathbf{M} (\mathbf{w} - \mathbf{w}')}, \tag{7.5}$$

where $M_{ij} = k_M(\xi_i, \xi_j)$ is the kernel matrix.

7.2.3 DRO Main Workflow

By modeling the distribution uncertainty set defined via MMD, the DRO problem (7.2) becomes tractable. The main steps are summarized below.

- **Step 1.** Characterize the nominal noise distribution ρ_0.
- **Step 2.** Given a current $\boldsymbol{\theta}$, solve the inner problem to determine the worst-case PDF of ξ:

$$\sup_{\rho(\boldsymbol{\xi}):\mathcal{D}(\rho_0,\rho)\leq\varepsilon} \mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\boldsymbol{\theta},\xi)] = \max_{\substack{\mathbf{w}':\|\mathbf{w}'\|_1=1,\\ 0\leq w_j'\leq 1,\forall j\in[n],\\ \|\mathbf{w}'-\mathbf{w}\|_M\leq\varepsilon}} \langle \mathbf{w}', f_{\boldsymbol{\theta}}\rangle, \tag{7.6}$$

  where $f_{\boldsymbol{\theta}} := f(\boldsymbol{\theta}, \cdot) \in \mathbb{R}^n$ is the output with a given parameter $\boldsymbol{\theta}$.

- **Step 3.** Solve the outer problem to update $\boldsymbol{\theta}$.

$$\min_{\boldsymbol{\theta}} \quad \langle \mathbf{w}', f_{\boldsymbol{\theta}}\rangle \tag{7.7}$$

- **Step 4.** If not converge, go back to Step 2.

Specifically, Step 2 can be solved analytically via convex programming. Step 3 can be solved via a numerical optimizer. However, one of the challenges in steps 2 and 3 is that we need to simulate multiple $f(\boldsymbol{\theta}, \xi)$, which can be expensive in practice. To address the

Algorithm 6 Overall DRBO algorithm with GP

Input: Initial sample set $\mathcal{S}_0 = \{\boldsymbol{\theta}^i, \xi^i, f(\boldsymbol{\theta}^i, \xi^i)\}_{i=1}^M$, reference PDF of noise level $\rho_0(\xi)$ with $\rho_0(\xi_i) = w_i, \forall i = [n]$, uncertainty ball radius ε, maximum iteration T

Output: The optimal circuit design variables $\boldsymbol{\theta}^\star$

1: **for** $t = 1, 2, ..., T$ **do**
2: Construct a GP model as the probabilistic surrogate model $\hat{f}(\boldsymbol{\theta}, \xi) = \mathcal{GP}(\boldsymbol{\theta}, \xi)$ based on $\mathcal{S}_{t-1}$
3: Define $\mathrm{LCB}(\boldsymbol{\theta}, \xi) := \mu(\mathcal{GP}(\boldsymbol{\theta}, \xi)) - \beta \cdot \sigma(\mathcal{GP}(\boldsymbol{\theta}, \xi))$
4: Define the PDF of the worst-case distribution $\mathbf{w}' := \mathrm{argmax}_{\mathbf{w}} \langle \mathbf{w}', \mathrm{LCB}(\boldsymbol{\theta}, \xi) \rangle$ s.t. $\mathbf{w}' : \|\mathbf{w}'\|_1 = 1$, $0 \leq w'_j \leq 1, \forall j \in [n]$, and $\|\mathbf{w}' - \mathbf{w}\|_M \leq \varepsilon$
5: Solve the robust parameter $\boldsymbol{\theta}_t = \mathrm{argmin}_{\boldsymbol{\theta}} \langle \mathbf{w}', \mathrm{LCB}(\boldsymbol{\theta}, \xi) \rangle$
6: Sample K noise levels from the reference PDF $\boldsymbol{\xi}_k \sim \rho_0$ and simulate $f(\boldsymbol{\theta}_t, \xi_k)$, for $k = 1, 2, \ldots, K$
7: Augment data set $\mathcal{S}_t \leftarrow \mathcal{S}_{t-1} \cup \{(\boldsymbol{\theta}_t, \xi_t, f(\boldsymbol{\theta}_t, \xi_k))\}_{k=1}^K$
8: **end for**
9: Return optimal $\boldsymbol{\theta}^\star$

computational issue, we apply a Bayesian optimization solver to the workflow. The key idea is to sequentially learn a surrogate model of $f(\boldsymbol{\theta}, \xi)$ and optimize it by iteratively adding informative samples.

7.2.4 BO solver for DRO problem

Next, we explain how to solve DRO via Bayesian optimization with a few quantum circuit simulations. Bayesian optimization sequentially builds a probabilistic surrogate model of $f(\boldsymbol{\theta}, \xi)$ and explores the design space by minimizing an acquisition function. The overall DRBO algorithm is summarized in Algorithm 4.

We first construct a probabilistic surrogate model $\hat{f}(\boldsymbol{\theta}, \xi)$, which can estimate both the output and its uncertainty given an input $(\boldsymbol{\theta}, \xi)$. Here, we use the Gaussian process regression model $\mathcal{GP}(\boldsymbol{\theta}, \xi)$ as the surrogate $\hat{f}(\boldsymbol{\theta}, \xi)$. Then we use its lower confidence bound (LCB) to replace the original objective function $f(\boldsymbol{\theta}, \xi)$ in Eqs. (7.6) and (7.7) in steps 2-4

$$f(\boldsymbol{\theta}, \xi) \to \mathrm{LCB}(\boldsymbol{\theta}, \xi) = \mu(\mathcal{GP}(\boldsymbol{\theta}, \xi)) - \beta \cdot \sigma(\mathcal{GP}(\boldsymbol{\theta}, \xi)), \tag{7.8}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the estimated mean and standard deviation, and β is a parameter to balance the model exploitation and exploration.

Gaussian process surrogate To build the Gaussian process regression model, we need to predefine the mean function $m(\cdot)$ and the kernel function $k_{\mathcal{GP}}(\cdot,\cdot)$. Given a dataset $\mathbf{X} = \{\mathbf{x}^i\}_{i=1}^M = \{\boldsymbol{\theta}^i, \xi^i\}_{i=1}^M$ and their simulation outputs $\mathbf{y} = \{f(\mathbf{x}^i) + \epsilon\}_{i=1}^M$, the GP model assumes that the simulation outputs follows a Gaussian distribution [160]:

$$\text{Prob}(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \mathbf{K}), \tag{7.9}$$

where $\boldsymbol{\mu} \in \mathbb{R}^M$ is the mean vector with $\mu_i = m(\mathbf{x})$, $\mathbf{K} \in \mathbb{R}^{M\times M}$ is the covariance matrix with $K_{i,j} = k_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_j)$. The measurement noise is characterized as a white noise ϵ in the simulation output.

Then, the GP model can offer a probabilistic prediction of a new data $\mathbf{x}'$, $\mathcal{GP}(\mathbf{x}') \sim N(\mu(\mathbf{x}'), \sigma^2(\mathbf{x}'))$, as follows:

$$\begin{aligned}
\mu(\mathbf{x}') &= \mathbf{k}_{GP}(\mathbf{x}', \mathbf{X})^T (\mathbf{K} + \epsilon^2\mathbf{I})^{-1}\mathbf{y} \\
\sigma^2(\mathbf{x}') &= k_{\mathcal{GP}}(\mathbf{x}', \mathbf{x}') - \mathbf{k}_{\mathcal{GP}}(\mathbf{x}', \mathbf{X})^T (\mathbf{K} + \epsilon^2\mathbf{I})^{-1}\mathbf{k}_{\mathcal{GP}}(\mathbf{X}, \boldsymbol{\theta}')
\end{aligned}$$

In our cases, we choose the prior mean as $m(\mathbf{x}) = 0$ and use the RBF kernel $k_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x})j\|^2}{2l^2}}$. Note that such a kernel $k_{\mathcal{GP}}$ in GPR is different from the kernel function k_M embedded in MMD. Beyond the Gaussian process, other surrogate models are also possible.

Optimal selection Regarding the selection of an optimal solution, it terms out to be non-trivial. Since we want to estimate the expectation over the noise distribution, it will be too expensive to estimate with real quantum devices. Instead, we choose the solution

with maximized model posterior, i.e., we choose the $\boldsymbol{\theta} = \text{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{\rho(\xi)}[\mu(\hat{f}(\boldsymbol{\theta}, \xi))] = \langle \mathbf{w}, \mu(\hat{f}_{\boldsymbol{\theta}}) \rangle$, where $\hat{f}_{\boldsymbol{\theta}}$ is the mean prediction from $\hat{f}$ with a given parameter $\boldsymbol{\theta}$. It is a common strategy for similar conditions [162, 163].

In addition, an accurate estimation over $\hat{f}(\boldsymbol{\theta}, \cdot)$ will benefit our model output. For this motivation, at the end of each iteration in Algorithm 4, we can add a batch of samples of ξ from Ξ in step 3. It can help build a better probabilistic model and fasten the BO solver convergence.

Remarks One possible further improvement is to treat the BO solver as a warm-start procedure. After returning a few high-quality solutions from BO, we can conduct the local numerical optimization by taking them as the initial. The local search step may introduce additional computational cost and need more calling of $f(\boldsymbol{\theta}, \xi)$ instead of the surrogate model, but it can lead to a potentially better solution. The hybrid of different solvers is also a common strategy in VQA parameter optimization [249, 253].

The proposed distributionally robust optimization can easily degenerate into stochastic optimization or robust optimization. Stochastic optimization, namely Eq. (7.1), does not consider the real-time change of the noise. Robust optimization degenerates the PDF of the noise level $\rho(\xi)$ to a single scalar. This case can easily lead to over-conservative parameter optimization.

The current distribution uncertainty set modeling of Maximum Mean Discrepancy has the great power of capturing the worst-case distribution. However, we need to discretize the noise level PDF in order to efficiently estimate the MMD. We are aware of the f-divergence modeling [151] of the distributional shift that does not distritize the noise level, but it performs poorly in our experiments due to the unfitted modeling of shifted noise level distribution.

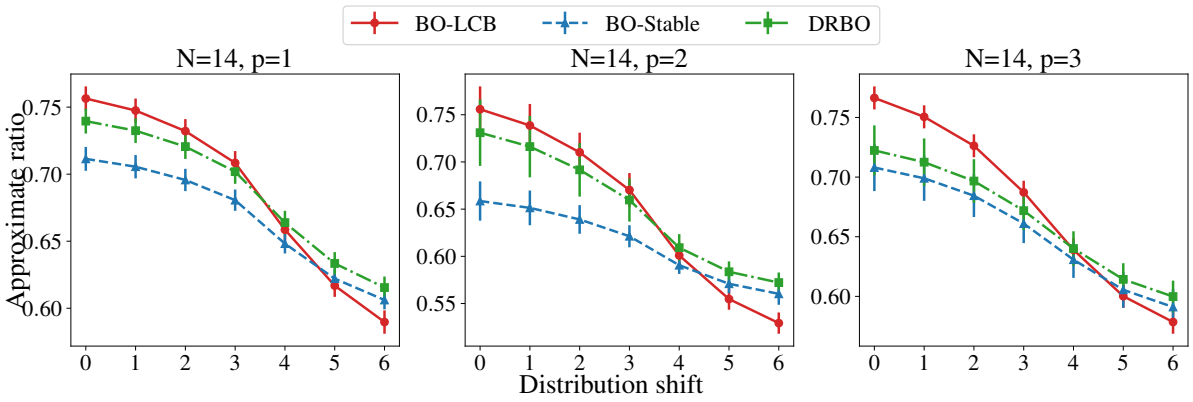

Figure 7.2: The results for solving $N = 14$ 3-regular graph MaxCut problems via QAOA. The x axis denotes the significance of noise shift, where the noise level PDF is the reference one at $x = 0$. The y axis is the expectation of the approximate ratio of the QAOA solution evaluated at different noise PDF. We report the average result over 10 non-isomorphic graphs. As we can see, the standard BO-LCB solution has the best performance under the reference noise. However, under an increasingly shifted noise, the DRBO solution begins to outperform the BO-LCB. Meanwhile, the BO-Stable solution is over-conservative with respect to the noise. It significantly scarifies the performance under the reference PDF and the slightly shifted PDFs to gain an improvement under significant shifts. These observations are consistent in the experiments with different QAOA depths.

7.3 Numerical Results

We validate the distributionally robust formulation of optimizing VQA parameters (7.2) in two widely used VQA applications: one is using QAOA for MaxCut and the other one is using VQE for one-dimensional Heisenberg model.

Here we conduct the numerical experiments on a simulator in order to adjust the noise level easily and correspondingly to validate the method. To apply the distributional robustness formulation in hardware experiments, the estimation of the noise model and noise level is another challenge, which is out of the scope of this work.

Baseline We compared the proposed DRBO solver to a standard Bayesian optimization for solving stochastic optimization, BO-LCB [165], and a robust Bayesian optimization,

BO-Stable [254]. In BO-LCB, we target problem 7.1 with a fixed reference distribution of noise level $\rho_0(\xi)$ using a Bayesian optimization approach. We use the same GP surrogate model and its LCB as Eq. (7.8), but without solving the outer problem (7.7). Specifically, the lines 4 and 5 of Algorithm 4 are combined as solving $\min_{\boldsymbol{\theta}} \langle \mathbf{w}, \mathrm{LCB}(\boldsymbol{\theta}, \xi) \rangle$.

In BO-Stable, we target a shift-aware problem but only focus on the worst noise level instead of the worst distributional noise level. We use the same GP surrogate model and its LCB as Eq. (7.8). Differing from using DRBO for problem (7.2), the lines 4 and 5 of Algorithm 4 are replaced with solving $\min_{\boldsymbol{\theta}} \mathrm{LCB}(\boldsymbol{\theta}, \xi^\star)$ where given a $\boldsymbol{\theta}$, the worst $\xi^\star$ is defined as $\xi^\star := \mathrm{argmax}_\xi \, \mathrm{LCB}(\boldsymbol{\theta}, \xi)$.

To access different VQA parameter optimization methods, we obtain different parameter solutions and evaluate them under different noise level distributions $\rho(\xi)$.

7.3.1 Experiments on QAOA

QAOA is a leading variational quantum algorithm for combinatorial optimization problems. It alternatively applies two operators, a phase-separation operator and a mixer operator, to drive a quantum system to the target solution state. Without hardware noise, a QAOA solution is denoted as $\boldsymbol{\psi}(\boldsymbol{\theta}) = e^{-i\beta_p H_M} e^{-i\gamma_p H_P} \cdots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_P} |\psi_0\rangle$.

We will take the MaxCut problem as a case study of QAOA. Given a graph $G = (V, E)$ with vertices V and edges E, the MaxCut problem aims to find a cut that partitions the graph vertices into two sets with the largest number of edges. Its cost function is written as

$$C = \sum_{(i,j) \in E} 1 - s_i s_j, \tag{7.10}$$

where s_i and s_j are binary variables associated to the vertices in V, which assume value 1 or -1 depending on which of the two partitions defined by the cut are assigned. Its cost Hamiltonian is defined as $\mathbf{H}_C = \sum_{(u,v) \in E} \frac{1}{2}(\mathbf{I} - \mathbf{Z}_i \mathbf{Z}_j)$, where $\mathbf{Z}_i$ denotes a Pauli-Z

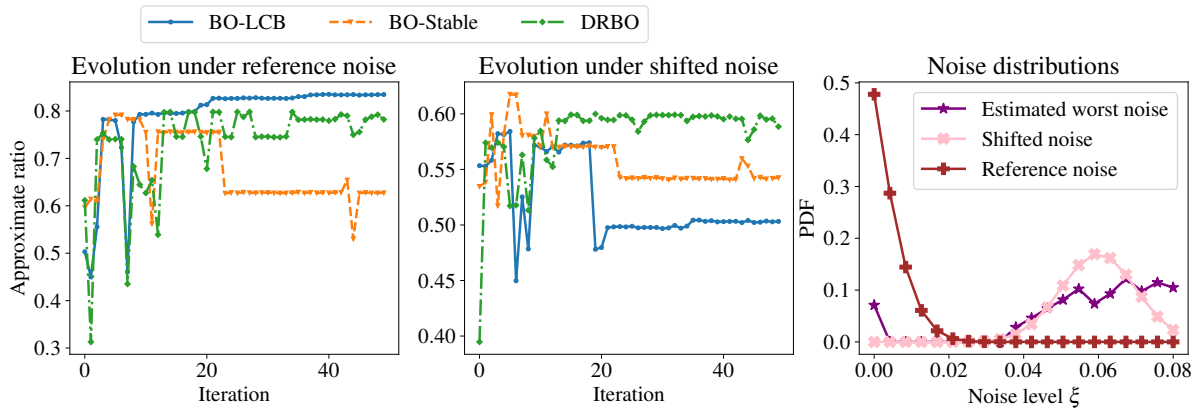

Figure 7.3: One example of evolving of solution in different BO algorithms. The x axis is the iterations in a BO algorithm, and the y axis is the expectation of cost function evaluated over noise level at a $\boldsymbol{\theta}$. The evaluated $\boldsymbol{\theta}$ at one iteration is obtained by maximizing the model posterior, which is unnecessary to be the explored $\boldsymbol{\theta}$ at that iteration. Under the reference noise PDF, the BO-LCB algorithm converges to a better solution, while the DRBO converges to a better solution under the shifted noise. The rightmost figure shows the example PDFs of the reference noise level, shifted noise level, and the estimated worst-case noise level from the DRBO algorithm.

operator.

In applying QAOA for solving the MaxCut problem, given a noise model with noise level ξ, we aim to optimize the QAOA parameters $\boldsymbol{\theta} = (\boldsymbol{\gamma}, \boldsymbol{\beta})$ such that the resulting solution $|\boldsymbol{\psi}(\boldsymbol{\theta}, \xi)\rangle$ has minimal energy $f(\boldsymbol{\theta}, \xi) = \langle \boldsymbol{\psi}(\boldsymbol{\theta}, \xi)|\mathbf{H}_C|\boldsymbol{\psi}(\boldsymbol{\theta}, \xi)\rangle$. Considering the uncertainty and real-time shift of noise level, we aim to find parameters $\boldsymbol{\theta}$ that make QAOA performance more robust towards the real-time noise by solving the DRO problem (7.2).

Here, we discretize the noise level into 20 bins in $[0, 0.08]$ evenly. We assume the reference noise follows a truncated Gaussian distribution, with the real-time shift may shift its mean to a larger value. To begin with, for a depth-p QAOA ansatz, we initialize the sampling set $\mathcal{S}_0 = \{\boldsymbol{\theta}^i, \xi^i, f(\boldsymbol{\theta}^i, \xi^i)\}_{i=1}^M$ by taking $M = 20p$, where $\boldsymbol{\theta}^i$ is drawn from the design space based on a Latin hypercube approach [255], and the noise level samples ξ^i are drawn from the reference distribution $\rho_0(\xi)$. We set the maximum BO iterations

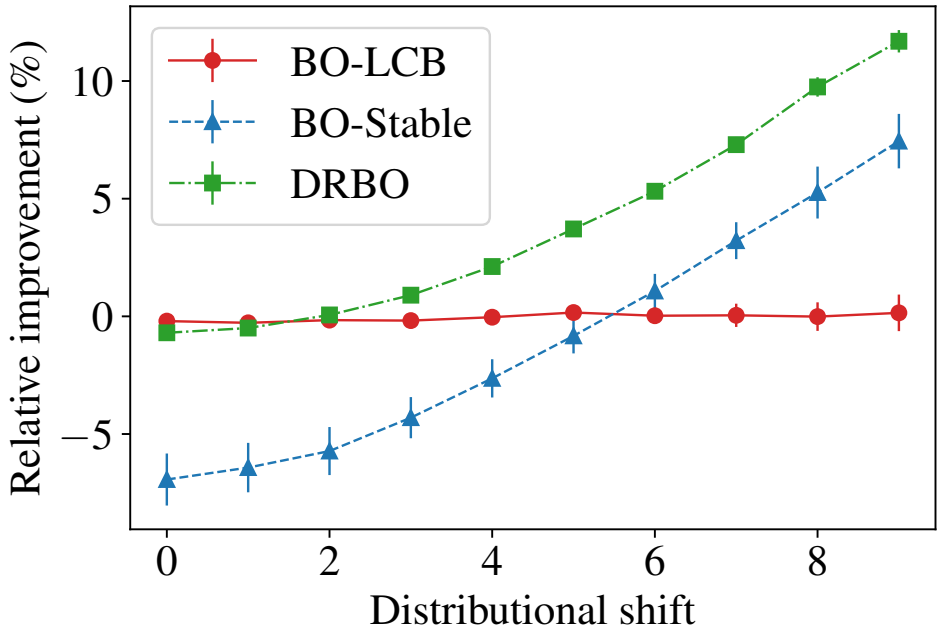

Figure 7.4: The results for solving the ground energy of a 6-spin, $J = 1, B = 0.2$ one-dimensional Heisenberg model via VQE with two-layer hardware efficient ansatz. The x axis denotes the significance of noise shift, where the noise level PDF is the reference one at $x = 0$. We first obtained the optimal parameter θ_0 in a noiseless simulation, which solves the problem perfectly with $f(\boldsymbol{\theta_0}) = -4.8$. Then we report the relative improvement of the energy $\frac{\mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\boldsymbol{\theta},\xi)] - \mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\boldsymbol{\theta_0},\xi)])}{\mathbb{E}_{\rho(\boldsymbol{\xi})}[f(\boldsymbol{\theta_0},\xi)]}$. Under all the shifted distributions, BO-LCB performs close to $\boldsymbol{\theta_0}$. DRBO scarifies limited performance under mild noise and performs much better than the BO-LCB and noiseless optimal $\boldsymbol{\theta_0}$ in significantly shifted noise. While the BO-Stable can also find the robust parameter under the shifted noise, it performs not as well as DRBO, especially when the noise shift is mild.

as $T = 20p$.

As shown in Fig. 7.2, we evaluate different BO-based parameter optimization results on 10 graphs with degree-3 and graph size $N = 14$. We report the average approximate ratio results under different shifted noise levels. The x-axis denotes the index of the levels of noise shift, with a higher one denoting a more significant shift, and index-0 denotes the reference noise. Since we solve the optimal $\boldsymbol{\theta}$ under shifted noise, the DRBO-solved QAOA is expected to perform worse than the one solved from a standard BO solver under the reference noise. However, as the noise shift becomes more and more significant, the

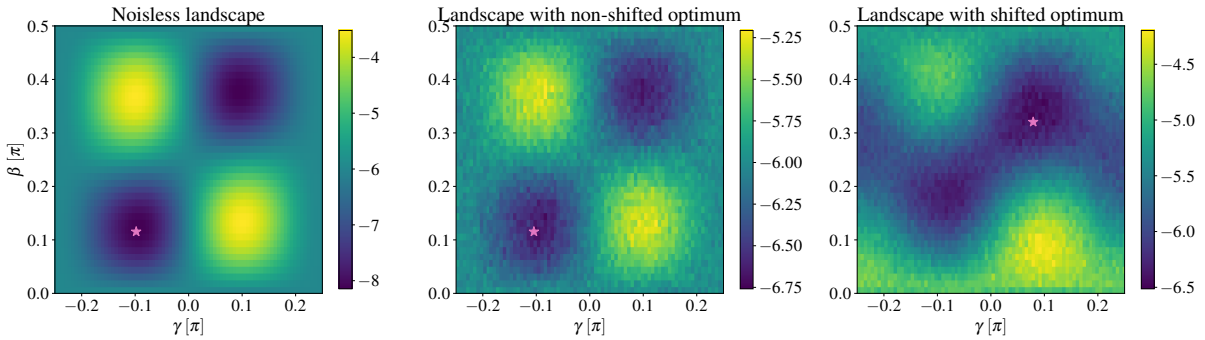

Figure 7.5: Example of energy landscapes under different noise models. The left heatmap is the depth-1 QAOA landscape for the MaxCut problem. The color denotes the solved energy. The optimal point in this landscape is highlighted as a triangle. The middle heatmap is the landscape under simple Pauli errors, which has been shown not to change the VQA optimal and uniformly flatten the landscape. The right heatmap is the landscape under the phase and amplitude damping noise, where the optimum is shifted and the energy landscape has a different shape. Under the noise, both the middle and the right ones have worse energies than the left noise-free landscape.

DRBO solution begins to show its advantages. Notably, for the BO-Stable, it performs better than BO-LCB under a significantly shifted noise as well. However, it is also over-conservative under the reference noise since it only considers a single worst noise level. The results and observations are consistent over different QAOA depths.

We plot the solution during the BO iterations in Fig. 7.3. During the iteration, the performance is evaluated under the optimal solution selected from the maximum posterior rather than from the solution of an acquisition function. We show the performance evaluated under the reference noise and the shifted noise. During the iterations, DRBO consistently converges to a shifted-noise preferred solution while the LCB converges to a reference-noise preferred one. We also show the PDFs of the reference noise, shifted noise, and the worst-case estimated from the DRBO algorithm. We can see that the MMD approach successfully captures the shifted noise under the worse-case distribution, enabling the DRBO to explore the parameters space that performs better under shifted noise.

7.3.2 Experiments on VQE

Variational quantum eigensolver (VQE) is another popular VQA, specifically designed to simulate quantum systems and find the ground state energy of quantum systems. We use the VQE algorithm with a hardware-efficient ansatz [256] for simulating the ground energy of a one-dimensional Heisenberg model defined as:

$$\mathbf{H} = J \sum_i \mathbf{X}_i \mathbf{X}_{i+1} + \mathbf{Y}_i \mathbf{Y}_{i+1} + \mathbf{Z}_i \mathbf{Z}_{i+1} + B \sum_i \mathbf{Z}_i, \tag{7.11}$$

where J is the strength of the spin–spin interaction and B is the magnetic field along the Z direction. Here, we use a hardware-efficient ansatz to implement the VQE algorithm (see more details in the supplementary). Given an ansatz parameterized by $\boldsymbol{\theta}$ and under a noise model with noise level ξ, we denote the state as $|\boldsymbol{\psi}(\boldsymbol{\theta}, \xi)\rangle$. The VQE algorithm cost function is defined as $f(\boldsymbol{\theta}, \xi) = \langle \boldsymbol{\psi}(\boldsymbol{\theta}, \xi) | \mathbf{H} | \boldsymbol{\psi}(\boldsymbol{\theta}, \xi) \rangle$. Here, we aim to find the ansatz parameters that lead to robust performance under shifted noise by solving the DRO problem (7.2).

Here, for a layer-2 hardware-efficient ansatz, we discretize the noise level into 20 bins in $[0, 0.08]$ evenly. We assume the reference noise follows a truncated Gaussian distribution, and the real-time PDF may shift its mean to a larger value.

In Fig. 7.4, we show the ground energy solved for a 6-spin system with $J = 1, B = 0.2$, whose ground state is highly entangled. Aiming to optimize the last layer parameters, we initialize the sampling set with $M = 40$ and set the maximum BO iteration $T = 40$ as well. We reported the results from a two-layer hardware-efficiency ansatz with a more detailed setup and results in Supplementary. Similar to the QAOA results, DRBO performs better than BO-LCB under a significantly shifted noise. Additionally, the DRBO solution almost does not scarify the reference noise performance. In the meanwhile, Bo-Stable solves an over-conservative result.

7.4 Discussion

7.4.1 Landscape shift

Ref. [257] has shown that the optimal variational parameters are unaffected by a broad class of noise models, such as measurement noise, gate noise, and Pauli channel noise. This phenomenon is called optimal parameter resilience. Meanwhile, some noise can shift the location of minima. A rich of work has studied how quantum noise can influence the VQA landscape [258–260]. We highlight that the shift location of optimal parameters motivates our work, i.e., given that the optimal parameter will change under different (shifted) noise, we aim to find a parameter with robust performance under the shifted noise environment.

In our simulation, we use the phase and amplitude damping noise model, which has been shown to change the values of optimal parameters. The landscape with a changed or unchanged optimum is illustrated in Fig. 7.5.

7.4.2 Variant problem formulation

Radius varying formulation Beyond optimization under real-time noise, another case where the distributionally robust optimization can be applied is as follows: we do not have a precise enough estimation toward the real noise level distribution as the reference distribution. As we collect more data on the noise level, we can have a more accurate estimation of its PDF. Therefore, as the iteration continues, we can gradually refine the center $\rho_0(\xi)$ of the uncertainty ball and reduce its radius ε.

Gate error modeling Beyond modeling hardware noise, another possible modeling is on the gate error $f(\boldsymbol{\theta} + \boldsymbol{\xi})$ in the parameterized quantum circuit realization, which assumes that the gate parameters are not exactly implemented but suffer from some

errors. In such a formulation, under different error levels of $\boldsymbol{\xi}$, the optimal $\boldsymbol{\theta}$ will have different values. The DRO formulation can optimize the VQA to find parameters that are robust to the shifted gate errors. A robustness analysis of such a formulation is discussed in [261].

7.5 Conclusion

Noise has been a major obstacle to the applications of near-term quantum computers, especially the variational quantum algorithms (VQAs). Various error mitigation techniques have been intensively studied. However, one of the challenges of noise modeling and error mitigation is that the quantum noise is usually changing as time evolves. Consequently, a VQA with optimized parameters may perform poorly in another noise environment.

In this chapter, we have presented a distributionally robust optimization formulation to optimize VQA parameters that are more robust under shifted quantum noise. The proposed formulation is efficiently handled by a distributionally robust Bayesian optimization (DRBO) solver. We validate the proposed method in two popular VQA benchmarks, QAOA for MaxCut and VQE with hardware-efficient ansatz for the one-dimensional Heisenberg model. The proposed distributionally robust optimization formulation does not mitigate the inherent quantum noise, but it fights against the noise at the algorithm level. It can be potentially integrated with various error mitigation techniques to improve the VQA robustness further.

Our formulation can be more impactful in large-size problems. When scaling to large-size problems or a deeper VQA ansatz, a smaller noise level may influence performance. For example, a larger problem needs a deeper circuit depth to implement QAOA. In this case, a small shift in the noise level can propagate and accumulate and potentially impact

the VQA performance. Therefore, the VQA parameter optimization under a shift can be more important.

To integrate the proposed distributionally robust formulation into more practical use cases, a better knowledge of the noise models is highly desired since we currently only model the variations of noise levels. To improve the distributionally robust optimization solver, some better techniques that do not need to discretize the noise level or efficiently handle high-dimensional parameter optimization can be developed.

We also have applied a similar distributionally robust optimization formulation for classical circuit optimization [31] in Chapter 3, where we identified the shifts of process variations. In this chapter, we focus on handling the parameter optimization of noisy variation quantum algorithms and highlight the challenge of real-time noise.

Chapter 8

Conclusion and Discussion

As we approach Moore's law's end, the process variations have been increasingly significant in the integrated circuit design. Meanwhile, researchers are actively searching for alternative unconventional computing approaches, where quantum computing has been a promising computing method due to its great computational power in nature and the recent great success in quantum hardware.

In this dissertation, we have discussed the uncertainty-aware design automation of both classical circuits and quantum computing. While the design abstraction can be categorized into different levels, we mainly focus on the circuit level in classical circuits and the block/gate level in quantum computing. However, we focus on the simulation, modeling, and optimization for both classical and quantum circuits.

In the classical part, we were motivated by the issue of process variations in the nano-size design. In Chapter 2, we first developed a tensor regression-based uncertainty quantification model for the high dimensional variations. The uncertainty quantification model is a polynomial surrogate model for the circuit performance, allowing us to fast simulate the circuit under variations. In Chapter 3, we highlight an over-conservative design issue in yield optimization and propose a chance-constrained yield-aware optimiza-

tion formulation. To solve the stochastic programming with probabilistic constraints, we leverage a polynomial bounding approach to upper bound the constraint violation probabilities and convert the probabilistic constraints into a certified deterministic one. Together with a polynomial-based performance modeling, the overall yield-aware problem can be solved using a global polynomial optimizer. Next, in Chapter 4, we highlight another fundamental issue in the uncertainty-aware design: the probability density function modeling of variations could be imprecise and differ from the practice. We name this issue as variational shift. For the first time, we propose a shift-aware circuit optimization problem with a min-max form, where we search for a design under the worst-case distribution of variations. We leverage an efficient Bayesian optimization solver with φ-divergence modeling on the uncertainty set of shifted variations.

In quantum computing, many new challenges differ from classical circuit simulation and optimization. First, the classical simulation of quantum systems is challenging in principle due to the exponentially growing complexity of a quantum system. In Chapter 6, we studied a variational method for simulating quantum approximation optimization algorithms. We show that the circuit entanglement entropy is a bottleneck for the variational simulation method, similar to other classical simulation methods like tensor network models. It indicates that the simulation-friendly circuit regime for the variational method is limited. In Chapter 7, we looked into the parameter optimization of variational quantum algorithms under real-time noise. Quantum noise has been one of the biggest obstacles to near-term quantum computing applications. Additionally, the noise from the environment interactions changes in real time. We borrow a similar idea from Chapter 4 to formulate a parameter optimization framework under real-time noise by modeling the noise level as a random variable. We model the noise level density uncertainty set using the Maximum Mean Discrepancy and solve the min-max problem with a Bayesian optimization approach.

In future works, design automation of quantum computing could have long-term interests. With the growth of quantum hardware, we are approaching to demonstrate the quantum advantage or utility with near-term devices. Meanwhile, the final goal is to develop a fault-tolerant quantum computer, which could have a revolutionary impact on our life. Computer-aided design tools will play an essential role in this path, including circuit simulation, quantum circuit optimization and compilation, circuit verification and certification, error mitigation and correction methodologies, etc. Although this dissertation mainly focuses on block/gate-level quantum computing and design, the cross-design-level end-to-end design will be essential for the success of both near-term and fault-tolerant quantum computing.

Appendix A

Appendices

A.1 Chapter 2

A.1.1 Proof of Lemma 1

We consider two cases $\alpha \in (0, 2)$ [82] and $\alpha = 2$ [70].

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 $y_r \neq 0$, $\lim_{\eta_r \to \infty} \kappa(\boldsymbol{\eta}) = \infty$ 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

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

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.

A.1.2 Detailed Derivation of Eq. (2.21)

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. (2.21).

A.2 Chapter 3

A.2.1 Solution to Optimal Polynomial Kinship (3.8)

Given a ρ-order optimal kinship function $\kappa_\rho(\cdot)$, we introduce two positive semidefinite matrices $\mathbf{Y}_1 \in \mathbb{R}^{(n_1+1)\times(n_1+1)}$ and $\mathbf{Y}_2 \in \mathbb{R}^{(n_2+1)\times(n_2+1)}$ with $n_1 = \lfloor(\rho-1)/2\rfloor$ and $n_2 = \lfloor(\rho-2)/2\rfloor$. We further define two series of Hankel matrices $\mathbf{H}_{1,m} \in \mathbb{R}^{(n_1+1)\times(n_1+1)}$ and $\mathbf{H}_{2,m} \in \mathbb{R}^{(n_2+1)\times(n_2+1)}$ as

$$\mathbf{H}_{k,m}(i,j) = \begin{cases} 1, & i+j = m+1 \\ 0, & \text{otherwise} \end{cases}, k = 1, 2. \tag{A.2}$$

Based on the sum-of-square representation of a nonnegative univariate polynomial, we can reformulate (3.8) as a finite-dimensional semidefinite programming (A.3), which can be handled by many efficient solvers and toolboxes [262]. The detailed proof will be similar to the Corollary 1 of [132], where the difference is the order of nonnegative polynomials.

$$\begin{gathered} \min_{\zeta_0,\ldots,\zeta_\rho,\mathbf{Y}_1,\mathbf{Y}_2} \ \sum_{i=0}^{\rho} \frac{(-1)^i}{i+1}\zeta_i \\ \text{s.t. } \zeta_0 = 1, \\ \sum_{i=0}^{\rho}(-1)^i\zeta_i = 0, \\ \mathrm{Tr}(\mathbf{Y}_1\mathbf{H}_{1,m}) + \mathrm{Tr}(\mathbf{Y}_2\mathbf{H}_{2,m}) = \\ \sum_{i=m+1}^{\rho} \frac{i!(-1)^{i-m-1}}{k!(i-m-1)!}\zeta_i, m = 0, 1, \ldots, \rho-2, \\ \mathbf{Y}_k \succeq 0, \ k = 1, 2. \end{gathered} \tag{A.3}$$

We list some examples of the solved polynomial as below: for $\rho = 5$, the polynomial coefficients are $[1, 7.87, 24.62, 37.49, 27.74, 8.00]$. For $\rho = 8$, the polynomial coefficients are $[1, 13.4, 75, 223.71, 384.3, 381.57, 203.57, 45.19]$.

A.2.2 Details about Benchmark Setup

In the MZI benchmark (Sec. 3.5.2), the process variations on the coupling coefficients are described by a truncated Gaussian mixture model with two components:

$$\mu(\xi) = \frac{1}{2}\mathcal{TN}_1(\bar{\boldsymbol{\mu}}_1, \boldsymbol{\Sigma}_1, \mathbf{a}_1, \mathbf{b}_1) + \frac{1}{2}\mathcal{TN}_2(\bar{\boldsymbol{\mu}}_2, \boldsymbol{\Sigma}_2, \mathbf{a}_2, \mathbf{b}_2), \tag{A.4}$$

where $\bar{\boldsymbol{\mu}}_1 = -\bar{\boldsymbol{\mu}}_2 = [3,3,3]^T$, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = 3^2 \begin{bmatrix} 1 & 0.4 & 0.1 \\ 0.4 & 1 & 0.4 \\ 0.1 & 0.4 & 1 \end{bmatrix}$, $\mathbf{a}_1 = -\mathbf{b}_2 = [-6,-6,-6]^T$, $\mathbf{a}_2 = -\mathbf{b}_1 = [-12,-12,-12]^T$.

In the microring benchmark (Sec. 3.5.3), the process variations on the coupling coefficients are described the same as Eq. (A.4) with different parameters $\bar{\boldsymbol{\mu}}_1 = -\bar{\boldsymbol{\mu}}_2 = [0.03, 0.03, 0.03, 0.03]^T$, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = 0.03^2 \begin{bmatrix} 1 & 0.4 & 0.1 & 0.4 \\ 0.4 & 1 & 0.4 & 0.1 \\ 0.1 & 0.4 & 1 & 0.4 \\ 0.4 & 0.1 & 0.4 & 1 \end{bmatrix}$,

$\mathbf{a}_1 = -\mathbf{b}_2 = [-0.06,-0.06,-0.06,-0.06]^T$, $\mathbf{a}_2 = -\mathbf{b}_1 = [-0.12,-0.12,-0.12,-0.12]^T$.

A.2.3 Additional Case Study

In this case study, for the same microring add-drop filter in section 3.5.3, we aim to optimize its bandwidth while holding a constraint on the bandwidth as well. The yield-aware chance-constrained design is formulated as

$$\begin{aligned} \max_{\mathbf{x}\in\mathbf{X}} \quad & \mathbb{E}_{\boldsymbol{\xi}}[\mathrm{BW}(\mathbf{x}, \boldsymbol{\xi})] \\ \text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\xi}}(\mathrm{BW}(\mathbf{x}, \boldsymbol{\xi}) \leq \mathrm{BW}_0) \geq 1 - \epsilon, \end{aligned} \tag{A.5}$$

Table A.1: Bandwidth-Constrained Optimization Results for Microring Add-drop Filter

Risk level ϵ	Method	$\mathbb{E}_{\boldsymbol{\xi}}$[BW] (GHz)	Δ (%)	Yield (%)	Simulation #
0.05	Moment [25]	N/A*	N/A*	N/A*	65
	Proposed	**105.66**	**3.89**	98.7	65
0.07	Moment [25]	93.96	7.53	100	65
	Proposed	**107.28**	**4.09**	96.8	65
0.1	Moment [25]	98.88	11.11	100	65
	Proposed	**109.01**	**3.67**	93.3	65
N/A	BYO [104]	102.05	N/A	99.8	2020

* The algorithm fails with no feasible solution.

where the bandwidth threshold BW_0 is 120 GHz. The design solutions and comparisons are summarized in Table A.1. Similar to the cases in section 3.5, our method still works when the objective and constraint are the same quantity and outperform the other approaches.

A.3 Chapter 6

A.3.1 Details of variational simulation for QAOA

Here, we show details of the variational simulation [34, 263], including initial state preparation, the parameter update for gates needed in the QAOA simulation, and model compression.

In preparing the initial state of a uniform superposition state, we can assign all the parameters to be zeros, which ensures that every bitstring has an equal amplitude output from the RBM model.

In simulating the problem Hamiltonian layer, we need to simulate multiple two-qubit RZZ gates. For example, for simulating a RZZ gate acting on qubits k and l, we need to add and additional hidden unit (labeled by c) to the RBM. The additional parameters

are also associated with qubits k and l. The nonlinear equation (6.4) becomes

$$e^{a'_k x_k + a'_l x_l}(1 + e^{W_{kc}x_k + W_{lc}x_l})\psi_\theta(\mathbf{x}) = C\,\langle \mathbf{x}|G|\psi_\theta\rangle\,. \tag{A.6}$$

To implement the RZZ gate: $\text{RZZ}(\gamma) = e^{-i\frac{\varphi}{2}\text{Z}\otimes\text{Z}}$ on qubit k and l, the solution to (A.6) reads:

$$\begin{aligned} W_{ic} &= -2\mathcal{A}(\gamma), \\ W_{jc} &= 2\mathcal{A}(\gamma), \\ a'_i &= a_i + \mathcal{A}(\gamma), \\ a'_j &= a_j - \mathcal{A}(\gamma), \\ C &= 2 \end{aligned} \tag{A.7}$$

where $\mathcal{A}(\gamma) = \text{Arccosh}(e^{i\gamma})$. By repeating the above update rule, given a current RBM model, the Hamiltonian layer can always be exactly simulated by inducing only a few additional parameters. In detail, we add one hidden layer to the RBM for each RZZ gate.

However, in order to simulate the RX gate in simulating the mixer layer, we have to approximate the gate operator by updating the RBM parameters via a learning procedure. We use a stochastic reconfiguration method as the learning algorithm, which will be detailed in Appendix A.3.2. In the implementation, we may also need to simulate the RX gate to evaluate the sample probability in MCMC. For example, for simulating a Pauli-X gate acting on qubit i, we want to solve

$$\begin{aligned} \ln C + a'_i x_i &= (1 - x_i)a_i, \\ b'_k + x_i W'_{ik} &= b_k + (1 - x_i)W_{ik} \end{aligned} \tag{A.8}$$

for $x_i = 0, 1$. The solution is

$$\begin{aligned} \ln C = a_i, a_i' = -a_i, \\ b_k' = b_k + W_{ik}, W_{ik}' = -W_{ik} \end{aligned} \tag{A.9}$$

with all other parameters unchanged.

After simulating one layer of problem Hamiltonian and mixer, to keep the RBM model compact with a reasonable number of hidden layers, we apply a compression step towards the RBM model. Specifically, we initialize a smaller RBM to represent a quantum state and update its parameters by stochastic optimization, aiming to approximate the state represented by the larger RBM. In our experiments, we compress the RBM model to its half size. For simulating QAOA circuits with near-optimal parameters, the variational method has shown its superiority in the model size and memory over a tensor network model. See more details in Ref. [34].

A.3.2 Stochastic reconfiguration

Stochastic reconfiguration is a gradient-descend-like algorithm that uses the information encoded in the quantum geometric tensor to precondition the gradient used in stochastic optimization, which could be derived from the quantum imaginary time evolution [264, 265].

Given a current state $|\boldsymbol{\psi_\theta}\rangle$ and a target state $|\boldsymbol{\phi}\rangle$, our goal is to minimize their distance:

$$D = 1 - F\left(\boldsymbol{\psi_\theta}, \boldsymbol{\phi}\right) = 1 - \frac{\left|\langle \boldsymbol{\phi}|\boldsymbol{\psi_\theta}\rangle\right|^2}{\langle \boldsymbol{\phi}|\boldsymbol{\phi}\rangle \langle \boldsymbol{\psi_\theta}|\boldsymbol{\psi_\theta}\rangle} \tag{A.10}$$

Let $\mathbf{O}_k$ be a diagonal operator in the computational basis such as $\langle \mathbf{x}'|\mathbf{O}_k|\mathbf{x}\rangle = \frac{\partial \ln \boldsymbol{\psi_\theta}}{\partial \theta_k}$. Define the expectation of an operator over $\boldsymbol{\psi}$ as $\langle \cdot \rangle_{\boldsymbol{\psi}} = \frac{\langle \boldsymbol{\psi}|\cdot|\boldsymbol{\psi}\rangle}{\langle \boldsymbol{\psi}|\boldsymbol{\psi}\rangle}$. To minimize (A.10) via

stochastic reconfiguration, the parameters are updated as follows:

$$\theta_k^{t+1} = \theta_k^t - \eta \sum_l S_{kl}^{-1} \frac{\partial D}{\partial \boldsymbol{\theta}^\star}, \tag{A.11}$$

where η is the learning rate and $\mathbf{S}$ is the Quantum Geometric Tensor or Quantum Fisher Information [265, 266] with

$$S_{kl} = \langle \mathbf{O}_k^\dagger \mathbf{O}_l \rangle_{\boldsymbol{\psi_\theta}} - \langle \mathbf{O}_k^\dagger \rangle_{\boldsymbol{\psi_\theta}} \langle \mathbf{O}_l \rangle_{\boldsymbol{\psi_\theta}} . \tag{A.12}$$

The estimated gradient is written as

$$\frac{\partial D}{\partial \boldsymbol{\theta}^\star} = \left\langle \frac{\boldsymbol{\phi}}{\boldsymbol{\psi_\theta}} \right\rangle_{\boldsymbol{\psi_\theta}} \left\langle \frac{\boldsymbol{\psi_\theta}}{\boldsymbol{\phi}} \right\rangle_{\boldsymbol{\phi}} \left[\langle \mathcal{O}_k^\star \rangle_{\boldsymbol{\psi_\theta}} - \frac{\langle \frac{\boldsymbol{\phi}}{\boldsymbol{\psi_\theta}} \mathcal{O}_k^\star \rangle_{\boldsymbol{\psi_\theta}}}{\langle \frac{\boldsymbol{\phi}}{\boldsymbol{\psi_\theta}} \rangle_{\boldsymbol{\psi_\theta}}} \right] . \tag{A.13}$$

A.3.3 Additional experiments

Under optimal parameter. In Fig. A.1, we report the evolution of QAOA circuits of various depths. Under the well-optimized parameter schedule, as the number of layers increases, the entanglement entropy initially rises, then stabilizes and eventually decreases. For QAOA circuits with different values of p, both the highest entanglement entropy and the lowest fidelity are observed in the $p = 4$ circuit. Interestingly, our simulation results indicate that a QAOA circuit with a higher p is not necessarily more challenging to simulate, which may appear counterintuitive.

Under grid-search parameter. In Fig. A.2, we perform a grid search on the one-dimensional parameter γ across different layers of a $p = 8$ QAOA circuit. We observe that both the entanglement entropy of the circuit and the instantaneous fidelity are more sensitive to γ in the early layers. This suggests that the variational simulation method necessitates a highly optimized QAOA parameter schedule, particularly in the initial

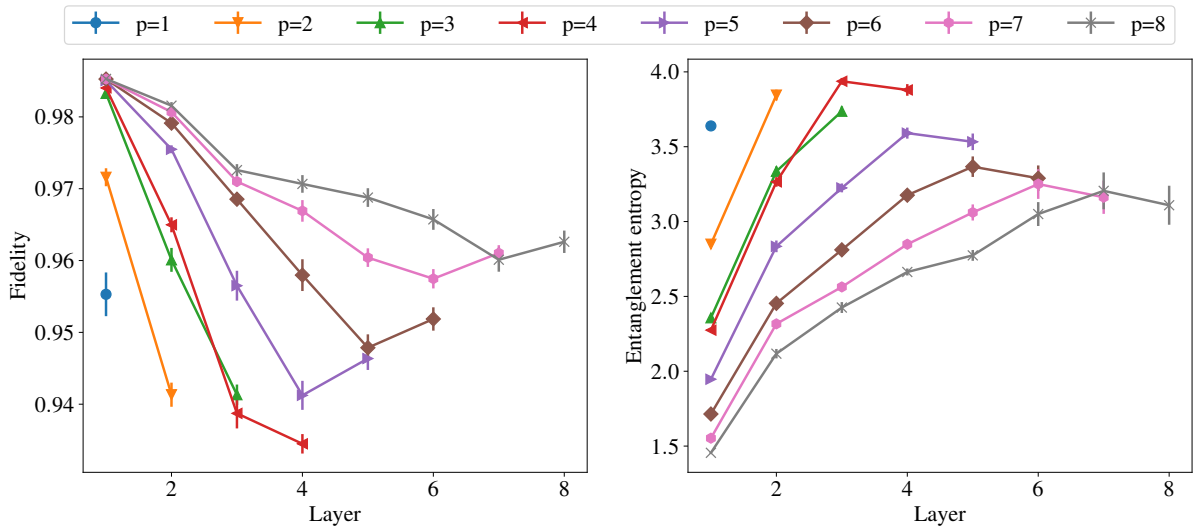

Figure A.1: The performance of approximated simulations for QAOA circuits with varying depths $p = 1, 2, \ldots, 8$ on $N = 16$ 3-regular graph MaxCut problems with well-optimized QAOA parameters. As the depth p increases, we observe a decrease in the approximation fidelity while the EE of the state increases initially, stabilizes, and then eventually decreases. Notably, the highest entanglement entropy and lowest fidelity are both observed at $p = 4$. For $p = 5, 6, 7, 8$, we notice a slight increase in fidelity at the last layer. The increase in fidelity is not guaranteed, and we interpret it as a sign of robust approximate performance in the last layer.

layers. For a circuit with a given parameter configuration, the relationship between entanglement entropy and instantaneous fidelity follows a consistent trend that shows higher entanglement entropy correlates with lower fidelity. We conduct the same experiment for circuits with $p = 4$ and $p = 6$, yielding similar results as depicted in Fig.A.3. Each point in the scatter plot corresponds to a circuit with a specific parameter schedule in a certain layer. The range of obtained entanglement entropy remains consistent across QAOA circuits with varying depths.

Sequence order of simulating rx gates. Given that the approximation of a series of RX gates is the origin of simulation error, we conducted experiments to investigate the impact of the sequence in which the RX gates are simulated. In line 8 of Alg. 5, the default simulation order is based on the node index in the graph. We randomly permutate

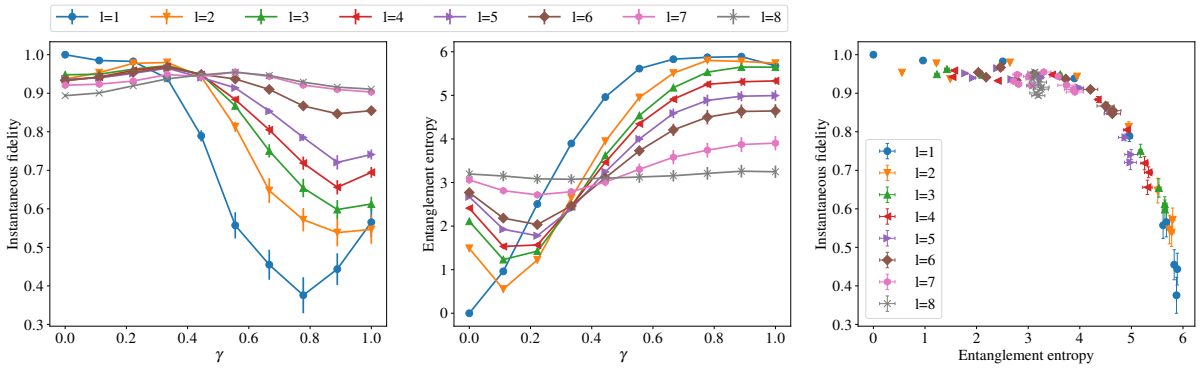

Figure A.2: The performance of approximated simulations for $p = 8$ QAOA circuits on $N = 16$ 3-regular graph MaxCut problems at the l-th layer. All QAOA parameters are well optimized, except for the l-th γ_l, which is determined through a grid search. When the value of l is relatively small, we observe that the QAOA circuit at the l-th layer becomes more sensitive to the specific value of γ_l in terms of both fidelity and entanglement entropy. By examining the scatter plot of entanglement entropy versus fidelity, we find that the observed trend aligns with the one observed in the simulations with optimized parameters. Specifically, circuits with higher entanglement entropy are more challenging to simulate accurately.

the list of update indexes for the variational simulation, which does not influence the exact simulation results since all single qubit RX gates commute with each other. The simulation results for the same $p = 4$ QAOA circuit with various RX update orders are presented in Fig. A.4. For $l = 1, 2, 3$, the QAOA parameters are well-optimized, resulting in similar fidelity across different update orders. However, in the last layer ($l = 4$), where we assigned an unoptimized value to γ_4, the simulation fidelities experienced a significant drop. Although different sequence orders for simulating RX gates resulted in varying degrees of fidelity, none of the sequences could substantially mitigate the drop in simulation fidelity. In our experiments, we did not observe a clear pattern among the sequences that could offer slight improvements to simulation fidelity.

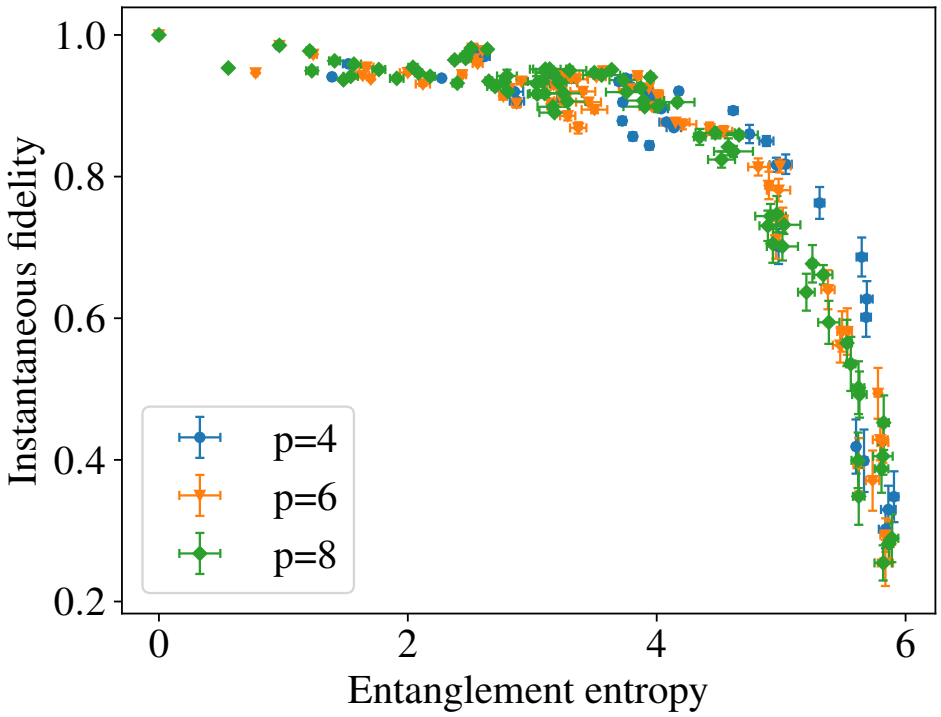

Figure A.3: We repeat the above experiments on QAOA circuits with depths with $p = 6$ and $p = 4$. By performing a one-dimensional grid search, we found that the observed relationship between entropy and fidelity persists, regardless of the specific depth of the QAOA circuit.

A.4 Chapter 7

A.4.1 Detailed experiments and more results

QAOA

In the QAOA algorithm, we characterized the noise level PDF as follows. We first generate a truncated Gaussian distribution with mean -0.01 and standard deviation 0.01. We estimate the probability density at each discretized level and do the normalization to obtain the reference PDF of the noise level. We follow a similar procedure to generate the PDF of shifted noise by shifting the mean of the initial truncated Gaussian distribution.

One example of the newly sampled $\boldsymbol{\theta}$ with $p = 1$ is plotted in Fig. A.5. The DRBO

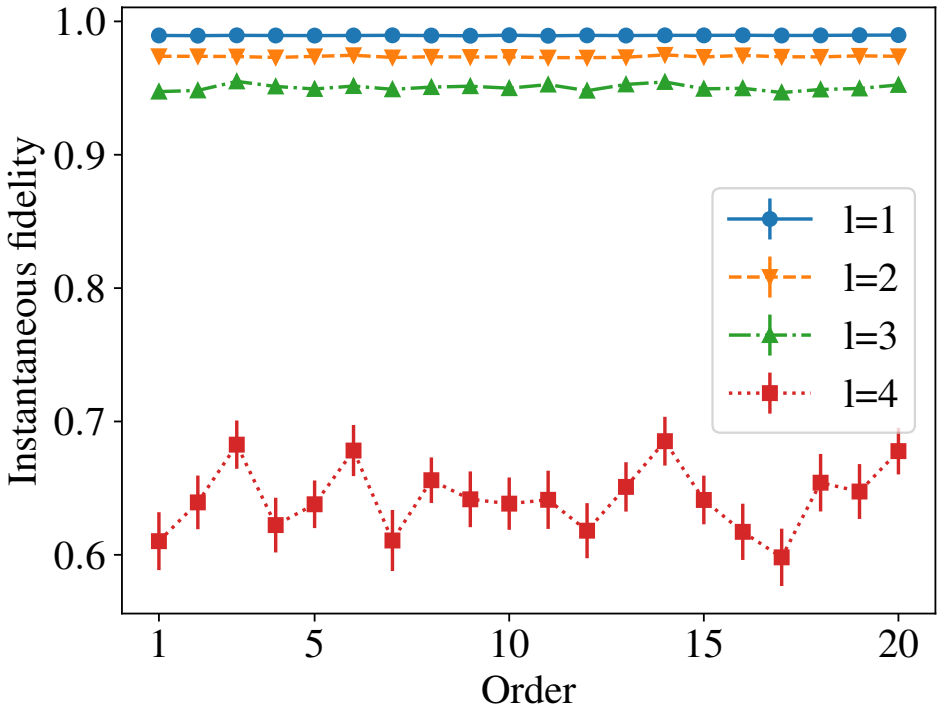

Figure A.4: The impact of the order in which the RX gate is approximated in QAOA circuits. In our experiment, we considered 20 different update orders obtained by permuting the list of update indexes. The approximations were performed on a $N = 10$ MaxCut problem using a $p = 4$ circuit with optimized parameters, except for the last γ which was set to 0.7. Our results indicate that although the approximated fidelity slightly varies under different permutation orders, no specific order can significantly mitigate the drop in simulation fidelity.

algorithm explores the parameter space toward the optimal one under a shifted noise, while the other algorithms exploit the space surrounding the optimal parameter under the reference noise. Therefore, the DRBO could find a parameter that performs better under shifted noise.

More results on MaxCut with graph size $N = 8, 10, 12$ and QAOA depth $p = 1, 2, 3$ are shown in Fig. A.6. The results are consistent with the ones in Fig. 7.2. The DRBO solution performs better than the baselines under significantly shifted noise, which demonstrates that our method could optimize the VQA parameters that are more robust to the real-time shifting noise.

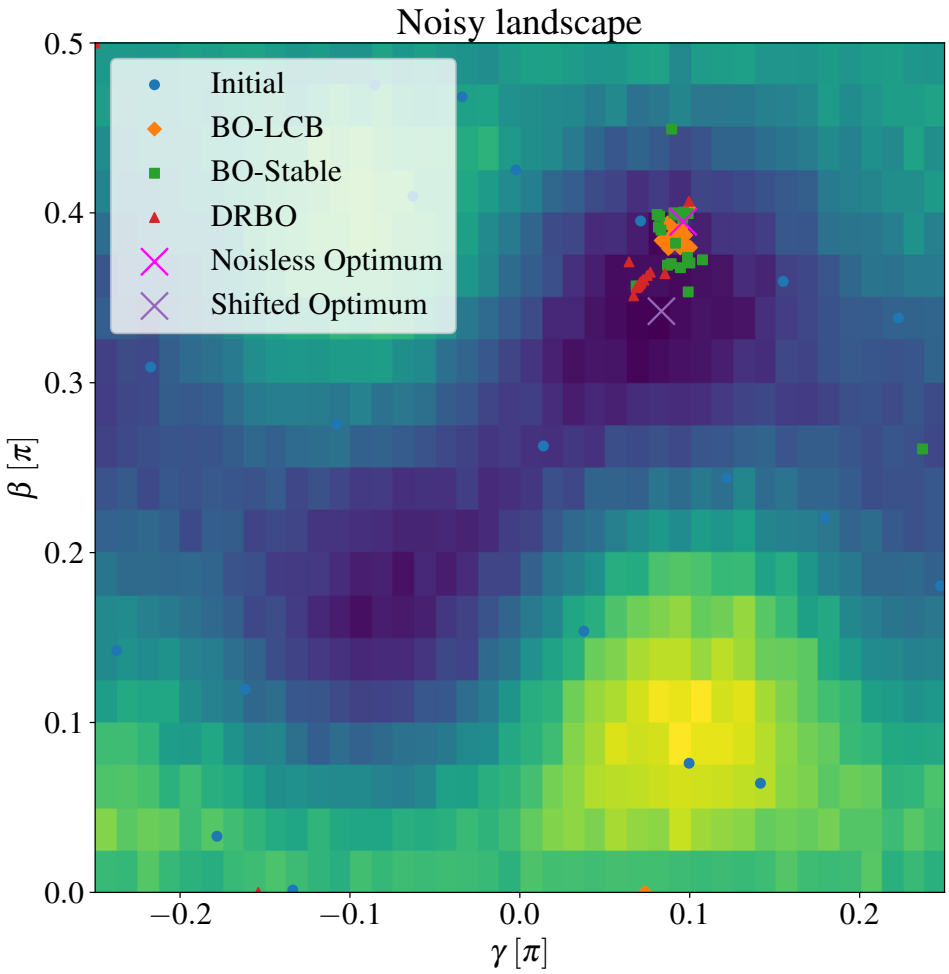

Figure A.5: An example of explored $\boldsymbol{\theta}$ in $p = 1$ noisy QAOA cost landscape in a $N = 8$ MaxCut problem. The optimum $\boldsymbol{\theta}$ differs from the noiseless optimum. Compared to the BO-LCB and BO-Stable, the DRBO explores the parameter space that performs well under the shifted noise.

VQE

The hardware-efficiency ansatz is set up as shown in Fig. A.7. The number of parameters grows quickly and becomes challenging for a BO solver. For simplicity, we only optimize the last N parameters and fix the others, similar to the idea of layer-wise optimization in Ref. [267]. For the demonstration purpose, the fixed parameters are obtained through a multi-start classical optimization routine. We follow the same procedure as QAOA to set up the noise level distribution of both the reference and the shifted ones.

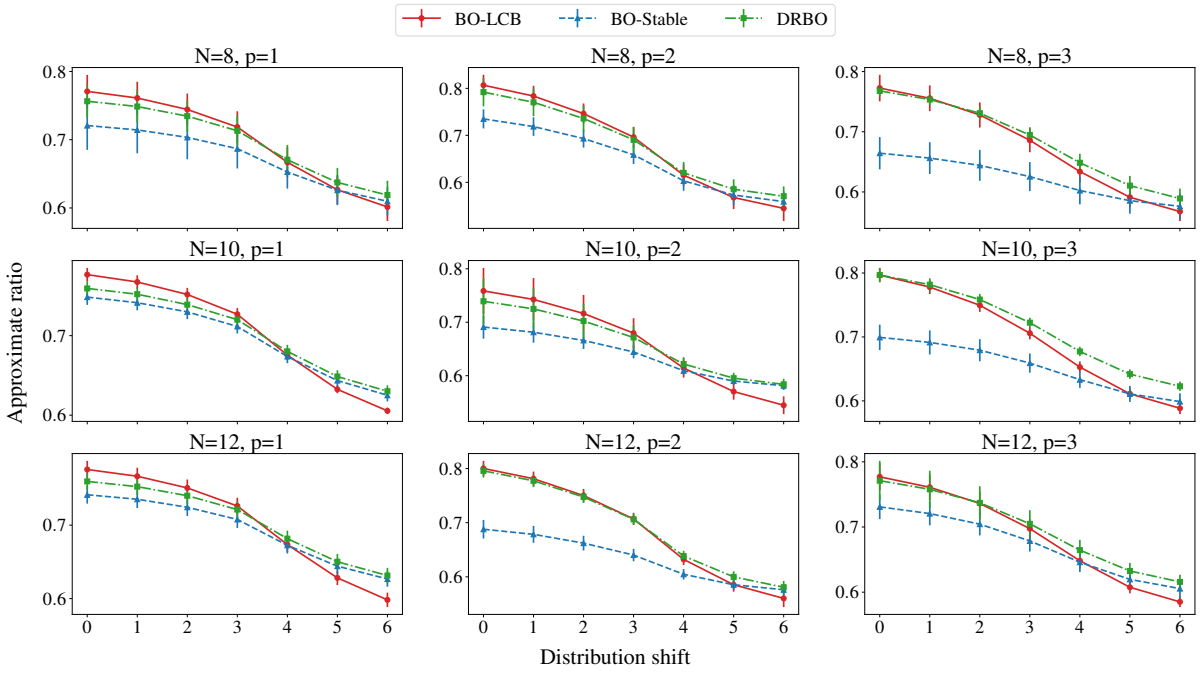

Figure A.6: More results on the Max-Cut experiments with graph sizes $N = 8, 10, 12$ and QAOA depth $p = 1, 2, 3$. While potentially sacrificing the performance under the reference noise a little, the DRBO solution performs better under the significantly shifted noise. Meanwhile, BO-Stable solutions are over-conservative.

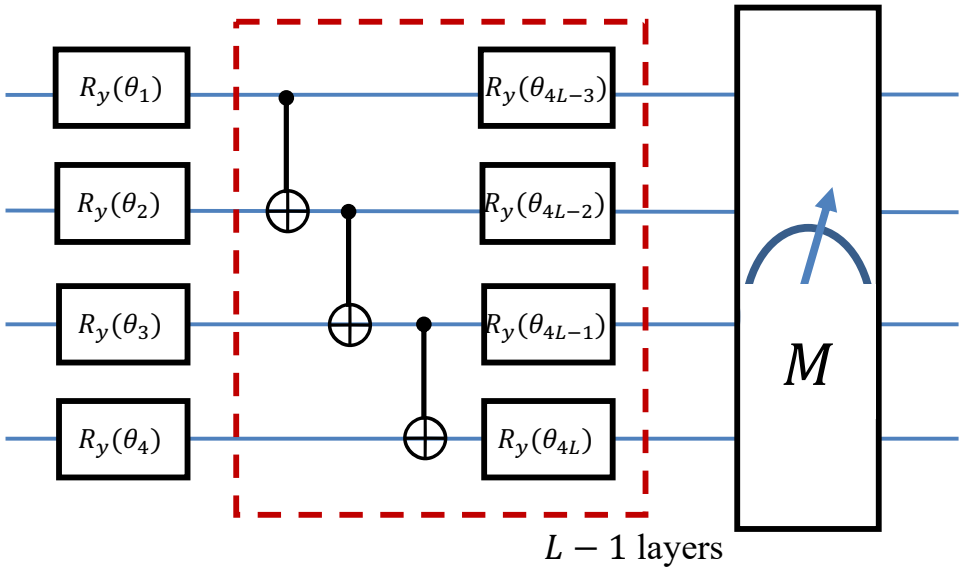

Figure A.7: The schematic of the hardware-efficient ansatz for variational quantum eigensolver. The layer of two-qubit entanglement and one-qubit rotation gates are repeated for $L-1$ times.

Bibliography

[1] C. A. Mack, *Fifty years of Moore's law*, *IEEE Transactions on semiconductor manufacturing* **24** (2011), no. 2 202–207.

[2] K. J. Kuhn, M. D. Giles, D. Becher, P. Kolar, A. Kornfeld, R. Kotlyar, S. T. Ma, A. Maheshwari, and S. Mudanai, *Process technology variation*, *IEEE Transactions on Electron Devices* **58** (2011), no. 8 2197–2208.

[3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, *Parameter variations and impact on circuits and microarchitecture*, in *Proceedings of the 40th annual Design Automation Conference*, pp. 338–342, 2003.

[4] 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.* **21** (2008), no. 1 63–71.

[5] G. Gielen, P. De Wit, E. Maricau, J. Loeckx, J. Martin-Martinez, B. Kaczer, G. Groeseneken, R. Rodriguez, and M. Nafria, *Emerging yield and reliability challenges in nanometer CMOS technologies*, in *Proc. Design, Autom. Test Eur. Conf. Exhibit.*, pp. 1322–1327, 2008.

[6] X. Chen, M. Mohamed, Z. Li, L. Shang, and A. R. Mickelson, *Process variation in silicon photonic devices*, *Appl. Opt.* **52** (2013), no. 31 7638–7647.

[7] L. Lavagno, I. L. Markov, G. Martin, and L. K. Scheffer, *Electronic design automation for IC implementation, circuit design, and process technology: circuit design, and process technology*. CRC Press, 2016.

[8] J. Ogrodzki, *Circuit simulation methods and algorithms*. crc Press, 2018.

[9] D. Pederson, *A historical review of circuit simulation*, *IEEE Transactions on Circuits and Systems* **31** (1984), no. 1 103–111.

[10] S. Duvall, *Statistical circuit modeling and optimization*, in *2000 5th International Workshop on Statistical Metrology (Cat.No.00TH8489*, pp. 56–63, 2000.

[11] J. Shalf, *The future of computing beyond Moore's law*, *Philosophical Transactions of the Royal Society A* **378** (2020), no. 2166 20190061.

[12] L. Gyongyosi and S. Imre, *A survey on quantum computing technology*, *Computer Science Review* **31** (2019) 51–71.

[13] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, *Stochastic p-bits for invertible logic*, *Physical Review X* **7** (2017), no. 3 031014.

[14] N. L. Kazanskiy, M. A. Butt, and S. N. Khonina, *Optical computing: Status and perspectives*, *Nanomaterials* **12** (2022), no. 13 2171.

[15] J. Preskill, *Quantum computing in the NISQ era and beyond*, *Quantum* **2** (2018) 79.

[16] G. De Micheli, J.-H. R. Jiang, R. Rand, K. Smith, and M. Soeken, *Advances in quantum computation and quantum technologies: A design automation perspective*, *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **12** (2022), no. 3 584–601.

[17] A. Zulehner and R. Wille, *Introducing design automation for quantum computing*, vol. 11. Springer, 2020.

[18] N. Quetschlich, L. Burgholzer, and R. Wille, *Mqt bench: Benchmarking software and design automation tools for quantum computing*, *arXiv preprint arXiv:2204.13719* (2022).

[19] I. Polian and A. G. Fowler, *Design automation challenges for scalable quantum architectures*, in *Proceedings of the 52nd annual design automation conference*, pp. 1–6, 2015.

[20] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et. al.*, *Quantum supremacy using a programmable superconducting processor*, *Nature* **574** (2019), no. 7779 505–510.

[21] "Ibm Quantum." `https://quantum-computing.ibm.com/`.

[22] "Quantinuum hardware." https://www.quantinuum.com/hardware.

[23] 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.* **7** (2017), no. 5 687–697.

[24] 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.* **34** (2014), no. 1 63–76.

[25] C. Cui, K. Liu, and Z. Zhang, *Chance-constrained and yield-aware optimization of photonic ICs with non-Gaussian correlated process variations*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* (2020).

[26] B. E. Stine, D. S. Boning, and J. E. Chung, *Analysis and decomposition of spatial variation in integrated circuit processes and devices*, *IEEE Transactions on Semiconductor Manufacturing* **10** (1997), no. 1 24–41.

[27] V. Reddy, J. Carulli, A. Krishnan, W. Bosch, and B. Burgess, *Impact of negative bias temperature instability on product parametric drift*, in *International Conference on Test*, pp. 148–155, 2004.

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

[29] Z. He and Z. Zhang, *High-dimensional uncertainty quantification via tensor regression with rank determination and adaptive sampling*, *IEEE Trans. Compon. Packag. Manuf. Technol.* **11** (2021), no. 9 1317–1328.

[30] Z. He and Z. Zhang, *PoBO: A polynomial bounding method for chance-constrained yield-aware optimization of photonic ics*, *IEEE Trans. CAD Integr. Circuits Syst.* **41** (2022), no. 11 4915–4926.

[31] Y. Pan, Z. He, G. Nanlin, and Z. Zhang, *Distributionally robust circuit design optimization under variation shifts*, in *Proc. Intl. Conf. Computer Aided Design*, pp. 1–8, 2023.

[32] X. Xu, S. Benjamin, J. Sun, X. Yuan, and P. Zhang, *A herculean task: Classical simulation of quantum computers*, *arXiv preprint arXiv:2302.08880* (2023).

[33] G. F. Viamontes, I. L. Markov, and J. P. Hayes, *Quantum circuit simulation*, vol. 410. Springer, 2009.

[34] M. Medvidović and G. Carleo, *Classical variational simulation of the quantum approximate optimization algorithm*, *npj Quantum Information* **7** (2021), no. 1 1–7.

[35] Z. He, Z. Zhang, and S. Mandra, *One the variational method for QAOA simulation*, *In preparation* (2023).

[36] A. A. Clerk, M. H. Devoret, S. M. Girvin, F. Marquardt, and R. J. Schoelkopf, *Introduction to quantum noise, measurement, and amplification*, *Reviews of Modern Physics* **82** (2010), no. 2 1155.

[37] S. Resch and U. R. Karpuzcu, *Benchmarking quantum computers and the impact of quantum noise*, *ACM Computing Surveys (CSUR)* **54** (2021), no. 7 1–35.

[38] Z. He, B. Peng, Y. Alexeev, and Z. Zhang, *Distributionally robust variational quantum algorithms with shifted noise*, *arXiv preprint arXiv:2308.14935* (2023).

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

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

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

[42] 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.* **32** (2013), no. 10 1533–1545.

[43] 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* **61** (2014), no. 7 2057–2065.

[44] 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*, pp. 1–8, 2019.

[45] 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.* **35** (2016), no. 10 1640–1652.

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

[47] 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.* **25** (2015), no. 8 505–507.

[48] 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.* **5** (2015), no. 5 685–696.

[49] 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.* **35** (2015), no. 8 1255–1268.

[50] 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*, pp. 1–6, 2019.

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

[52] 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.* **29** (2010), no. 11 1661–1668.

[53] 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.* **39** (2019), no. 8 1649–1661.

[54] 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.*, pp. 1–4, 2016.

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

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

[57] 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.*, pp. 453–458, 2010.

[58] 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*, pp. 1–8, 2014.

[59] Z. Zhang, T.-W. Weng, and L. Daniel, *A big-data approach to handle process variations: Uncertainty quantification by tensor recovery*, in *Proc. IEEE Workshop Signal Power Integr.*, pp. 1–4, 2016.

[60] P. Rai, *Sparse low rank approximation of multivariate functions–Applications in uncertainty quantification*. PhD thesis, Ecole Centrale de Nantes (ECN), 2014.

[61] 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.* **321** (2016) 1144–1169.

[62] 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.* **3** (2015), no. 1 897–921.

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

[64] Z. He and Z. Zhang, *Progress of tensor-based high-dimensional uncertainty quantification of process variations*, in *2021 International Applied Computational Electromagnetics Society Symposium (ACES)*, pp. 1–3, IEEE, 2021.

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

[66] 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*, pp. 1–6, 2019.

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

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

[69] J. Kossaifi, Z. C. Lipton, A. Kolbeinsson, A. Khanna, T. Furlanello, and A. Anandkumar, *Tensor regression networks*, *J. Mach. Learn. Res.* **21** (2020) 1–21.

[70] W. Guo, I. Kotsia, and I. Patras, *Tensor learning for regression*, *IEEE Trans. Image Process.* **21** (2011), no. 2 816–827.

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

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

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

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

[75] Z. He, B. Zhao, and Z. Zhang, *Active sampling for accelerated mri with low-rank tensors*, in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3024–3028, IEEE, 2022.

[76] R. Guhaniyogi, S. Qamar, and D. B. Dunson, *Bayesian tensor regression*, *J. Mach. Learn. Res.* **18** (2017), no. 1 2733–2763.

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

[78] W. Gautschi, *On generating orthogonal polynomials*, *SIAM J. Sci. Stat. Comp.* **3** (1982), no. 3 289–317.

[79] R. G. Ghanem and P. D. Spanos, *Stochastic finite element method: Response statistics*, in *Stochastic finite elements: a spectral approach*, pp. 101–119. Springer, 1991.

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

[81] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, *SIAM Rev.* **51** (2009), no. 3 455–500.

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

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

[84] 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* **42** (2000), no. 1 55–61.

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

[86] 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.

[87] 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.* **53** (2017), no. 12 10802–10823.

[88] 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.* **181** (2010), no. 2 259–270.

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

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

[91] S. Marelli and B. Sudret, *UQLab: A framework for uncertainty quantification in matlab*, in *Vulnerability, Uncertainty, and Risk*, American Society of Civil Engineers, June, 2014.

[92] T. Lipka, J. Müller, and H. K. Trieu, *Systematic nonuniformity analysis of amorphous silicon-on-insulator photonic microring resonators*, *J. Light. Technol.* **34** (2016), no. 13 3163–3170.

[93] W. Bogaerts, Y. Xing, and U. Khan, *Layout-aware variability analysis, yield prediction, and optimization in photonic integrated circuits*, *IEEE J. Sel. Top. Quantum Electron.* **25** (2019), no. 5 1–13.

[94] K. J. Antreich, H. E. Graeb, and C. U. Wieser, *Circuit analysis and optimization driven by worst-case distances*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **13** (1994), no. 1 57–71.

[95] F. Gong, Y. Shi, H. Yu, and L. He, *Variability-aware parametric yield estimation for analog/mixed-signal circuits: Concepts, algorithms, and challenges*, *IEEE Des. Test* **31** (2014), no. 4 6–15.

[96] Z. Lu, J. Jhoja, J. Klein, X. Wang, A. Liu, J. Flueckiger, J. Pond, and L. Chrostowski, *Performance prediction for silicon photonics integrated circuits with layout-dependent correlated manufacturing variability*, *Opt. Express* **25** (2017), no. 9 9712–9733.

[97] J. Pond, J. Klein, J. Flückiger, X. Wang, Z. Lu, J. Jhoja, and L. Chrostowski, *Predicting the yield of photonic integrated circuits using statistical compact modeling*, in *Integrated Optics: Physics and Simulations III*, vol. 10242, p. 102420S, 2017.

[98] Y. Xu, K.-L. Hsiung, X. Li, L. T. Pileggi, and S. P. Boyd, *Regular analog/RF integrated circuits design using optimization with recourse including ellipsoidal uncertainty*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **28** (2009), no. 5 623–637.

[99] G. Yu and P. Li, *Yield-aware analog integrated circuit optimization using geostatistics motivated performance modeling*, in *Proc. Intl. Conf. Computer Aided Design*, pp. 464–469, 2007.

[100] S. K. Tiwary, P. K. Tiwary, and R. A. Rutenbar, *Generation of yield-aware Pareto surfaces for hierarchical circuit design space exploration*, in *Proc. Design Autom. Conf*, pp. 31–36, 2006.

[101] Y. Li and V. Stojanovic, *Yield-driven iterative robust circuit optimization algorithm*, in *Proc. Design Autom. Conf*, pp. 599–604, 2009.

[102] B. Liu, F. V. Fernández, and G. G. Gielen, *Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **30** (2011), no. 6 793–805.

[103] M. Barros, J. Guilherme, and N. Horta, *Analog circuits optimization based on evolutionary computation techniques*, *Integration* **43** (2010), no. 1 136–155.

[104] M. Wang, F. Yang, C. Yan, X. Zeng, and X. Hu, *Efficient Bayesian yield optimization approach for analog and SRAM circuits*, in *Proc. Design Autom. Conf*, pp. 1–6, 2017.

[105] A. Singhee, S. Singhal, and R. A. Rutenbar, *Practical, fast Monte Carlo statistical static timing analysis: Why and how*, in *Proc. Intl. Conf. Computer Aided Design*, pp. 190–195, 2008.

[106] A. Papoulis and H. Saunders, *Probability, random variables and stochastic processes*. McGraw-Hill, 2001.

[107] C. Gu and J. Roychowdhury, *An efficient, fully nonlinear, variability-aware non-Monte-Carlo yield estimation procedure with applications to SRAM cells and ring oscillators*, in *Proc. Asia South Pac. Design Autom. Conf.*, pp. 754–761, 2008.

[108] F. Gong, X. Liu, H. Yu, S. X. Tan, J. Ren, and L. He, *A fast non-Monte-Carlo yield analysis and optimization by stochastic orthogonal polynomials*, *ACM Trans. Des. Autom. Electron. Syst.* **17** (2012), no. 1 1–23.

[109] Z. Gao and R. Rohrer, *Efficient non-Monte-Carlo yield estimation*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* (2021) 1–1.

[110] J. Yao, Z. Ye, and Y. Wang, *An efficient SRAM yield analysis and optimization method with adaptive online surrogate modeling*, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **23** (2014), no. 7 1245–1253.

[111] X. Li, J. Le, P. Gopalakrishnan, and L. T. Pileggi, *Asymptotic probability extraction for nonnormal performance distributions*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **26** (2006), no. 1 16–37.

[112] X. Li, P. Gopalakrishnan, Y. Xu, and T. Pileggi, *Robust analog/RF circuit design with projection-based posynomial modeling*, in *Proc. Intl. Conf. Computer Aided Design*, pp. 855–862, 2004.

[113] X. Li, Y. Zhan, and L. T. Pileggi, *Quadratic statistical MAX approximation for parametric yield estimation of analog/rf integrated circuits*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **27** (2008), no. 5 831–843.

[114] A. Ciccazzo, G. Di Pillo, and V. Latorre, *A SVM surrogate model-based method for parametric yield optimization*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **35** (2015), no. 7 1224–1228.

[115] H. Ma, E.-P. Li, A. C. Cangellaris, and X. Chen, *Support vector regression-based active subspace (SVR-AS) modeling of high-speed links for fast and accurate sensitivity analysis*, *IEEE Access* **8** (2020) 74339–74348.

[116] A. C. Sanabria-Borbón, S. Soto-Aguilar, J. J. Estrada-López, D. Allaire, and E. Sánchez-Sinencio, *Gaussian-process-based surrogate for optimization-aided and process-variations-aware analog circuit design*, *Electronics* **9** (2020), no. 4 685.

[117] M. Wang, W. Lv, F. Yang, C. Yan, W. Cai, D. Zhou, and X. Zeng, *Efficient yield optimization for analog and SRAM circuits via Gaussian process regression and adaptive yield estimation*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **37** (2017), no. 10 1929–1942.

[118] Y. Wang, M. Orshansky, and C. Caramanis, *Enabling efficient analog synthesis by coupling sparse regression and polynomial optimization*, in *Proc. Design Autom. Conf*, pp. 1–6, 2014.

[119] R. Trinchero and F. G. Canavero, *Combining LS-SVM and GP regression for the uncertainty quantification of the EMI of power converters affected by several uncertain parameters*, *IEEE Trans. Electromagn. Compat.* **62** (2020), no. 5 1755–1762.

[120] A. Kaintura, T. Dhaene, and D. Spina, *Review of polynomial chaos-based methods for uncertainty quantification in modern integrated circuits*, *Electronics* **7** (2018), no. 3 30.

[121] F. Wang, S. Yin, M. Jun, X. Li, T. Mukherjee, R. Negi, and L. Pileggi, *Re-thinking polynomial optimization: efficient programming of reconfigurable radio frequency (RF) systems by convexification*, in *Proc. Asia South Pac. Design Autom. Conf.*, pp. 545–550, 2016.

[122] J. Tao, Y. Su, D. Zhou, X. Zeng, and X. Li, *Graph-constrained sparse performance modeling for analog circuit optimization via SDP relaxation*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **38** (2018), no. 8 1385–1398.

[123] A. Waqas, P. Manfredi, and D. Melati, *Performance variability analysis of photonic circuits with many correlated parameters*, *J. Light. Technol.* (2021).

[124] A. Waqas, D. Melati, P. Manfredi, and A. Melloni, *Stochastic process design kits for photonic circuits based on polynomial chaos augmented macro-modelling*, *Opt. Express* **26** (2018), no. 5 5894–5907.

[125] T.-W. Weng, Z. Zhang, Z. Su, Y. Marzouk, A. Melloni, and L. Daniel, *Uncertainty quantification of silicon photonic devices with correlated and non-Gaussian random parameters*, *Opt. Express* **23** (2015), no. 4 4242–4254.

[126] T.-W. Weng, D. Melati, A. Melloni, and L. Daniel, *Stochastic simulation and robust design optimization of integrated photonic filters*, *Nanophotonics* **6** (2017), no. 1 299–308.

[127] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

[128] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz, *Stochastic nonlinear model predictive control with probabilistic constraints*, in *Proc. Am. Control Conf.*, pp. 2413–2419, 2014.

[129] M. P. Vitus, Z. Zhou, and C. J. Tomlin, *Stochastic control with uncertain parameters via chance constrained control*, *IEEE Trans. Autom. Control* **61** (2015), no. 10 2892–2905.

[130] Z. Wang, C. Shen, F. Liu, X. Wu, C.-C. Liu, and F. Gao, *Chance-constrained economic dispatch with non-Gaussian correlated wind power uncertainty*, *IEEE Trans. Power Syst.* **32** (2017), no. 6 4880–4893.

[131] B. P. Van Parys, P. J. Goulart, and D. Kuhn, *Generalized Gauss inequalities via semidefinite programming*, *Math. Program.* **156** (2016), no. 1-2 271–302.

[132] C. Feng, F. Dabbene, and C. M. Lagoa, *A kinship function approach to robust and probabilistic optimization under polynomial uncertainty*, *IEEE Trans. Autom. Control* **56** (2010), no. 7 1509–1523.

[133] G. C. Calafiore and L. El Ghaoui, *On distributionally robust chance-constrained linear programs*, *J. Optim. Theory Appl.* **130** (2006), no. 1 1–22.

[134] D. Henrion, J.-B. Lasserre, and J. Löfberg, *Gloptipoly 3: moments, optimization and semidefinite programming*, *Optim. Methods Softw.* **24** (2009), no. 4-5 761–779.

[135] G. H. Golub and J. H. Welsch, *Calculation of Gauss quadrature rules*, *Math. Comp.* **23** (1969), no. 106 221–230.

[136] T. Gerstner and M. Griebel, *Numerical integration using sparse grids*, *Numer. Algorithms* **18** (1998), no. 3 209–232.

[137] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, *SIAM J. Optim.* **11** (2001), no. 3 796–817.

[138] J. B. Lasserre, *A semidefinite programming approach to the generalized problem of moments*, *Math. Program.* **112** (2008), no. 1 65–92.

[139] J. Nie, *Optimality conditions and finite convergence of lasserre's hierarchy*, *Math. Program.* **146** (2014), no. 1 97–121.

[140] A. Singhee and R. A. Rutenbar, *Why quasi-Monte Carlo is better than Monte Carlo or Latin hypercube sampling for statistical circuit analysis*, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **29** (2010), no. 11 1763–1776.

[141] I. S. Stievano, P. Manfredi, and F. G. Canavero, *Parameters variability effects on multiconductor interconnects via Hermite polynomial chaos*, *IEEE Trans. CPMT* **1** (2011), no. 8 1234–1239.

[142] C. McAndrew, *Statistical modeling for circuit simulation*, in *Int. Symp. Quality Electronic Design*, pp. 357–362, 2003.

[143] I. Stevanovic and C. C. McAndrew, *Quadratic backward propagation of variance for nonlinear statistical circuit modeling*, *IEEE Trans. CAD of Integrated Circuits and Systems* **28** (2009), no. 9 1428–1432.

[144] X. Li, J. Le, and L. T. Pileggi, *Statistical performance modeling and optimization*, *Found. Trends Electron. Des. Autom.* **1** (apr, 2006) 331–480.

[145] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical analysis and optimization for VLSI: Timing and power*, vol. 59. Springer, 2005.

[146] S. Zhang, F. Yang, D. Zhou, and X. Zeng, *Bayesian methods for the yield optimization of analog and SRAM circuits*, in *Asia and South Pacific Design Automation Conf.*, pp. 440–445, 2020.

[147] Z. Gao, J. Tao, F. Yang, Y. Su, D. Zhou, and X. Zeng, *Efficient performance trade-off modeling for analog circuit based on Bayesian neural network*, in *Intl. Conf. Computer-Aided Design*, pp. 1–8, 2019.

[148] A. Dharchoudhury and S.-M. Kang, *Worst-case analysis and optimization of VLSI circuit performances*, *IEEE Trans. CAD Integr. Circuits Syst.* **14** (1995), no. 4 481–492.

[149] J. Kirschner, I. Bogunovic, S. Jegelka, and A. Krause, *Distributionally robust Bayesian optimization*, in *International Conference on Artificial Intelligence and Statistics*, pp. 2174–2184, PMLR, 2020.

[150] T. Nguyen, S. Gupta, H. Ha, S. Rana, and S. Venkatesh, *Distributionally robust Bayesian quadrature optimization*, in *International Conference on Artificial Intelligence and Statistics*, pp. 1921–1931, PMLR, 2020.

[151] H. Husain, V. Nguyen, and A. v. d. Hengel, *Distributionally robust Bayesian optimization with ϕ-divergences*, *arXiv preprint arXiv:2203.02128* (2022).

[152] A. Lange, C. Sohrmann, R. Jancke, J. Haase, B. Cheng, A. Asenov, and U. Schlichtmann, *Multivariate modeling of variability supporting non-Gaussian and correlated parameters*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **35** (2016), no. 2 197–210.

[153] L. Yu, L. Wei, D. Antoniadis, I. Elfadel, and D. Boning, *Statistical modeling with the virtual source mosfet model*, in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1454–1457, IEEE, 2013.

[154] B. Raj, A. Saxena, and S. Dasgupta, *Nanoscale FinFET based SRAM cell design: Analysis of performance metric, process variation, underlapped FinFET, and temperature effect*, *IEEE Circuits and Systems Magazine* **11** (2011), no. 3 38–50.

[155] F. Roger, W. Reinprecht, and R. Minixhofer, *Process variation aware ESD design window considerations on a 0.18 μm analog, mixed-signal high voltage technology*, in *Proc. EOS/ESD Symp.*, pp. 1–7, 2011.

[156] S. Han, J. Choung, B.-S. Kim, B. H. Lee, H. Choi, and J. Kim, *Statistical aging analysis with process variation consideration*, in *International Conference on Computer-Aided Design*, pp. 412–419, 2011.

[157] W. Wang, Z. Wei, S. Yang, and Y. Cao, *An efficient method to identify critical gates under circuit aging*, in *2007 IEEE/ACM International Conference on Computer-Aided Design*, pp. 735–740, IEEE, 2007.

[158] H. Rahimian and S. Mehrotra, *Distributionally robust optimization: A review*, *arXiv preprint arXiv:1908.05659* (2019).

[159] S. M. Ali and S. D. Silvey, *A general class of coefficients of divergence of one distribution from another*, *Journal of the Royal Statistical Society: Series B (Methodological)* **28** (1966), no. 1 131–142.

[160] C. E. Rasmussen, *Gaussian processes in machine learning*, in *Summer school on machine learning*, pp. 63–71, Springer, 2003.

[161] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2 (3). MIT press Cambridge, MA, 2006.

[162] S. Cakmak, R. Astudillo Marban, P. Frazier, and E. Zhou, *Bayesian optimization of risk measures*, *Advances in Neural Information Processing Systems* **33** (2020) 20130–20141.

[163] P. I. Frazier, *A tutorial on Bayesian optimization*, *arXiv preprint arXiv:1807.02811* (2018).

[164] P. Virtanen *et. al.*, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, *Nature Methods* **17** (2020) 261–272.

[165] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, *Gaussian process optimization in the bandit setting: No regret and experimental design*, in *International Conference on Machine Learning*, (Madison, WI, USA), p. 1015–1022, Omnipress, 2010.

[166] Z. Gao and R. Rohrer, *Efficient non-monte-carlo yield estimation*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **41** (2022), no. 5 1222–1235.

[167] Y. Alexeev, D. Bacon, K. R. Brown, R. Calderbank, L. D. Carr, F. T. Chong, B. DeMarco, D. Englund, E. Farhi, B. Fefferman, *et. al.*, *Quantum computer systems for scientific discovery*, *PRX Quantum* **2** (2021), no. 1 017001.

[168] S. Mandrà, J. Marshall, E. G. Rieffel, and R. Biswas, *Hybridq: A hybrid simulator for quantum circuits*, in *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)*, pp. 99–109, IEEE, 2021.

[169] A. Li, B. Fang, C. Granade, G. Prawiroatmodjo, B. Heim, M. Roetteler, and S. Krishnamoorthy, *SV-Sim: Scalable pgas-based state vector simulation of quantum circuits*, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–14, 2021.

[170] B. Villalonga, D. Lyakh, S. Boixo, H. Neven, T. S. Humble, R. Biswas, E. G. Rieffel, A. Ho, and S. Mandrà, *Establishing the quantum supremacy frontier with a 281 pflop/s simulation*, *Quantum Science and Technology* **5** (2020), no. 3 034003.

[171] D. Lykov, R. Schutski, A. Galda, V. Vinokur, and Y. Alexeev, *Tensor network quantum simulator with step-dependent parallelization*, in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 582–593, IEEE, 2022.

[172] M. S. Rudolph, J. Miller, J. Chen, A. Acharya, and A. Perdomo-Ortiz, *Synergy between quantum circuits and tensor networks: Short-cutting the race to practical quantum advantage*, *arXiv preprint arXiv:2208.13673* (2022).

[173] F. Pan and P. Zhang, *Simulation of quantum circuits using the big-batch tensor network method*, *Physical Review Letters* **128** (2022), no. 3 030501.

[174] C. Huang, F. Zhang, M. Newman, X. Ni, D. Ding, J. Cai, X. Gao, T. Wang, F. Wu, G. Zhang, *et. al.*, *Efficient parallelization of tensor network contraction for simulating quantum computation*, *Nature Computational Science* **1** (2021), no. 9 578–587.

[175] R. Wille, S. Hillmich, and L. Burgholzer, *Tools for quantum computing based on decision diagrams*, *ACM Transactions on Quantum Computing* **3** (2022), no. 3 1–17.

[176] S. Bravyi and D. Gosset, *Improved classical simulation of quantum circuits dominated by Clifford gates*, *Physical review letters* **116** (2016), no. 25 250501.

[177] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, *Simulation of quantum circuits by low-rank stabilizer decompositions*, *Quantum* **3** (2019) 181.

[178] E. Farhi, J. Goldstone, and S. Gutmann, *A quantum approximate optimization algorithm*, *arXiv preprint arXiv:1411.4028* (2014).

[179] C. Ibrahim, D. Lykov, Z. He, Y. Alexeev, and I. Safro, *Constructing optimal contraction trees for tensor network quantum circuit simulation*, in *2022 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–8, IEEE, 2022.

[180] S. Hadfield, Z. Wang, B. O'gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, *From the quantum approximate optimization algorithm to a quantum alternating operator ansatz*, *Algorithms* **12** (2019), no. 2 34.

[181] K. Blekos, D. Brand, A. Ceschini, C.-H. Chou, R.-H. Li, K. Pandya, and A. Summer, *A review on quantum approximate optimization algorithm and its variants*, *arXiv preprint arXiv:2306.09198* (2023).

[182] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, *Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices*, *Physical Review X* **10** (2020), no. 2 021067.

[183] Z. He, R. Shaydulin, S. Chakrabarti, D. Herman, C. Li, Y. Sun, and M. Pistoia, *Alignment between initial state and mixer improves QAOA performance for constrained portfolio optimization*, *arXiv preprint arXiv:2305.03857* (2023).

[184] L. T. Brady, L. Kocia, P. Bienias, A. Bapat, Y. Kharkov, and A. V. Gorshkov, *Behavior of analog quantum algorithms*, *arXiv preprint arXiv:2107.01218* (2021).

[185] J. Basso, E. Farhi, K. Marwaha, B. Villalonga, and L. Zhou, *The quantum approximate optimization algorithm at high depth for MaxCut on large-girth regular graphs and the Sherrington-Kirkpatrick model*, *arXiv preprint arXiv:2110.14206* (2021).

[186] R. Shaydulin, P. C. Lotshaw, J. Larson, J. Ostrowski, and T. S. Humble, *Parameter transfer for quantum approximate optimization of weighted maxcut*, *ACM Transactions on Quantum Computing* **4** (2023), no. 3 1–15.

[187] S. H. Sureshbabu, D. Herman, R. Shaydulin, J. Basso, S. Chakrabarti, Y. Sun, and M. Pistoia, *Parameter setting in quantum approximate optimization of weighted problems*, *arXiv preprint arXiv:2305.15201* (2023).

[188] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo, *et. al.*, *Quantum approximate optimization of non-planar graph problems on a planar superconducting processor*, *Nature Physics* **17** (2021), no. 3 332–336.

[189] Z. Liang, Z. Song, J. Cheng, Z. He, J. Liu, H. Wang, R. Qin, Y. Wang, S. Han, X. Qian, *et. al.*, *Hybrid gate-pulse model for variational quantum algorithms*, *arXiv preprint arXiv:2212.00661* (2022).

[190] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, *Benchmarking the quantum approximate optimization algorithm*, *Quantum Information Processing* **19** (2020) 1–24.

[191] S. Bravyi, D. Gosset, and R. Movassagh, *Classical algorithms for quantum mean values*, *Nature Physics* **17** (2021), no. 3 337–341.

[192] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, *Deep autoregressive models for the efficient variational simulation of many-body quantum systems*, *Physical review letters* **124** (2020), no. 2 020503.

[193] G. Carleo and M. Troyer, *Solving the quantum many-body problem with artificial neural networks*, *Science* **355** (2017), no. 6325 602–606.

[194] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, *Machine learning and the physical sciences*, *Reviews of Modern Physics* **91** (2019), no. 4 045002.

[195] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, *Restricted Boltzmann machines in quantum physics*, *Nature Physics* **15** (2019), no. 9 887–892.

[196] L. Huang and L. Wang, *Accelerated Monte Carlo simulations with restricted Boltzmann machines*, *Physical Review B* **95** (2017), no. 3 035105.

[197] Y. Huang, J. E. Moore, *et. al.*, *Neural network representation of tensor network and chiral states*, *Physical review letters* **127** (2021), no. 17 170601.

[198] O. Sharir, A. Shashua, and G. Carleo, *Neural tensor contractions and the expressive power of deep neural quantum states*, *Physical Review B* **106** (2022), no. 20 205136.

[199] D.-L. Deng, X. Li, and S. D. Sarma, *Quantum entanglement in neural network states*, *Physical Review X* **7** (2017), no. 2 021021.

[200] Z.-A. Jia, L. Wei, Y.-C. Wu, G.-C. Guo, and G.-P. Guo, *Entanglement area law for shallow and deep quantum neural network states*, *New Journal of Physics* **22** (2020), no. 5 053022.

[201] A. Borin and D. A. Abanin, *Approximating power of machine-learning ansatz for quantum many-body states*, *Physical Review B* **101** (2020), no. 19 195141.

[202] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, *Equivalence of restricted Boltzmann machines and tensor network states*, *Physical Review B* **97** (2018), no. 8 085104.

[203] J. Eisert, *Entanglement and tensor network states*, *arXiv preprint arXiv:1308.3318* (2013).

[204] M. Liu, C. Oh, J. Liu, L. Jiang, and Y. Alexeev, *Complexity of Gaussian boson sampling with tensor networks*, *arXiv preprint arXiv:2301.12814* (2023).

[205] D. Stilck França and R. Garcia-Patron, *Limitations of optimization algorithms on noisy quantum devices*, *Nature Physics* **17** (2021), no. 11 1221–1227.

[206] M. Ben-Or, D. Gottesman, and A. Hassidim, *Quantum refrigerator*, *arXiv preprint arXiv:1301.1995* (2013).

[207] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, *The journal of chemical physics* **21** (1953), no. 6 1087–1092.

[208] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, *Biometrika* **57** (1970), no. 1 97–109.

[209] S. Sorella, *Green function Monte Carlo with stochastic reconfiguration*, *Physical review letters* **80** (1998), no. 20 4558.

[210] J. Wurtz and D. Lykov, *Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs*, *Physical Review A* **104** (2021), no. 5 052419.

[211] K. Binder, D. M. Ceperley, J.-P. Hansen, M. Kalos, D. Landau, D. Levesque, H. Mueller-Krumbhaar, D. Stauffer, and J.-J. Weis, *Monte Carlo methods in statistical physics*, vol. 7. Springer Science & Business Media, 2012.

[212] P. Díez-Valle, D. Porras, and J. J. García-Ripoll, *Quantum approximate optimization algorithm pseudo-Boltzmann states*, *Physical review letters* **130** (2023), no. 5 050601.

[213] E. Farhi and A. W. Harrow, *Quantum supremacy through the quantum approximate optimization algorithm*, *arXiv preprint arXiv:1602.07674* (2016).

[214] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et. al.*, *Variational quantum algorithms*, *Nature Reviews Physics* **3** (2021), no. 9 625–644.

[215] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. Müller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme, *Quantum optimization using variational algorithms on near-term quantum devices*, *Quantum Science and Technology* **3** (jun, 2018) 030503.

[216] D. J. Egger, J. Mareček, and S. Woerner, *Warm-starting quantum optimization*, *Quantum* **5** (2021) 479.

[217] R. Orús, S. Mugel, and E. Lizaso, *Quantum computing for finance: Overview and prospects*, *Reviews in Physics* **4** (2019) 100028.

[218] D. Herman, C. Googin, X. Liu, A. Galda, I. Safro, Y. Sun, M. Pistoia, and Y. Alexeev, *A survey of quantum computing for finance*, *arXiv preprint arXiv:2201.02773* (2022).

[219] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, *Quantum machine learning*, *Nature* **549** (2017), no. 7671 195–202.

[220] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, *Challenges and opportunities in quantum machine learning*, *Nature Computational Science* **2** (2022), no. 9 567–576.

[221] J. Liu, M. Liu, J.-P. Liu, Z. Ye, Y. Alexeev, J. Eisert, and L. Jiang, *Towards provably efficient quantum algorithms for large-scale machine-learning models*, *arXiv preprint arXiv:2303.03428* (2023).

[222] A. Miessen, P. J. Ollitrault, F. Tacchino, and I. Tavernelli, *Quantum algorithms for quantum dynamics*, *Nature Computational Science* **3** (2023), no. 1 25–37.

[223] B. Peng, S. Gulania, Y. Alexeev, and N. Govind, *Quantum time dynamics employing the Yang-Baxter equation for circuit compression*, *Physical Review A* **106** (2022), no. 1 012412.

[224] S. Gulania, Z. He, B. Peng, N. Govind, and Y. Alexeev, *QuYBE-an algebraic compiler for quantum circuit compression*, in *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, pp. 406–410, IEEE, 2022.

[225] D. A. Fedorov, B. Peng, N. Govind, and Y. Alexeev, *Vqe method: a short survey and recent developments*, *Materials Theory* **6** (2022), no. 1 1–21.

[226] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, *Quantum computational chemistry*, *Reviews of Modern Physics* **92** (2020), no. 1 015003.

[227] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya, *et. al.*, *Quantum chemistry in the age of quantum computing*, *Chemical reviews* **119** (2019), no. 19 10856–10915.

[228] L. Bittel and M. Kliesch, *Training variational quantum algorithms is NP-hard*, *Physical review letters* **127** (2021), no. 12 120502.

[229] K. J. Sung, J. Yao, M. P. Harrigan, N. C. Rubin, Z. Jiang, L. Lin, R. Babbush, and J. R. McClean, *Using models to improve optimizers for variational quantum algorithms*, *Quantum Science and Technology* **5** (2020), no. 4 044008.

[230] X. Bonet-Monroig, H. Wang, D. Vermetten, B. Senjean, C. Moussa, T. Bäck, V. Dunjko, and T. E. O'Brien, *Performance comparison of optimization methods on variational quantum algorithms*, *Physical Review A* **107** (2023), no. 3 032407.

[231] A. Gilyén, S. Arunachalam, and N. Wiebe, *Optimizing quantum optimization algorithms via faster quantum gradient computation*, in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1425–1444, SIAM, 2019.

[232] G. González-García, R. Trivedi, and J. I. Cirac, *Error propagation in NISQ devices for solving classical optimization problems*, *PRX Quantum* **3** (2022), no. 4 040326.

[233] G. De Palma, M. Marvian, C. Rouzé, and D. S. França, *Limitations of variational quantum algorithms: a quantum optimal transport approach*, *PRX Quantum* **4** (2023), no. 1 010309.

[234] R. Harper, S. T. Flammia, and J. J. Wallman, *Efficient learning of quantum noise*, *Nature Physics* **16** (2020), no. 12 1184–1188.

[235] S. Endo, S. C. Benjamin, and Y. Li, *Practical quantum error mitigation for near-future applications*, *Physical Review X* **8** (2018), no. 3 031027.

[236] Y. Suzuki, S. Endo, K. Fujii, and Y. Tokunaga, *Quantum error mitigation as a universal error reduction technique: applications from the NISQ to the fault-tolerant quantum computing eras*, *PRX Quantum* **3** (2022), no. 1 010345.

[237] J. Liu and H. Zhou, *Reliability modeling of NISQ-era quantum computers*, in *2020 IEEE international symposium on workload characterization (IISWC)*, pp. 94–105, IEEE, 2020.

[238] H. Wang, Y. Ding, J. Gu, Y. Lin, D. Z. Pan, F. T. Chong, and S. Han, *QuantumNAS: Noise-adaptive search for robust quantum circuits*, in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 692–708, IEEE, 2022.

[239] A. B. Magann, C. Arenz, M. D. Grace, T.-S. Ho, R. L. Kosut, J. R. McClean, H. A. Rabitz, and M. Sarovar, *From pulses to circuits and back again: A quantum optimal control perspective on variational quantum algorithms*, *PRX Quantum* **2** (2021), no. 1 010101.

[240] S. Dasgupta and T. S. Humble, *Assessing the stability of noisy quantum computation*, *arXiv preprint arXiv:2208.07219* (2022).

[241] S. Dasgupta and T. S. Humble, *Characterizing the reproducibility of noisy quantum circuits*, *Entropy* **24** (2022), no. 2 244.

[242] F. Lin, X. Fang, and Z. Gao, *Distributionally robust optimization: A review on theory and applications*, *Numerical Algebra, Control and Optimization* **12** (2022), no. 1 159–212.

[243] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, *Wasserstein distributionally robust optimization: Theory and applications in machine learning*, in *Operations research & management science in the age of analytics*, pp. 130–166. Informs, 2019.

[244] E. Delage and Y. Ye, *Distributionally robust optimization under moment uncertainty with application to data-driven problems*, *Operations research* **58** (2010), no. 3 595–612.

[245] S. S. Tay, C. S. Foo, U. Daisuke, R. Leong, and B. K. H. Low, *Efficient distributionally robust Bayesian optimization with worst-case sensitivity*, in *International Conference on Machine Learning*, pp. 21180–21204, PMLR, 2022.

[246] G. Iannelli and K. Jansen, *Noisy bayesian optimization for variational quantum eigensolvers*, *arXiv preprint arXiv:2112.00426* (2021).

[247] S. Duffield, M. Benedetti, and M. Rosenkranz, *Bayesian learning of parameterised quantum circuits*, *arXiv preprint arXiv:2206.07559* (2022).

[248] C. N. Self, K. E. Khosla, A. W. Smith, F. Sauvage, P. D. Haynes, J. Knolle, F. Mintert, and M. Kim, *Variational quantum algorithm with information sharing*, *npj Quantum Information* **7** (2021), no. 1 1–7.

[249] J. Mueller, W. Lavrijsen, C. Iancu, *et. al.*, *Accelerating noisy VQE optimization with Gaussian processes*, *arXiv preprint arXiv:2204.07331* (2022).

[250] S. Tibaldi, D. Vodola, E. Tignone, and E. Ercolessi, *Bayesian optimization for QAOA*, *arXiv preprint arXiv:2209.03824* (2022).

[251] J. R. Finzgar, M. Schuetz, K. Brubaker, and H. Katzgraber, *Designing quantum annealing schedules with Bayesian optimization*, *arXiv preprint arXiv:2305.13365* (2023).

[252] K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf, *et. al.*, *Kernel mean embedding of distributions: A review and beyond*, *Foundations and Trends® in Machine Learning* **10** (2017), no. 1-2 1–141.

[253] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. De Jong, *Classical optimizers for noisy intermediate-scale quantum devices*, in *2020 IEEE international conference on quantum computing and engineering (QCE)*, pp. 267–277, IEEE, 2020.

[254] I. Bogunovic, J. Scarlett, S. Jegelka, and V. Cevher, *Adversarially robust optimization with Gaussian processes*, *Advances in neural information processing systems* **31** (2018).

[255] K. Q. Ye, *Orthogonal column Latin hypercubes and their application in computer experiments*, *Journal of the American Statistical Association* **93** (1998), no. 444 1430–1439.

[256] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets*, *nature* **549** (2017), no. 7671 242–246.

[257] K. Sharma, S. Khatri, M. Cerezo, and P. J. Coles, *Noise resilience of variational quantum compiling*, *New Journal of Physics* **22** (2020), no. 4 043006.

[258] S. Wang, P. Czarnik, A. Arrasmith, M. Cerezo, L. Cincio, and P. J. Coles, *Can error mitigation improve trainability of noisy variational quantum algorithms?*, *arXiv preprint arXiv:2109.01051* (2021).

[259] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, *Noise-induced barren plateaus in variational quantum algorithms*, *Nature communications* **12** (2021), no. 1 6961.

[260] E. Fontana, M. Cerezo, A. Arrasmith, I. Rungger, and P. J. Coles, *Non-trivial symmetries in quantum landscapes and their resilience to quantum noise*, *Quantum* **6** (2022) 804.

[261] D. Rabinovich, E. Campos, S. Adhikary, E. Pankovets, D. Vinichenko, and J. Biamonte, *On the gate-error robustness of variational quantum algorithms*, *arXiv preprint arXiv:2301.00048* (2022).

[262] J. Lofberg, *YALMIP : a toolbox for modeling and optimization in MATLAB*, in *Proc. Intl. Conf. Robot. Autom.*, pp. 284–289, 2004.

[263] B. Jónsson, B. Bauer, and G. Carleo, *Neural-network states for the classical simulation of quantum computing*, *arXiv preprint arXiv:1808.05232* (2018).

[264] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, *Theory of variational quantum simulation*, *Quantum* **3** (2019) 191.

[265] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, *Quantum natural gradient*, *Quantum* **4** (2020) 269.

[266] S.-I. Amari, *Natural gradient works efficiently in learning*, *Neural computation* **10** (1998), no. 2 251–276.

[267] X. Liu, A. Angone, R. Shaydulin, I. Safro, Y. Alexeev, and L. Cincio, *Layer VQE: A variational approach for combinatorial optimization on noisy quantum computers*, *IEEE Transactions on Quantum Engineering* **3** (2022) 1–20.